

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava 4

Pinta.sk

Digital Sweatshop

Dokumentácia k inžinierskemu dielu

Študijný program: Informačné systémy, Softvérové inžinierstvo

Ročník: 1.

Predmet: Tímový projekt I.

Vedúci projektu: Ing. Dušan Zeleník

Členovia tímu: Bc. Filip Bednárík

Bc. Róbert Černý

Bc. Marek Lenčes

Bc. Miroslav Molnár

Bc. Patrik Štrba

Bc. Martin Toma

Bc. Miroslav Vojtuš

Ak. rok: 2013/2014

Obsah

1	Úvod	6
2	Globálne ciele projektu	7
2.1	Zimný semester	7
2.2	Letný semester	7
2.2.1	Ciele vymedzujúce oblasť navrhovaného systému	7
2.2.2	Nefunkcionálne ciele pre letný semester	7
3	Opisy príbehov	9
3.1	Pred prvým šprintom	9
3.1.1	Analýza existujúcich riešení	9
3.1.2	Výber a inštalácia podporných nástrojov	9
3.1.3	Vytvorenie webového sídla	11
3.1.4	Lean Canvas	12
3.2	Šprint #1	13
3.2.1	Vytvorenie základných rozhraní aplikácie	13
3.2.2	Rozbitie prvej User Story	13
3.2.3	Návrh rozhrania webovej aplikácie	14
3.2.4	Autorizácia a autentifikácia používateľa v systéme	15
3.2.5	Upload súborov ZIP a ich spracovanie	15
3.2.6	Definovanie typov úloh, vytvorenie štruktúry	17
3.3	Šprint #2	19
3.3.1	Zjednotenie Git repozitára	19
3.3.2	Dokončenie detailov	19
3.4	Šprint #3	20
3.4.1	Usporiadanie produktového backlogu	20
3.4.2	Vytvorenie metodík	20
3.4.3	Revízia doterajších zápisníc	20
3.4.4	Vytvorenie formulára pre vyplnenie profilu	20
3.4.5	Nasadenie produkčného servera	20
3.4.6	Získanie informácií prostredníctvom IP používateľa	20
3.5	Šprint #4	22
3.5.1	Vytvorenie filtrov a kreditového systému	22
3.5.2	Vytvorenie referral systému	23
3.5.3	Vytvorenie profilovej stránky	23
3.5.4	Integrácia so sociálnou sieťou Facebook	24
3.5.5	Integrácia so sociálnou sieťou LinkedIn	25
3.5.6	Vytvorenie achievement systému	26
3.6	Šprint #5	28
3.6.1	Vytváranie dokumentácie	28
3.6.2	Facebook lajky a komentáre	28
3.6.3	Integrácia s Google účtom	29
3.6.4	Integrácia s Twitter účtom	29
3.7	Šprint #6	30
3.7.1	Vytvorenie viackrokových formulárov	30
3.7.2	Rebranding - web tímu, dizajn	31
3.7.3	Zlepšená organizácia profilu	31
3.8	Šprint #7	35

3.8.1	Zarovnanie dizajnu produktovej aplikácie s novou identitou tímu	35
3.8.2	Survey task	35
3.8.3	Vytvorenie grafických návodov pre landing page	35
3.8.4	Integrácia so sociálnou sieťou Twitter	35
3.9	Šprint #8	37
3.9.1	Zmena dizajnu badges	37
3.9.2	Vytváranie záznamov z externej aplikácie	38
3.9.3	Úloha s pridávaním tzv. Facebook lajkov	39
3.9.4	Intenacionalizácia (SK, ENG)	41
3.9.5	Zobrazovanie výsledkov	42
3.10	Šprint #9	44
3.10.1	Vytvorenie survey kampane	44
3.10.2	Oprava kľúčových chýb v systéme	46
3.10.3	Zmena layoutu pre zobrazenie výsledkov. Zovšeobecnenie dátového modelu pre existujúce typy úloh a ich podtypy (survey)	46
3.10.4	Vytvorenie prezentačnej stránky aplikácie	47
3.10.5	Refaktoring spojenej úlohy	48
3.10.6	Proces zefektívnenia tvorby zápisníc - automatický generátor šablóny	48
3.11	Šprint #10	50
3.11.1	Vytvorenie prototypu notice úlohy	50
3.11.2	Návrh a implementácia zobrazovania úloh	50
3.11.3	Návrh API rozhrania	52
3.11.4	Proces vytvárania kampane	53
3.11.5	Generátor zápisníc v.2 a úprava vzhľadu	55
3.11.6	Rozšírenie výsledkov o ďalšie typy grafov. Rýchla zmena bez znovu načítania dát	55
3.11.7	Zoznam úloh	56
3.11.8	Flow s riešením úloh	57
3.12	Šprint #11	58
3.12.1	Identifikácia chýb v systéme a ich odstránenie	58
3.12.2	Vylepšenie zobrazenia výsledkov. Pridanie filtrov. Csv výstup dát.	59
3.12.3	Implementácia HTTP overovania používateľa pre API	60
3.12.4	Vytvorenie prvej verzie transakcií	61
3.12.5	Optimalizácia postupu pri vytvorení úlohy a pridanie úpravy celého postupu pri vytváraní úlohy	61
3.12.6	Zlepšiť zobrazovanie badges	62
3.12.7	Pridanie referral linku do profilu, pridanie zručností a zobrazenie aktuálnych úloh na dashboarde	63
3.13	Šprint #12	65
3.13.1	Vylepšenie navigačného panelu, oprava niekoľkých chýb	65
3.13.2	Transakčný systém 2	65
3.13.3	Úprava načítavania úrovne jazyka z LinkedIn	67
3.13.4	Vytvorenie dashboardu	67
3.13.5	Znovupoužiteľnosť výsledkov v detaile tasku	68
3.13.6	Oprava chýb	69
3.13.7	Global grid flow design	70

3.13.8	Zobrazenie notifikácií na hlavnej stránke	71
3.13.9	Doplnenie seedov do DB, tvorba dokumentácie a zrevidovanie obsahu doterajších zápisníc	72
3.14	Šprint #13	73
3.14.1	Stretnutie s marketérkou, oprava chýb	73
3.14.2	Internacionalizácia v2	73
3.14.3	Responzívny dizajn hlavnej stránky	74
3.14.4	Unikátne dáta v profile	75
3.14.5	Oprava chýb v systéme	76
3.14.6	Zlepšenie procesu vytvárania kampaní	77
3.14.7	Nové typy Survey podúloh	78
3.14.8	Hall of Fame	79
3.14.9	Nastavenia pre používateľa a zobrazovanie úloh	80
3.15	Šprint #14	81
3.15.1	IIT.SRC, Úprava homepage obsahu	81
3.15.2	Umožniť používateľovi ovplyvniť vzhľad textu otázky	82
3.15.3	Refaktorovanie jadra aplikácie	82
3.15.4	Vytvorenie funkčných a unit testov	83
3.15.5	Validácia nových typov úloh a ukladanie výsledkov	84
3.15.6	Autocomplete	85
3.16	Šprint #15	86
3.16.1	Oprava identifikovaných chýb v systéme	86
3.16.2	Automatické uzamykanie kampaní (CRON)	87
3.16.3	Rozdelenie seedov	88
3.16.4	Oprava chýb v systéme	88
3.16.5	Oprava chýb	92
3.16.6	Nasadzovanie aplikácie na produkčný server a riešenie chýb s tým spojených	92
3.16.7	Oprava chýb	93
3.16.8	Oprava chýb	93
3.17	Šprint #16	94
3.17.1	Robime.IT článok, UI vylepšenia a bugfixy	94
3.17.2	Pridanie nových typov achievementov. Vytvoriť používateľsky prívetivejšie obrázky	95
3.17.3	Vytvorenie statických stránok	96
3.17.4	Doplnenie transakčnej logiky do notice kampane	98
3.17.5	Automatické počítanie ceny kampane	98
3.17.6	Vytvorenie prezentačného videa	99
3.17.7	Nové funkcie do dashboardu	99
3.17.8	User management + transakcie	100
3.17.9	Filtre	100
3.17.10	Welcome tasky pre IIT.SRC	101
3.17.11	Úprava aplikácie a oprava chýb a nasadenie do produkcie	102
3.17.12	Povolenie správnych hodnôt v Education a Positions	103
3.18	Záverečný šprint	104
4	Celkový pohľad	105
4.1	Opis prototypu	105
4.2	Funkcionálne požiadavky	105
4.3	Prípady použitia	106

4.4	Dátový model	107
4.5	Letný semester	109
4.5.1	Celkový dátový model	110
5	Používateľská príručka	111
5.1	Prihlásenie do systému	111
5.2	Registrácia do systému	112
5.3	Opätovné zaslanie aktivačného emailu	112
5.4	Sekcia pre pracovníka	113
5.4.1	Profil	113
5.4.2	Badges	114
5.4.3	All tasks	114
5.5	Sekcia pre zadávateľa	115
5.5.1	Profil	115
5.5.2	Badges	115
5.5.3	Tasks	115
6	Testovanie treťou stranou	119
6.1	Výsledky testovania	122
6.2	Vyhodnotenie testovania	124
7	Čo sme nestihli	125
8	Čo sme sa naučili	126

1 Úvod

Účel dokumentu

Tento dokument vznikol na predmete tímový projekt iniciatívou členov tímu 11. Slúži na dokumentovanie projektu s názvom Digital Sweatshop.

Štruktúra dokumentu

Štruktúra dokumentu je prispôsobená agilnému, prírastkovému a iteratívne projektu manažmentu (SCRUM). V jednotlivých kapitolách dokumentu je charakterizovaný konkrétny používateľský príbeh spoločne s jeho špecifikáciou, implementáciou a testovaním.

Zadanie projektu

Známy Digital Sweatshop je napríklad AMT. Spravme vlastný "pracovný tábor", kde objednávateľ zadá prácu a pracant ju za odmenu vykoná. Od experimentov za pívá, cez marketingové prieskumy za zľavy, až po domáce úlohy za body tak zmeňme model mikropráce na Webe.

Prečo si mám vybrať tento projekt:

Okrem toho, že sa veľa naučíš, vytvoríš niečo, čo už teraz má svojich cieľových zákazníkov. Už prvý prototyp budú využívať chtiví experimentujúci skupiny PeWe a k vypusteniu na svetlo sveta to už bude len krôčik.

Napríklad:

pracuješ na svojej diplomovke, kde je nutné vašu metódu nejako overiť. Vytvoríš svoj projekt, napríklad stránku, ktorá zobrazuje výstupy tvojej metódy používateľovi. Ako to overiť, ako to chceš stihnúť? Nuž, na poslednú chvíľu zoženieš 20 ľudí, ktorí ti niečo poklikajú a aj tak to bude málo. Preto vytvoríme tento SweatShop. Zadáš úlohu, opíšeš misiu, nastaviš odmenu, použiješ jednoduché API, pridáš ešte zopár otázok, na ktoré chceš odpoveď a čakáš. Misia sa zobrazí všetkým čakajúcim pracantom a brigáda môže začať. Po ukončení dostaneš výsledky a pracanti svoju odmenu.

Idea je tá istá s akou pracujú aj iné projekty pre mikroprácu. Zaujímavá obmena je v spôsobe odmeňovania. Kto by už robil za pár centov nudnú prácu. Vytvoríme platformu, kde formou hry vyhodnotíme a odmeníme zúčastnených pracantov. Záleží na zadávateľovi, či dá výhercovi čokoládu, alebo víno. Tento model v reále už funguje minimálne v skupine PeWe, kde sú experimentujúci odmeňovaní podobnými darmi. Podme túto realitu zastrešiť systémom, ktorý sa o mnohé postará. Napríklad aj o automatickú kontrolu, ktorú zadávateľ môže definovať. Nepôjde preto o prácu, ktorú možno oklamať a podmienky pre každého tak zostanú rovnaké.

Základnou náplňou projektu je prostredníctvom webovej aplikácie poskytnúť istý druh rozhrania medzi dvoma skupinami ľudí, z ktorých jedna skupina potrebuje získať určitú formu spätnej väzby a druhá skupina ľudí si chce privyrobiť a v neposlednom rade aj trochu pomôcť. Využitie tohto rozhrania je tak veľké, že môžeme pokryť až veľmi širokú oblasť. Od akademickej pôdy pri rôznych formách úloh, od dotazníkov cez jednoduché úlohy až po zložitejšie overovanie hypotéz

vrámci bakalárskych, diplomových či dizertačných prác.

Ďalšou veľkou oblasťou je marketing. Nahradenie zastaraných marketingových techník a aktivít

(telefónny marketing) za priame oslovenie cieľovej skupiny prostredníctvom nášho rozhrania. Možnosti využitia tohto rozhrania sú teda naozaj skoro neobmedzené.

2 Globálne ciele projektu

2.1 Zimný semester

Základným cieľom projektu je vytvoriť aktívne používanú webovú či mobilnú aplikáciu, ktorej hlavným cieľom je sprostredkovať mikroprácu. Chceme dosiahnuť spokojnosť na oboch stranách (zadávateľ aj pracovník). Zadávateľovi chceme ponúknuť možnosť získať kvalitnú spätnú väzbu tým že efektívne nájdeme správnych kandidátov na danú úlohu. Pracovníkovi chceme dať možnosť si privyrobiť a motivovať ho systémom reputácie a odmeňovania.

Projekt má potenciál poskytnúť rozhranie využiteľné v rôznych oblastiach. Hlavným cieľom tohto projektu je však pokrytie akademickej a marketingovej oblasti. V budúcnosti sa však určite zameriame aj na iné oblasti.

2.2 Letný semester

V tejto časti sú opísané globálne ciele, ktoré sme si stanovili pre letný semester. vychádzajúc zo zistených nedostatkov vyvíjaného systému a skúseností z vývoja a tímovej práce. Nasledujúce globálne ciele sú zoradené podľa priorit.

2.2.1 Ciele vymedzujúce oblasť navrhovaného systému

1. vybudovať online priestor, ktorý bude plniť úlohu trhu so spätnou väzbou
2. ponúknuť jednoduchú možnosť cielenia kampaní podľa rôznych atribútov (pohlavie, vek, miesto bydliska, vzdelanie, vedomosti, záujmy...)
3. motivovať poskytovateľov spätnej väzby tým, že za svoju participáciu získajú odmenu a že celý proces poskytnutia tejto spätnej väzby bude jednoduchý a intuitívny,
4. poskytnúť zadávateľom úloh možnosť vizualizácie výsledkov.

2.2.2 Nefunkcionálne ciele pre letný semester

Tieto ciele sme si stanovili aby sme dospeli k maximálne kvalitnému, konkurencie schopnému produktu v rámci súťaže TP-cup. Tieto ciele nás donútia rozšíriť aktuálny backlog a zároveň zúžiť profil navrhovanej aplikácie. Dôvodom je široký záber user stories, ktoré sme navrhli v priebehu minulého semestra. Tieto pravdepodobne nebude možné splniť v primeranej kvalite. Aby sme splnili ciele vymedzujúce oblasť ktorú riešime identifikovali sme nasledovné nefunkcionálne ciele:

1. Vytvoríť plne funkčné procesy pre minimálne jeden identifikovaný typ úlohy.

Na základe retrospektívy doterajšej práce sme sa na začiatku letného semestra zhodli, že aktuálne úlohy a ich riešenie je v niektorých prípadoch príliš roztrúsené a nie sú identifikované body spojenia medzi typmi úloh. Napríklad úloha tagovania obrázkov má odlišnejší proces tvorby a vyplňania ako úloha survey. Preto sme si stanovili za cieľ určiť jeden, kľúčový typ úlohy a ten na všetkých úrovniach postupne vyladiť. Výhodou oproti prístupu 1 skupina : 1 typ úlohy => n riešení, je hlavne ľahké rozbitie na podúlohy a tým pádom aj cielená špecializácia jednotlivých vývojárov.

Identifikované pod-segmenty procesu tvorby úloh:

- zadanie úlohy
- riešenie úlohy
- zhodnotenie výsledkov úlohy

2. Zapojiť do riešenia aj tretiu stranu. Priebežné overenie vyvíjaného softvéru.

Tvorba softvéru v uzavretých podmienkach často vyústi do vyprodukovania nepotrebných funkcií a často aj k odklonu od primárneho účelu. Aby sme predišli týmto problémom, stanovili sme si za cieľ zainteresovať do vývoja aj potenciálnu klientelu. Najmä však potenciálneho zadávateľa. Ten by mal byť zdrojom kapitálu a štartérom vnútornej ekonomiky systému.

3. Zvýšenie kvality používateľského rozhrania.

Tento cieľ zahŕňa úpravy grafickej prezentácie aj funkcionálne zmeny pre z kvalitné UX (user experience). Používateľské rozhranie je aktuálne v podstate neprezentovateľné. Bude nutné zahrnúť do šprintov prácu na vylepšenie grafického vzhľadu a zjednotenie vzhľadu jednotlivých elementov. Pre z kvalitné UX bude nutné zjednotiť ovládacie prvky systému.

4. Zabezpečenie kvality

Keďže sme z dôvodu časovej náročnosti TDD prístupu tento opustili, museli sme si stanoviť aj tento cieľ ako významný. Preto bolo súčasťou navrhovaného plánu zo zimného semestra šprint orientovaný priamo na odhaľovanie a riešenie bugov v systéme. Testovanie musí pokryť systém v plnej miere zverejnenú funkcionálnosť, ktorá sa bude prezentovať na TP-cupe.

5. Zefektívniť prácu na tímových stretnutiach

Stretnutia sa nám darilo dodržiavať podľa stanoveného harmonogramu. 3 hodiny boli postačujúce na prejednanie problematiky avšak komunikácia bola nekoordinovaná a často neproduktívna. Preto sme si ako globálny cieľ určili aj zosilnenie role Scrum mastera. Ten bude musieť dohliadať na plynulosť a organizovanosť stretnutí.

3 Opisy príbehov

3.1 Pred prvým šprintom

3.1.1 Analýza existujúcich riešení

Autor: Bc. Marek Lenčes

Amazon Mechanical Turk

Podobným riešením pre ponuku mikropráce je systém Amazon Mechanical Turk (MTurk). Obsahuje úlohy, ktoré sa nazývajú HITy (Human intelligence task), ktoré sú jednoduchými úlohami ako napríklad vyhľadanie informácií o firme na webe, tagovanie obrázkov, alebo písanie recenzií. MTurk sa momentálne nachádza v beta verzii a je dostupný len pre zadávateľov na území USA, no pracovníci môžu byť z celého sveta. Zadávateľ môže vyžadovať od pracovníka špeciálnu kvalifikáciu na vykonanie úlohy, ak je to potrebné. Pracovníka môže zadávateľ aj odmietnuť, a tým sa pracovníkovi zhorší reputácia v rámci systému, a bude mať problém získať úlohu na splnenie. Zadávateľia vyplácajú 10% z úspešne vykonaných úloh, prevádzkovateľovi systému, ktorým je spoločnosť Amazon.

The screenshot shows the Amazon Mechanical Turk website interface. At the top, there's a navigation bar with 'Your Account', 'HITS', and 'Qualifications' tabs. Below that, a banner states 'Mechanical Turk is a marketplace for work.' and 'We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it's convenient. 443,562 HITs available. View them now.' The main content is divided into two columns. The left column, 'Make Money by working on HITs', explains that HITs are individual tasks and lists benefits for workers: working from home, flexible hours, and payment for good work. It includes a flow diagram: 'Find an interesting task' (with a magnifying glass icon) -> 'Work' (with a gear icon) -> 'Earn money' (with a dollar sign icon). The right column, 'Get Results from Mechanical Turk Workers', explains how requesters can use the platform and lists benefits: access to a global workforce, fast completion, and payment upon satisfaction. It includes a flow diagram: 'Fund your account' (with a plus sign icon) -> 'Load your tasks' (with a document icon) -> 'Get results' (with a star icon).

Obr. 1: Amazon MTurk

3.1.2 Výber a inštalácia podporných nástrojov

Autor: Bc. Patrik Štrba, Bc. Martin Toma

Ruby on Rails

Jedným z prvých spoločných rozhodnutí tímu bol výber základných technológií. Vzhľadom na špecifiká projektu (webová aplikácia) a pomerne „start-up“ like charakter sme po miernom výbere zúžili možné technológie na tieto tri:

- Groovy (JVM based)
- Ruby (Ruby on Rails)

- Python (Django)

Naším prvým rozhodnutím bol jazyk Groovy založený na jazyku JAVA. Prvé dva šprinty sme venovali aktivitám, ktoré ráтали s týmto jazykom. Viaceré problémy nás však donútili prehodnotiť tento výber a rozhodnúť sa, či nám rýchla zmena technológií v súčte neušetrí mnoho času v porovnaní s tým, keby sme riešili problémy našej pôvodne vybranej technológií.

Rozhodli sme sa kým bol ešte čas zmeniť technológiu na projektovým vedúcim odporúčaný framework Ruby on Rails, v ktorom momentálne vyvíjame. Konkrétne používame verziu Ruby 2.0 a verziu Rails 4.0. Mierne problémy nám spôsobuje pomerne čerstvo vydaná verzia ale chceli sme vyvíjať v tom najnovšom a najmodernejšom prostredí aké je k dispozícii.

JetBrains RubyMine

Pri výbere nástroja na vývoj sme museli vybrať nástroj, ktorý spĺňa nasledujúce podmienky:

- musí byť zadarmo (alebo školská licencia)
- multiplatformové
- dobrá podpora jazyka ruby
- dobrá podpora webového frameworku ruby on rails
- aktívny vývoj

Na základe týchto podmienok nám ostal len nástroj RubyMine od spoločnosti JetBrains.

Atlassian JIRA

Manažment úloh je proces, ktorý taktiež vyžaduje softvérovú podporu. Pri výbere nástroja na tento účel sme požadovali tieto vlastnosti:

- podpora agilných metód (konkrétne SCRUM),
- jednoduchosť a intuitívnosť,
- možnosť externého hostingu.

Z týchto kritérií nám zostalo viacero kandidátov, no v konečnom dôsledku sme sa rozhodli pre JIRU od spoločnosti Atlassian. Tento produkt je v oblasti manažmentu úloh momentálne svetovou jednotkou a to bol rovnako dôvod nášho výberu. Momentálne teda používame JIRU vo verzii 5.1.6, s rozšírením Grasshopper pre podporu agilnej metódy SCRUM.

Bitbucket

V tejto oblasti existujú dve populárne služby na zdieľanie repozitárov a to:

- Bitbucket
- Github

Služba musela spĺňať nasledovné podmienky:

- musí byť zadarmo (minimálne pre 7 ľudí)
- možnosť vytvorenia privátneho repozitára
- musí podporovať správu oprávnení

Tieto podmienky spĺňajú obe služby a preto sme sa spoločne rozhodli pre github. Avšak neskôr pre problémy pri integrácii s nástrojom JIRA sme sa rozhodli použiť Bitbucket, ktorý poskytuje rovnaká firma ako nástroj JIRA.

Jenkins

Pri procese nasadenia sme museli vybrať nástroj, ktorý spĺňa nasledujúce podmienky:

- musí byť zadarmo
- podpora jazyka ruby
- jednoduchá správa
- podpora verziovacieho nástroja git
- dostupný pre unix/linux systémy
- podpora v službe Bitbucket

Na základe týchto podmienok a skúsenosti sme vybrali nástroj Jenkins, ktorý po pridaní zmien do repozitára (vetvy master) spustí proces nasadenia a aktualizuje produkčnú verziu projektu.

3.1.3 Vytvorenie webového sídla

Autor: Bc. Róbert Černý, Bc. Marek Lenčes, Bc. Miroslav Molnár

Medzi primárne ciele nášho tímu pre prvý šprint patrí vytvorenie webovej stránky tímu 11. Služi na prezentovanie vykonanej práce tímu, ako aj poskytnutie dôležitých informačných zdrojov o postupe a aktuálnom stave vyvíjaného projektu. Spoločne sme určili tieto požiadavky, ktoré webová forma prezentácie nášho tímu bude spĺňať:

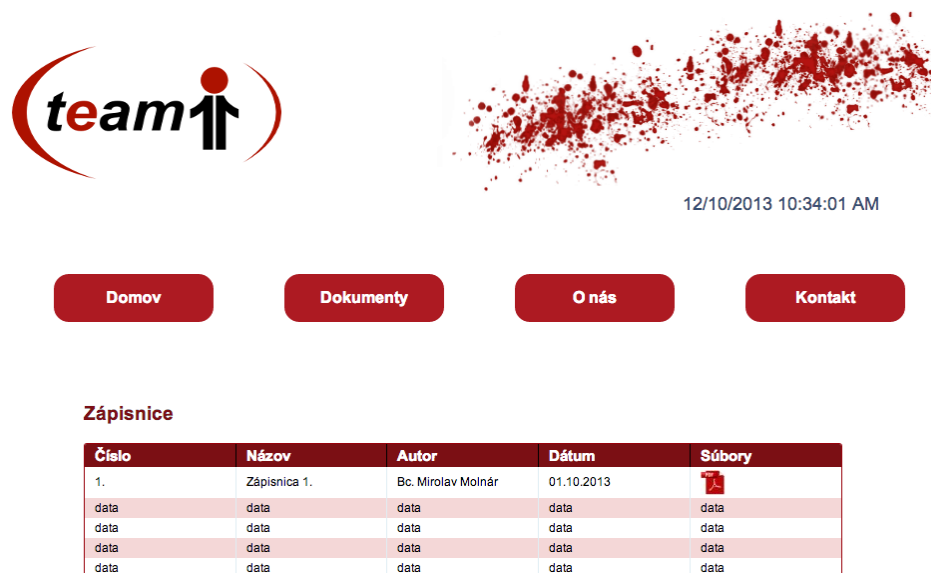
- ľahká dostupnosť a kvalita informačných zdrojov (dokumenty, plagát)
- kvalita nad kvantitou
- intuitívna navigácia na stránke
- responzivnosť
- minimalistický dizajn
- dizajn prispôbený nášmu zadaniu (logo, motívy)

Na stránke sa budú nachádzať informácie o projekte Digital Sweatshop, ako aj motivácia a potenciál tohto projektu. Zodpovieme na základné otázky ako:

- Čo to je Digital Sweatshop ?
- Prečo to robíme ?
- Ako to funguje ?
- Kto sú členovia tímu 11 ?

Návštevník si môže dôkladne prezrieť zostavenie tímu a popis jednotlivých členov. Každý člen má pri svojom popise aj aktuálnu fotku. Pre možnosť komunikácie je v popise člena tímu uvedený aj jeho emailový kontakt.

Samozrejmosťou je sekcia dokumentov, kde budú voľne dostupné dokumenty na stiahnutie. Ich cieľom je zaznamenať postup, aký sme pre plnenie tohto projektu zvolili. Prvý prototyp spĺňa nutné podmienky avšak, všetci sme si vedomí, že po bližšej špecifikácii, vytvorení loga a príznačného názvu pre náš projekt, dáme našej webovej stránke iný rozmer. Náhľad na prvú verziu nášho webového sídla po prvom šprinte je zobrazený nižšie.



Obr. 2: Prototyp tímového webu

3.1.4 Lean Canvas

Autor: Všetci členovia

Academics and Marketing

PROBLEM Obtaining high-quality product feedback or feedback from specific group.	SOLUTION Clients are posting tasks and users are solving them.	UNIQUE VALUE PROPOSITION Very user/friendly (intuitive) UI. Focus also on Slovak market. User motivation program Game-like features.	UNFAIR ADVANTAGE Something that can't be easily copied or bought 9	CUSTOMER SEGMENTS Academics and Science communities Marketing and PR EARLY ADOPTERS Both -> IT Experienced Clients -> solvent Users -> excited
	EXISTING ALTERNATIVES Telemarketing Surveys Public interviews		KEY METRICS Number of posted tasks. Number of processed tasks. Number of registered users/clients. Number of active users/clients.	
COST STRUCTURE Maintance Hosting and scaling costs		REVENUE STREAMS From clients (posting tasks) From both (small part from task revenue)		

Obr. 3: Lean Canvas

3.2 Šprint #1

3.2.1 Vytvorenie základných rozhraní aplikácie

Registrácia a prihlásenie

Autor: Bc. Patrik Štrba, Bc. Martin Toma

Ako skoro každá webová aplikácia, tak aj tá naša vyžaduje funkcionality registrácie a následného prihlásenia používateľov. Preto sme sa už v druhom šprinte rozhodli implementovať základ tejto funkcionality. Navrhli a implementovali sme teda základné možnosti registrácie s použitím jednoduchej metódy šifrovania hesla v databáze. Po šprinte sme teda boli schopní simulovať registráciu a následné prihlásenie do aplikácie. Zhodli sme sa, že neskôr bude asi potrebné implementovať sofistikovanejšie metódy šifrovania, overenie registrácie prostredníctvom emailu a možnosti obnovenia strateného hesla. To sme však v danej dobe neriešili.

Vytvorenie jednoduchej úlohy

Autor: Bc. Róbert Černý, Bc. Filip Bednárík

Na pohľad veľmi jednoduchá úloha avšak pri jej implementácii je nutné vytvoriť dátový model a prispôbiť ho požiadavkam na systém. Vytváranie a plnenie úloh v našej aplikácii tvorí jadro systému, preto je dôležité, aby sa logika poriadne premyslela. V druhom šprinte sme implementovali základný systém správy úloh a podúloh, čo je jadrom nášho systému. Na začiatok sme vytvorili model Task a Subtask, ich prepojenie a jednoduchý CRUD, ktorý nám umožňuje spravovať záznamy v databáze. Pomaly sme sa začali rozhodovať aké vlastnosti a atribúty budú niesť úlohy a podúlohy a tiež sme sa rozhodli, že riešiť sa budú podúlohy, ktoré úlohy budú agregovať. Identifikovali sme základné vlastnosti úlohy a to sú: Názov, Popis, Kategória/Typ a počet ľudí, ktorý budú túto úlohu môcť riešiť.

3.2.2 Rozbitie prvej User Story

Autor: Bc. Filip Bednárík

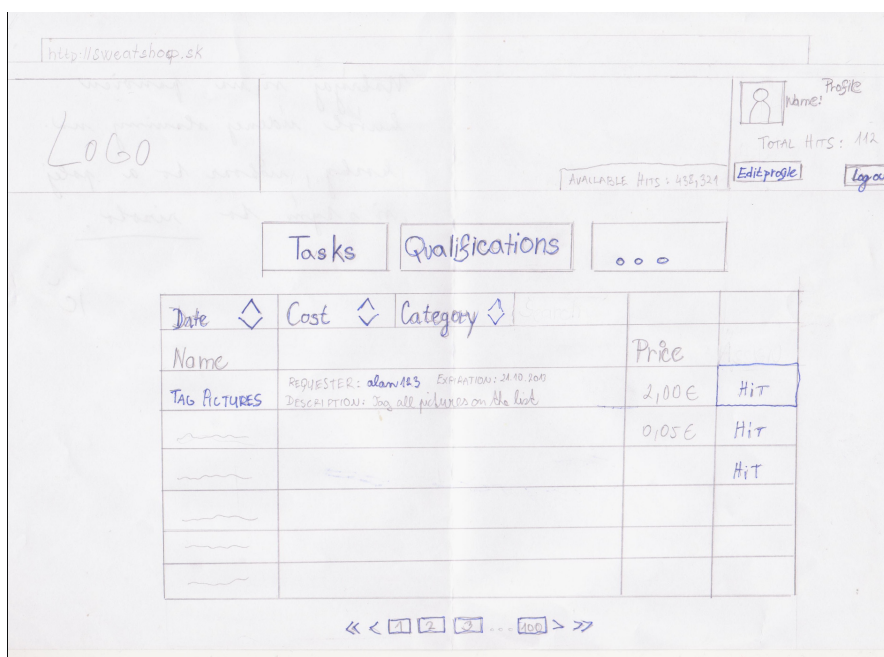
Našou primárnym cieľom je spraviť aplikáciu, ktorá bude využiteľná pre našich kolegov v akademickej doméne, kde je veľký dopyt po takomto systéme, ktorý poskytuje overenie rôznych počítačových algoritmov človekom. Najzákladnejším user story je, že zadávateľ nahrá do systému obrázky a pracovníci mu pri riešení úlohy napíšu aké objekty sa na obrázku nachádzajú. V tomto šprinte sme integrovali pridávanie a rozbaľovanie zip súborov s obrázkami a vytváranie úloh a podúloh na základe týchto obrázkov. Z pohľadu zadávateľa sme teda upravili základný CRUD tak aby vytváranie úlohy obsahovalo tlačidlo vybrať súbor a po nahratí zip súboru sa tento rozbaľil a vytvorili sa podúlohy pre každý jeden obrázok. Z pohľadu pracovníka sme navrhli obrazovku v ktorej vidia pracovníci všetky dostupné úlohy a môžu si zvoliť úlohu, ktorú budú riešiť. Pri riešení úlohy prechádzajú všetky podúlohy danej úlohy a postupne vytvárajú nové riešenia pre danú podúlohu. Pri dokončení všetkých podúloh sa označí úloha za ukončenú. Pri odídení zo stránky riešenia a návrate na stránku zoznamu všetkých úloh má pracovník možnosť sa vrátiť k riešeniu úlohy a to tak, že mu systém ponúkne vyriešiť poslednú nevyriešenú podúlohu danej úlohy.

3.2.3 Návrh rozhrania webovej aplikácie

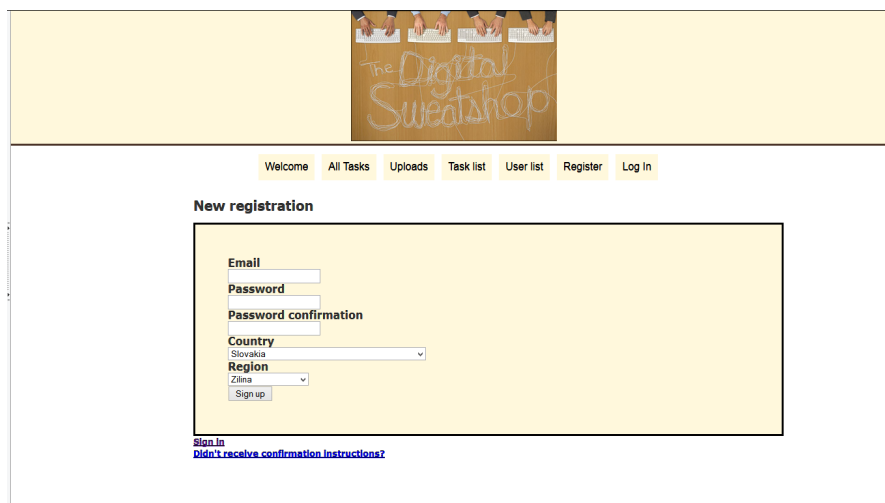
Autor: Bc. Róbert Černý

Webové rozhranie aplikácie je momentálne zložené z jednoduchých HTML súborov bez štýlovania. Vytvorením prvého návrhu dizajnu stránky zjednodušíme a sprehľadníme proces vývoja ďalších úloh. Cieľom je vytvoriť jednotné grafické rozhranie, ktorého princípy sa použijú v celej aplikácii. Základnými požiadavkami je vytvorenie navigačného panela, pre jednoduchšie zapracovávanie privilégií používateľov ako aj pre orientáciu v aplikácii. Dôležitým aspektom je aj zobrazovanie tabuliek v jednotnom formáte, nakoľko sa v systéme často objavujú. Prvá verzia UI slúži iba na demonštrovanie implementovaných používateľských príbehov a v budúcnosti sa prepracuje z rôznych aspektov.

Low fidelity návrh UI



Obr. 4: Prvý návrh grafického používateľského rozhrania aplikácie



Obr. 5: Prvý návrh grafického používateľského rozhrania aplikácie

3.2.4 Autorizácia a autentifikácia používateľa v systéme

Autor: Bc. Patrik Štrba, Bc. Martin Toma

Neskôr sme neboli dostatočne spokojný s implementáciou registrácie a prihlásenia, tak sme sa po dohovore dohodli použiť radšej už vytvorený gem, ktorý zjednoduší implementáciu a pridá ďalšiu funkcionality potrebnú do budúcnosti.

V tomto šprinte sme použili gem *devise*, ktorý rieši problém autentifikácie a zároveň umožňuje nastaviť, ktorá funkcionality sa má použiť. Tento gem obsahuje teda autentifikáciu, ale aj registráciu a funkcionality okolo toho, napríklad potvrdenie účtu pomocou emailovej adresy alebo zabudnuté heslo.

Pre korektné fungovanie tohto gemu sme museli najskôr vytvoriť model používateľa, ktorý sme neskôr spojili s funkcionality gemu.

Po dokončení autentifikácie sme prešli na analýzu autorizácie. Pre autorizáciu sme zvolili gem *can-can*, ktorý spolu s gemom *devise* zabezpečujú autentifikáciu a autorizáciu. Avšak sme posúdili, že autorizáciu ešte nie je potrebné implementovať do systému vzhľadom na malý počet používateľských rolí a malého vytvorenej funkcionality.

3.2.5 Upload súborov ZIP a ich spracovanie

Autor: Bc. Miroslav Vojtuš

Analýza

Ako jednu zo základných úloh sme identifikovali tagovanie obrázkov. Pre získanie obrázkov je nutné zvážiť dostupné možnosti. V rámci analýzy sme zvažovali:

- jednotlivé odovzdanie http protokolom - po jednom obrázku
- implementáciou rest API alebo WS klientom
- simulované hromadné odovzdanie - postupné vyberanie obrázkov po jednom a odoslanie naraz
- zoskupenie obrázkov do ZIP archívu s pevne stanovenou štruktúrou obsahu

Špecifikácia požiadaviek

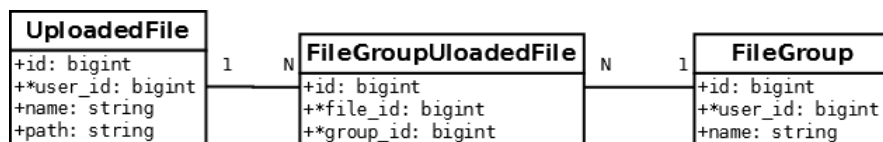
Po funkcionálnej stránke je odovzdanie jednoduchý transport súboru, ktorého obsah sa vyhodnotí a rozbalí na strane serveru a informuje používateľa o stave importu.

Nefunkcionálne požiadavky sme identifikovali nasledovné:

- stabilita prenosu
- stabilita importu
- udržateľnosť indexov súborov pre spätné operácie(CRUD)

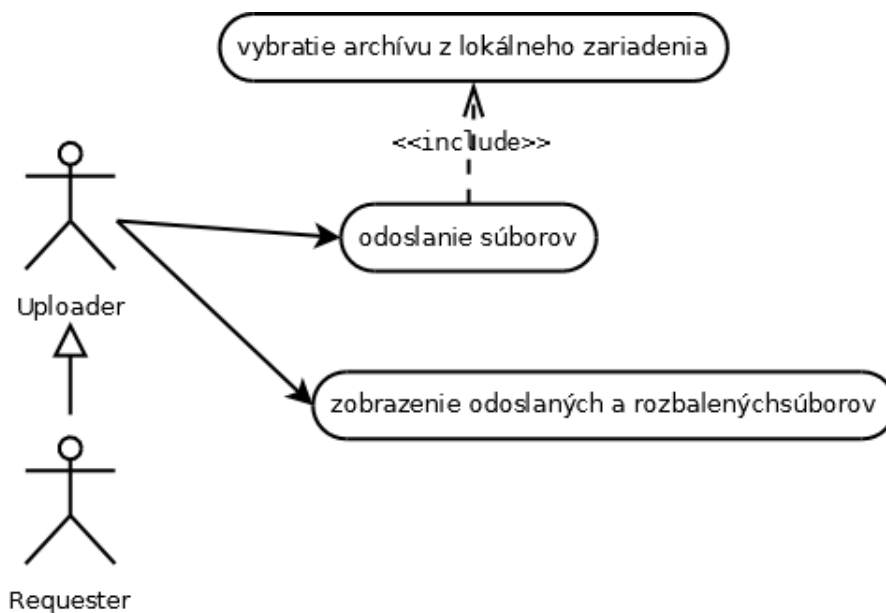
Návrh

Dátová vrstva je opísaná nasledujúcim obrázkom. V entite UploadedFile je path relatívna cesta k súboru na disku. Entita FileGroup zastrešuje zoskupenie súborov a nahrádza pojem archív, vo výsledku súbory už nie sú v jednom archíve ale len zoskupené.



Obr. 6: Dátový model

Príbeh je opísaný UC diagramom na nasledujúcom obrázku. Zadávateľ vystupujúci v role odosielateľa má možnosť odoslať súbor vybratím z lokálneho zariadenia a zobraziť si už odoslané súbory.



Obr. 7: Model prípadu použitia

Implementácia a testovanie

Príbeh je implementovaný ako jednoduchý Ruby skript naviazaný na Ruby on Rails programový rámec, kde je hlavnú funkcionality implementovaný v modeli FileUpload.

V aktuálnej verzii je validita rozbaleného archívu zabezpečená jednoduchou validáciou pomocou rozpoznávania názvov súborov a odvodenia ich typu(MIME-TYPE). Dátový obsah jednotlivých súborov nie je analyzovaný, nakoľko záleží na zobrazovači ako ošetrí bezpečnosť zobrazených dát z pohľadu potenciálnych útokov. V prípade obrázkov nehrozí žiadne vážnejšie riziko okrem odosielania nežiadúceho alebo zakázaného obsahu. V takomto prípade je nutné manuálna kontrola.

K súborom sa dá pristupovať jednotlivo pomocou abstraktnej cesty odvodennej z identifikátora súboru: '/file/:id' (napr. <http://www.pinta.sk/file/1>). Fyzická cesta k súborom je blokovaná pre verejnosť - len pre privátny prístup.

Testovanie je zabezpečené niekoľkými jednotkovými testami na úrovni odosielenia súborov pre základne hraničné prípady ako:

- prázdny archív
- neznámy MIME-TYPE
- viac ako N súborov, kde N je konštanta definovaná konfiguráciou
- archív s podzložkami
- nevalidný binárny súbor s príponou .zip - simulácia útoku

3.2.6 Definovanie typov úloh, vytvorenie štruktúry

Autor: Bc. Marek Lenčes, Bc. Miroslav Molnár

Bol vytvorený prieskum aktuálne najpopulárnejších úloh na konkurenčnom serveri Amazon MTurk, ktorý taktiež slúži na to aby používatelia mohli vytvárať a vypracovávať úlohy. MTurk je komerčný a ak si chce používateľ nejakú úlohu vytvoriť na serveri, musí im za vytvorenie zaplatiť. MTurk taktiež nieje rozšírený na viaceré krajiny ako je napríklad Slovensko. Po prejdení dostupných úloh sme vypísali príklady tých najpoužívanejších a vytvorili sme kategórie do ktorých sme ich zaradili. Spísali sme pre úlohy potrebné vstupy a výstupy.

Kategórie úloh:

Vyberanie dát z obrázka

Používateľ z obrázku vyčítava údaje, ktoré vyžaduje zadávateľ.

Príklady:

Zapísať informácie z pokladničného bloku(mena a ceny produktov).

Vstupy: obrázkov

Výstupy: textové informácie

Tagovanie obrázkov niekoľkými slovami

Vstupy: obrázok

Výstupy: textový dokument s konkrétnym počtom slov

Survey (prieskum): vyplnenie dotazníku

Príklady:

Vypracuj dotazník.

Vstupy: všeobecné informácie o sebe

Výstupy: vyplnený dotazník

Nájdenie informácii na webe: nájsť informáciu zadaných zadávateľom.

Príklady:

Nájsť kontaktnú osobu a jej email adresu pre danú URL.

Vstupy: odkaz URL

Výstupy: kontaktná osoba a údaje o nej.

Nájsť najrelevantnejší výsledok pre každé poskytnuté kľúčové slovo v internetovom vyhľadávacom google.com.

Vstupy: kľúčové slová

Výstupy: množina najrelevantnejších slov.

Nájsť webové sídlo (URL) firmy.

Vstupy: názov firmy

Výstupy: odkaz URL

kategorizácia, sortovanie, zaradenie dát: zaradte dáta do zadávateľom určených kategórií.

Príklady:

Kategorizácia obrázkov podľa kritérií (odfotený vnútri, vonku, doma, v robote).

Vstupy: obrázky

Výstupy: výberové tlačidlá s viac možnosťami pre obrázky

vytváranie textu, upraviť textu: vytvorenie dokumentu podľa zadávateľom zadaných požiadaviek.

Príklady:

Oprav a prepíš text z PDF

Vstupy: pdf

Výstupy: textový súbor

Prelož zadaný text do iného jazyka

Vstupy: text, PDF, obrázok,..

Výstupy: text

Napísať recenziu na produkt, vytvoriť dokument s požadovanými vlastnosťami a informáciami.

Vstupy: text popis

Výstupy: text dokument

Práca s médiami a spracovaní informácii z nich

Príklady:

Prepíš audio nahrávku do textu

Vstupy: URL odkaz na audio nahrávku

Výstupy: text

Vytvor titulky na daný film.

Vstupy: URL na daný film uložený na

Výstupy: súbor s titulkami

3.3 Šprint #2

3.3.1 Zjednotenie Git repozitára

Autor: Bc. Patrik Štrba

V tomto šprinte sme menili architektúru tímového repozitára. Doteraz sme používali všetci jeden spoločný repozitár, ktorý sme upravili z dôvodov bezpečnosti, konzistencie a jednoduchšej údržby. Následne sme prešli na vlastné repozitáre a funkcionality sa pridávala do tímového repozitára pomocou pull-requestov. Týmto sme zabezpečili transparentnosť nad vytvorenýnou funkcionalitou a dosiahli väčšiu kontrolu pridanej funkcionality.

Taktiež je táto nová architektúra vhodnejšia pre tvorbu code-review, kedy si môže každý člen z tímu pozrieť pull-request a schváliť ho, prípadne k nemu zabechať komentáre.

Pri schvaľovaní pull-requestu sme zvolili taktiku, kedy je potrebné pre schválenie pull-requestu vyjadrený súhlas minimálne dvomi členmi a to manažér podpory vývoja a vedúci tímu.

3.3.2 Dokončenie detailov

Autor: Všetci členovia

Nakoľko sa v predchádzajúcom šprinte naplánovalo viac netriviálnych používateľských príbehov, bude potrebné implementácie upraviť. Každý člen tímu pracoval na vlastnej vetve, preto bude tento proces jednoduchší. Tieto úpravy sú nutné z pohľadu konzistentnosti a spolupráce jednotlivých funkcií navzájom.

3.4 Šprint #3

3.4.1 Usporiadanie produktového backlogu

Autor: Bc. Patrik Štrba, Bc. Martin Toma

V piatom šprinte po implementácii niekoľkých základných funkcionalít sme sa zhodli, že by sme potrebovať upratať produktový backlog aby sme vytvorili lepší základ do ďalších šprintov. Stretli sme sa preto špeciálne v neformálnom prostredí s projektovým vedúcim a prebehol brainstorming o možnostiach a smerovaní projektu. Výsledkom bola identifikácia nových features a upratanie existujúcich. Takisto sme sformulovali základné prípady použitia, ktoré je potrebné mať stále v hlavne pri vývoji celej aplikácie.

3.4.2 Vytvorenie metodík

Autor: Všetci členovia

V tíme zastáva každý člen svoju rolu v rámci riadenia. Jednou z jeho povinností je navrhnúť metodiku (jednoznačnú, záväznú, potrebnú), ktorú budú členovia tímu brať ako návod na to, ako správne fungovať. Výstupom sú teda dokumenty obsahujúce pravidlá, ktoré jednoznačne vysvetľujú čo sa bude robiť akým spôsobom a kto je za ne zodpovedný.

Očakávame, že používanie metodík zlepší naše výsledky, nakoľko bude riadenie oveľa jasnejšie a bude mať písomnú formu, ktorá bude prístupná celému tímu.

3.4.3 Revízia doterajších zápisníc

Autor: Bc. Marek Lenčěš

Zistili sme drobné nedostatky pri vypracovávaní zápisníc, preto sme sa na ne viac zamerali, a zapísali všetky dôležité údaje, ktoré sme na stretnutiach preberali. Tiež sme jasne definovali, kto mal v ktorý šprint pridelenú akú úlohu.

3.4.4 Vytvorenie formulára pre vyplnenie profilu

Autor: Bc. Miroslav Molnár

Pre detegovanie vlastností a schopností pracovníka sme sa dohodli, že nám najlepšie pomôže profil. Zadávateľ pomocou týchto informácií môže pre svoju úlohu vybrať pracovníka s potrebnými vlastnosťami a schopnosťami. Z tohto pohľadu profil slúži ako jeden z faktorov vplývajúci na zvýšenie kvality práce v našom systéme.

3.4.5 Nasadenie produkčného servera

Autor: Bc. Patrik Štrba

V tomto šprinte sme dostali priradené časti subdomén a preto sme mohli vytvoriť už funkčnú verziu projektu prezentovateľnú aj ostatným členom v tíme a verejnosti.

3.4.6 Získanie informácií prostredníctvom IP používateľa

Autor: Bc. Miroslav Vojtuš

Analýza

Pomocou predvyplnenia základných informácií môžeme používateľovi uľahčiť registráciu. Z IP adresy vieme získať odhadom krajinu a kraj používateľa. Existujú dostupné implementácie ktoré sa líšia v presnosti odhadu. Máme možnosť využiť externé API rozhrania ktoré ponúka napríklad služba IP info db alebo GeoIP od MaxMind Inc., prípadne vykonať vyhľadávanie nad lokálnou db pomocou vlastných prostriedkov.

Špecifikácia

Funkcionálne:

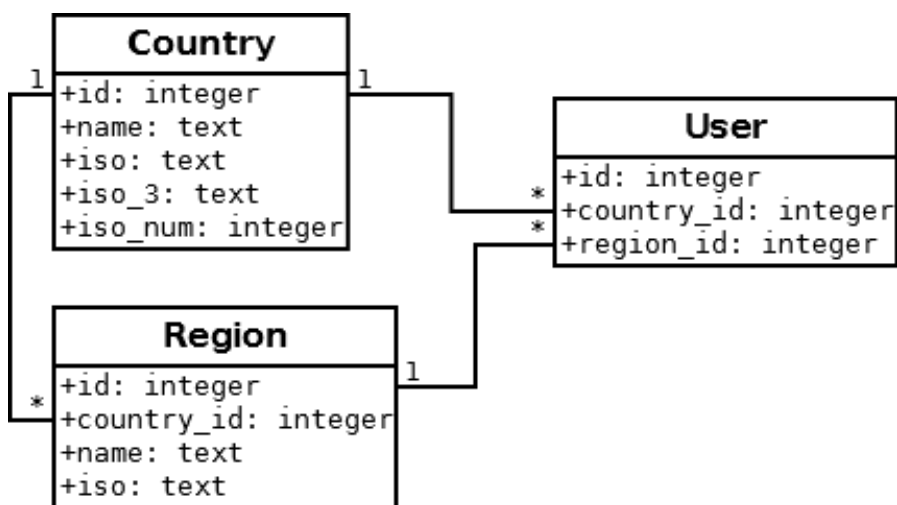
- predvyplnenie polí “country” a “region” automaticky na základe IP

Nefunkcionálne:

- spoľahlivosť - maximalizovať dostupnosť služby
- použiteľnosť - informácie musia byť vyplnené v reálnom čase

Návrh

Dátový model:



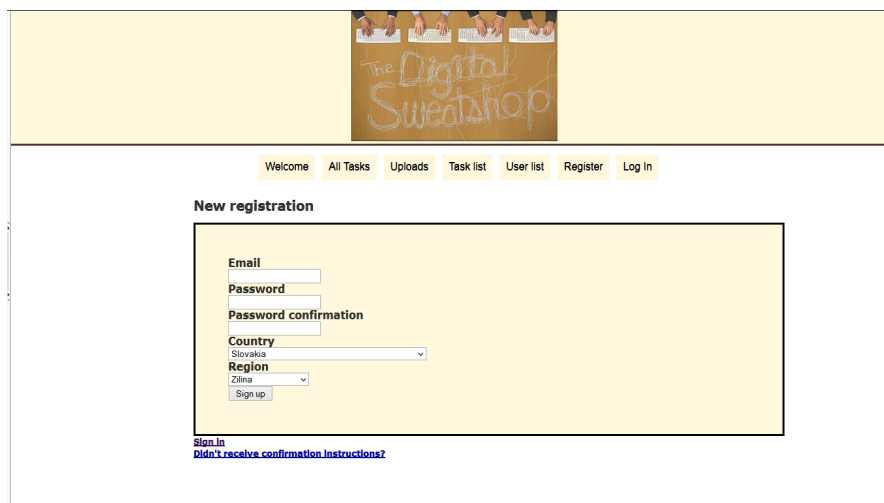
Obr. 8: Dátový model

Implementácia a testovanie

V implementácii sme sa rozhodli využiť lokálnu databázu nad ktorou vykonáme prehľadávanie pomocou rails gemu *geoip*.

Všetky dátové entity sú ukladané v lokálnej databáze. Gem *geoip* vykonáva operácie nad vlastnou lokálnou databázou. Obsah tejto je synchronizovaný pri zavádzaní s databázou nášho systému. Používateľské rozhranie: Pri registrácii má používateľ dopredne vyplnené polia získateľné pomocou služby GeoIp - Country, Region.

Nakoľko je na implementáciu použitý externý gem, nie sú vykonané žiadne automatické testy. Integrovaný test bol vykonaný manuálne pre overenie kvality použitej implementácie. Polia sú v lokálnych podmienkach vyplnené v reálnom čase do 0,5 sekundy. Odhad je presný na krajinu, región je pre slovenskú republiku nepresný.



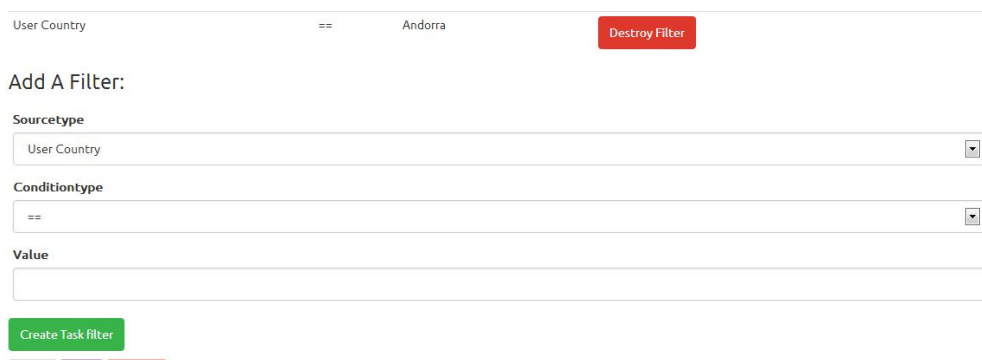
Obr. 9: Používateľské rozhranie

3.5 Šprint #4

3.5.1 Vytvorenie filtrov a kreditového systému

Autor: Bc. Filip Bednárík

Ak chce zadávateľ určiť pre koho sú úlohy určené potrebuje filtrovať pracovníkov, ktorí môžu úlohu ktorú zadal riešiť. Preto sme sa rozhodli v tomto šprinte pridať funkciu správy filtrov pre zadanú úlohu. Filtre je možné zadávať až po vytvorení úlohy. Jednoducho v grafickom rozhraní. Jednotlivé filtre sa skladajú z troch častí: SourceType, ConditionType a Value.



Obr. 10: Pridávanie filtrov

Source type je nejaká vlastnosť používateľa alebo niečo čím sa odlišuje od ostatných. Momentálne sú implementované základné typy zdroja filtra a to sú:

- User Country (Krajina používateľa)
- User Solved Tasks (Množina všetkých používateľom vytvorených úloh)

ConditionType je operátor podmienky. Momentálne sú implementované nasledujúce základné typy podmienok:

- == (Zdroj sa musí rovnať hodnote)
- != (Zdroj sa nesmie rovnať hodnote)

- > (Zdroj musí byť väčší ako hodnota)
- < (Zdroj musí byť menší ako hodnota)
- >= (Zdroj musí byť väčší alebo rovný ako hodnota)
- <= (Zdroj musí byť menší alebo rovný ako hodnota)
- includes (Zdroj ako množina musí obsahovať hodnotu)

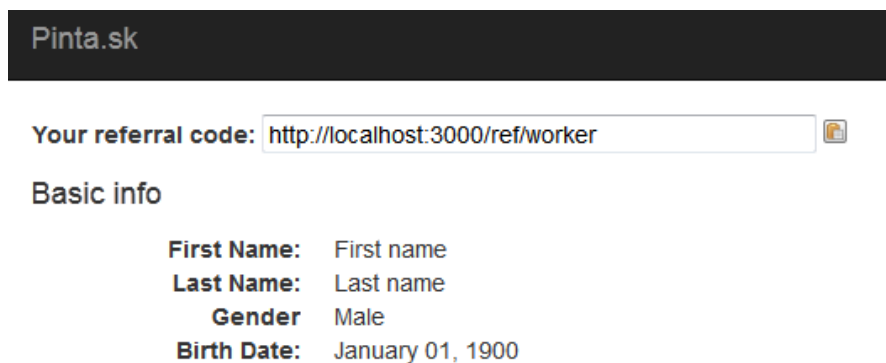
Value je akákoľvek platná hodnota.

Jednou zo základných funkcií našej aplikácie je kreditný systém. Vďaka kreditnému systému totiž môžeme odmeňovať pracovníkov a spravovať zdroje zadávateľov. Preto sme v tomto šprinte implementovali základný kreditný systém, ktorý zahŕňa počet kreditov pre každého používateľa, výšku odmeny za vyriešenie jednej podúlohy a počet kreditov, ktoré je ochotný zadávateľ minúť na jednu úlohu. Pri vyriešení podúlohy pracovníkom sa odpočíta odmena za vyriešenie podúlohy z celkového rozpočtu pre danú úlohu a tieto kredity sa pripočítajú na pracovníkov účet. Museli sme upraviť aj zobrazenie zoznamu úloh pre pracovníkov aby sme zamedzili riešeniu úloh, ktoré už nemajú rozpočet.

3.5.2 Vytvorenie referral systému

Autor: Bc. Marek Lenčes

Po aplikovaní kreditného systému sme zvážili možnosť využitia referral odkazov pre získanie čo najväčšieho množstva nových pracovníkov, ale aj zadávateľov. Vďaka referral odkazom je možné nie len získať nových pracovníkov, ale tých ktorí tento referral odkaz poskytnú, môžeme odmeniť nejakou výškou kreditu. Referral odkaz v profile používateľa je znázornený na Obrázku 11



Obr. 11: Referral odkazy

3.5.3 Vytvorenie profilovej stránky

Autor: Bc. Miroslav Molnár

Profil je jednou z najdôležitejších častí celého nášho systému. Zastrešuje všetky informácie o používateľoch. Tieto informácie sa následne používajú pri identifikácii cieľovej skupiny pracovníkov pre konkrétne zadanie. Vďaka kvalitným informáciám z profilu sa znižuje riziko zlého vypracovania úlohy a tak sa systém stáva efektívnejší a stabilnejší. To je cieľom našej aplikácie, poskytnúť používateľom kvalitné služby.

Analýza

Pre profil nie sú vhodné všetky informácie o užívateľovi. Boli detegované a vybraté hlavné dáta do profilu:

- Firstname - meno užívateľa
- Lastname - priezvisko užívateľa
- Birth_date- dátum narodenia užívateľa
- Gender - pohlavie užívateľa
- User_description- textový popis užívateľa
- Twitter_pin- pin užívateľa na užívateľský účet užívateľa na twitter.

Boli detekované aj ďalšie dáta ktoré sú vhodné pre ich umiestnenie do profilu:

- Language_skills – jazykové znalosti
- Computer_skills - počítačové znalosti

Špecifikácia požiadaviek

Vytvorte profil užívateľa s dátami:

- meno
- priezvisko
- dátum narodenia
- pohlavie

Doimplementujte výpočet:

- Vek používateľa
- Počet spravených úloh
- Percentuálne hodnotenie spravených úloh

Návrh

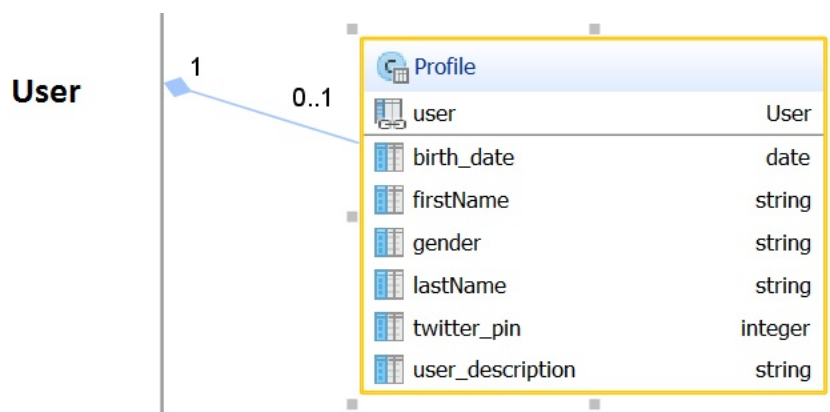
Bol navrhnutý dátový model popísaný na obrázku nižšie, ktorý zobrazuje aktuálne dáta v databáze.

Implementácia a testovanie

Profil je jednoznačne prepojený s používateľom. Pri vytváraní nového používateľa sa vytvára aj jeho profil s predvolenými dátami. Po zmazaní používateľa sa zmaže aj profil na neho naviazaný.

3.5.4 Integrácia so sociálnou sieťou Facebook

Autor: Bc. Patrik Štrba



Obr. 12:

Analýza

V dnešnom svete existuje veľké množstvo sociálnych sietí. Sociálne siete sú veľkým zdrojom dát, ale aj veľmi vhodné na propagáciu a zviditeľnenie aplikácie, či jej produktov.

Špecifikácia požiadaviek

Potreba integrácie so sociálnymi sieťami vzniká ako potreba integrácie sociálnych prvkov a prilákanie nových potenciálnych používateľov touto cestou.

Návrh

Aplikácia bude schopná komunikovať cez rozhranie so sociálnou sieťou facebook, kde bude použitý autorizačný mechanizmus OAuth. Pomocou tohto autorizačného mechanizmu spárujeme používateľa aby sme mohli získať o ňom údaje, ktoré budeme potrebovať v aplikácii. Používateľ bude mať taktiež možnosť sa zapojiť do verejnej diskusie, ktorá bude fungovať pre všetkých, ktorí majú vytvorený účet na tejto sociálnej sieti. Používateľ bude taktiež mať možnosť vyjadriť jeho spokojnosť s danou úlohou, prípadne s inou aktivitou či produktom.

Implementácia a testovanie

Funkcionalita je implementovaná pomocou gemu Koala. Tento gem zabezpečuje komunikácie a tak je možné abstrahovať a zjednodušiť prácu pri integrácii. V rámci implementácie sa implementovala funkcionality na komunikáciu v diskusiách, ktoré boli pridelené jednotlivým úlohám. Taktiež bola implementovaná možnosť označovať jednotlivé úlohy používateľom, že sa im páčia.

3.5.5 Integrácia so sociálnou sieťou LinkedIn

Autor: Bc. Martin Toma

Analýza

V našom projekte je veľmi dôležité pracovať s relevantnými základnými údajmi o používateľoch. V aplikácii bude možnosť definovať úlohy pre špecifické pohlavie, vek, vzdelanie, schopnosti a znalosti. Preto sme hľadali spôsob ako získať tieto

údaje. Dôležité však je aby používateľ nemusel pracne vyplňovať tieto údaje. Sociálna sieť LinkedIn momentálne zažíva vrchol popularity. Automaticky sa sňou spája pracovný život ľudí a teda formálne a relevantné údaje. Integráciou s touto sociálnou sieťou teda používateľovi ušetríme čas a zároveň my získame relevantné údaje.

Špecifikácia požiadaviek

Požiadavkou je aby sme boli schopní základné informácie ako :

- meno a priezvisko,
- dátum narodenia (vek),
- motto,
- zoznam znalostí a pracovné skúsenosti,
- vzdelanie.

Ďalšou požiadavkou je dať používateľovi možnosť vybrať si, či chce aktualizovať konkrétny atribút vo svojom profile.

Návrh

Pomocou LinkedIn API a s využitím autorizačného mechanizmu OAuth spolu s externým gemom LinkedIn zabezpečíme prístup k dátam jednotlivých používateľov. Na tento účel bola na sociálnej sieti LinkedIn vytvorená aplikácia Pinta.sk, s nastavením potrebných oprávnení.

Implementácia a testovanie

V samotnej aplikácii je funkcionality implementovaná pomocou `linkedin_controller` modulu, ktorý má za úlohu komunikovať s LinkedIn API. Súčasťou návrhu je dátový model pre spomínané entity. Je ho možné vidieť na obrázku x. kde je zobrazený aktuálny celkový dátový model aplikácie. Implementovala sa teda funkcionality prepojenia používateľského účtu našej aplikácie s existujúcim účtom na sociálnej sieti LinkedIn. Ďalej bola implementovaná funkcionality aktualizovania profilu pomocou týchto dát. Funkcionality výberu špecifických atribútov sa z časového dôvodu počas šprintu odobrala z backlogu a bude implementovaná v najbližších šprintoch.

3.5.6 Vytvorenie achievement systému

Autor: Bc. Miroslav Vojtuš

Analýza

Za účelom motivácie a zároveň sledovania stavu používateľa sme zvážili ako vhodnú metódu pridelenia statusov na základe dosiahnutých úspechov, vo hernom svete sa nazýva takýto status achievement.

Pre udržanie používateľa v systéme a zabezpečenie pocitu uspokojenia dostáva priebežne po vykonaní akcií nové statusy, za ktoré môže byť v budúcnosti odmenený aj vecne. V systéme môžeme požívať tieto statusy na filtrovanie pracovníkov.

Taktiež slúži na vedenie používateľa systémom. Ukazuje možnosti, ktoré systém ponúka. Časom sa môže báza achievement-ov rozširovať. Základne achivementy sú opísané v nasledujúcej tabuľke:

názov	podmienka	viacnásobný?
apprentice-requester	viac ako 1 zadaná úloha	NIE
journeyman-requester	viac ako 10 zadaných úloh	NIE
expert-requester	viac ako 100 zadaných úloh	NIE
artisan-requester	viac ako 1000 zadaných úloh	NIE
apprentice-worker	viac ako 1 splnená úloha	NIE
journeyman-worker	viac ako 10 splnených úloh	NIE
expert-worker	viac ako 100 splnených úloh	NIE
artisan-worker	viac ako 1000 splnených úloh	NIE
identified-user	vyplnené základné polia v profile	NIE

Špecifikácia požiadaviek

Funkcionálne:

- zobrazenie zoznamu dostupných achievement-ov
- zobrazenie o získaní nového achievement-u

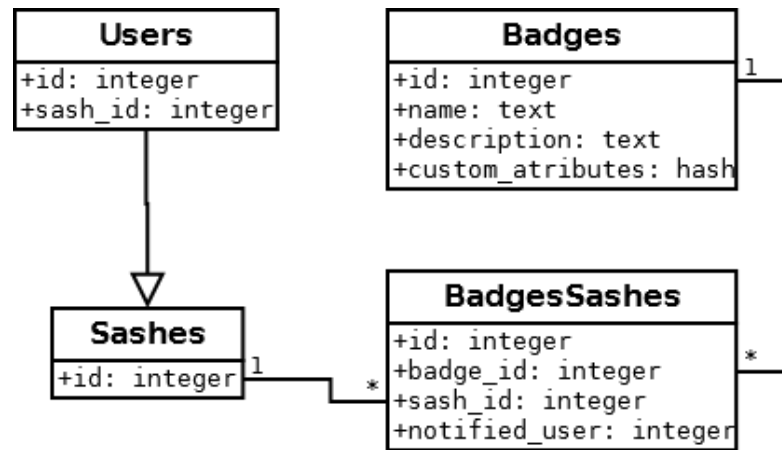
Nefunkcionálne:

- rozširiteľnosť - musí byť jednoduché pridať nový achievement
- rýchlosť - vyhodnotenie achievement-ov nesmie zásadne ovplyvňovať chod systému
- použiteľnosť - nie je to zásadná súčasť systému a achievement-y by nemali používateľa rozptyľovať od úloh. Zobrazenie informácie musí byť prívetivou cestou.

Implementácia a testovanie

Pre implementáciu bol použitý rails gem merit. Ktorý poskytuje základnú funkcionálnosť získavania achievement-ov pod názvom badge. Zároveň ponúka možnosti pridelenia bodov prípadne iných motivačných bonusov.

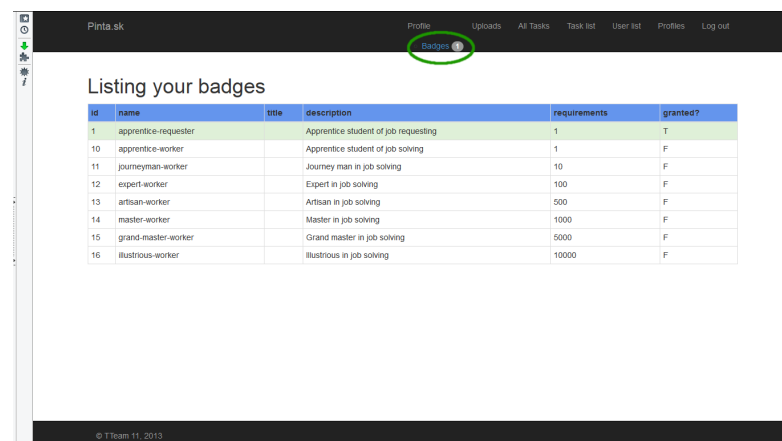
Dátový model pre achievement-y:



Obr. 13: Dátový model pre achievmenty

Všetky dátové entity okrem Badges sú uložené v databáze. Badges sú definované priamo v zdrojových kódach. Logika pridelenia achievementu je viazaná na Badges.name. Nakoľko je ruby interpretovaný jazyk, nie je problém pridať nový achievement spolu s logikou za behu systému.

Používateľské rozhranie:



Obr. 14: Používateľské rozhranie

Používateľské rozhranie bolo implementované s ohľadom na požiadavku použiteľnosti. Informácia o novom achievement-e je zobrazená ako jednoduché počítadlo pri navigačnom uzle prislúchajúcom achievement-om. Po rozkliknutí používateľ vidí stav dostupných achievement-ov.

3.6 Šprint #5

3.6.1 Vytváranie dokumentácie

Autor: Všetci členovia

Všetci členovia sa podieľali na vytváraní a dopĺňaní dokumentácie k riadeniu, aj k inžinierskemu dielu.

3.6.2 Facebook lajky a komentáre

Autor: Bc. Patrik Štrba

V tomto šprinte bolo potrebné spraviť integráciu sociálnych prvkov pre náš projekt. Na základe tímového stretnutia sme sa rozhodli integrovať prvky zo sociálnej siete facebook a to konkrétne tlačidlo “páči sa mi to” a komentáre. Najskôr bolo potrebné vytvoriť aplikáciu na facebooku, pomocou ktorej sa následne integrovali prvky na stránku. Následne sme tieto prvky sme analyzovali a navrhli dizajn a štruktúru čo by bolo pre nás výhodné. Následne sme tieto prvky integrovali do projektu takým spôsobom, aby bolo možné tieto prvky používať aj v budúcnosti jednoduchším spôsobom.

3.6.3 Integrácia s Google účtom

Autor: Bc. Marek Lenčoš

Analýza

Dnes je už vo všetkých významných webových aplikáciách možnosť integrácie účtu webovej aplikácie s inými účtami, napríklad sociálnymi sieťami, z ktorých je možné získať informácie o používateľovi a zjednodušiť mu tak vyplnenie jeho profilu. Významné postavenie na internete majú služby firmy Google, ktoré sú využívané veľkým množstvom používateľov, a sú teda vhodné na získanie údajov o prihlásenom používateľovi.

Špecifikácia požiadaviek

Keďže primárnym cieľom integrácie našej aplikácie s účtom Google je získanie údajov, musíme sa zamerať na údaje, ktoré nás zaujímajú v našej aplikácii, predovšetkým su to osobné údaje o používateľovi, meno, dátum narodenia, profilová fotka. Neskôr možno využijeme ďalšie informácie, ktoré služby Google ponúkajú.

Návrh

Integráciu s účtom Google bude potrebné vytvoriť pomocou nejakého rozhrania, ktoré ďalej umožní získavanie údajov zo služieb Google.

Implementácia a testovanie

Implementácia je vykonaná pomocou rozhrania OAuth, ktoré zabezpečí spárovanie účtu našej webovej aplikácie a účtu Google. Ďalej je možné získať želané údaje, ktoré využijeme pri naplnení profilu používateľa.

3.6.4 Integrácia s Twitter účtom

Autor: Bc. Miroslav Molnár Pre získanie ďalších informácií o používateľovi a možnosť v budúcnosti komunikovať zo sociálnou sieťou, sa navrhla integrácia s používateľským účtom na Twitteri. Ďalšou výhodou je možnosť prihlasovať používateľov cez ich existujúce konto na Twitteri, posielanie správ na sociálnu sieť, alebo sledovať činnosť používateľa. Prepojenie s touto sociálnou sieťou nám umožňuje rýchlejšie rozšírenie našej aplikácie medzi súčasnou generáciou mladých ľudí. Čím viac používateľov našu aplikáciu bude používať, tým bude použiteľnejšia pre všetkých používateľov.

3.7 Šprint #6

3.7.1 Vytvorenie viackrokových formulárov

Autor: Bc. Filip Bednárík

Pri type úlohy Survey (dotazník) sme prišli na to, že potrebujeme viac krokov na to aby sme korektne vedeli zobrazit grafické rozhranie pridávania podúloh pre zadávateľa. Preto sme v tomto šprinte implementovali tzv. Wizard formy a tzv. Nested formy.

Pri každom kroku sa posielajú údaje z formulára na server bez znovunačítania stránky prehliadačom, čo nám dovoľuje použiť pekný efekt posunu formuláru doľava a tiež spríjemňuje prácu s formulárom zadávateľovi. Pri každom načítaní kroku sa berú do úvahy dáta prijaté pri kliknutí na tlačidlo “Next”.

Najprv zadávateľ vyberie kategóriu úlohy podľa čoho sa mu neskôr v kroku 3 zobrazí korektný formulár pre vytvorenie podúloh pre daný typ úlohy.

Obr. 15: Formulár zadávania úlohy

V kroku 2 zadávateľ zadá základné informácie o úlohe. Zadávateľ sa môže posúvať vo formulári aj klávesami doľava a doprava.

Obr. 16: Formulár zadávania úlohy

Tretí krok bol najťažší z implementačného hľadiska. Ide totiž o vhniezdený for-

mulár obsahujúci n záznamov do tabuliek podúloh viažucich sa na hlavnú entitu úlohy. Zadávatel má možnosť bez znovunačítania stránky jednoducho pridávať nové podúlohy, mazať ich alebo ich upravovať.

Obr. 17: Formulár zadávania úlohy

3.7.2 Rebranding - web tímu, dizajn

Autor: Bc. Martin Toma, Bc. Marek Lenčoš

V rámci rebrandingu sme vytvorili nové logo, ktoré zároveň reprezentuje naše nové pomenovanie Pinta.sk. S tým súvisí aj rekonštrukcia tímového webu, ktorý už nevyhovoval novým požiadavkám. Webu sme dali nový dynamický responzívny dizajn, založený na šablóne Bootstrap, ukážku nového webu môžeme vidieť na obr



Obr. 18:

Rovnako sme navrhli a vytvorili nový obsah, ktorý prezentuje náš projekt. Snažili sme sa použiť pokročilejšie copywriting techniky spojené s technikami SEO. Výsledkom je teda nová identita produktu a rovnako aj tímu 11, ktorý od tohto šprintu nesie oficiálne meno Pinta Group a produkt Pinta.sk.

3.7.3 Zlepšená organizácia profilu

Autor: Bc. Miroslav Vojtuš

Analýza

Aktuálna verzia profilu je postačujúca pre vývojové účely. Avšak jej vzhľad a štruktúra nezodpovedá doterajšiemu designu. Je potrebné doplniť možnosť manuálne vyplňať polia v profile, používateľsky prívetivou cestou.

S pohľadu použiteľnosti existujú dve základné cesty:

- synchronný prístup
 - každá akcia jeden kompletný request s návratom kompletného pohľadu
 - ľahké na implementáciu - rails ponúka generátory CRUD operácií
- asynchrónny prístup
 - každá akcia jeden čiastočný dopyt na server s návratom len čísta obnoveného pohľadu
 - náročnejšia implementácia, vyžadujúca viac vlastného kódu
 - ťažšie z pohľadu udržateľnosti

Vzhľadom k tomu, že vyvíjame moderný webový systém, najvhodnejšou variantou je asynchrónny prístup, ktorý vša nesie nevýhodu ťažšej implementácie, náročnejšej na údržbu.

Špecifikácia

Funkcionálne požiadavky:

- zobrazenie profilu v štruktúrovanej forme
- editovanie jednotlivých kategórií
- pridanie novej položky do dynamických kategórií (zoznam)
- odstránenie položky z dynamických štruktúr
- iné operácie ako je čítanie povoliť len majiteľovi profilu

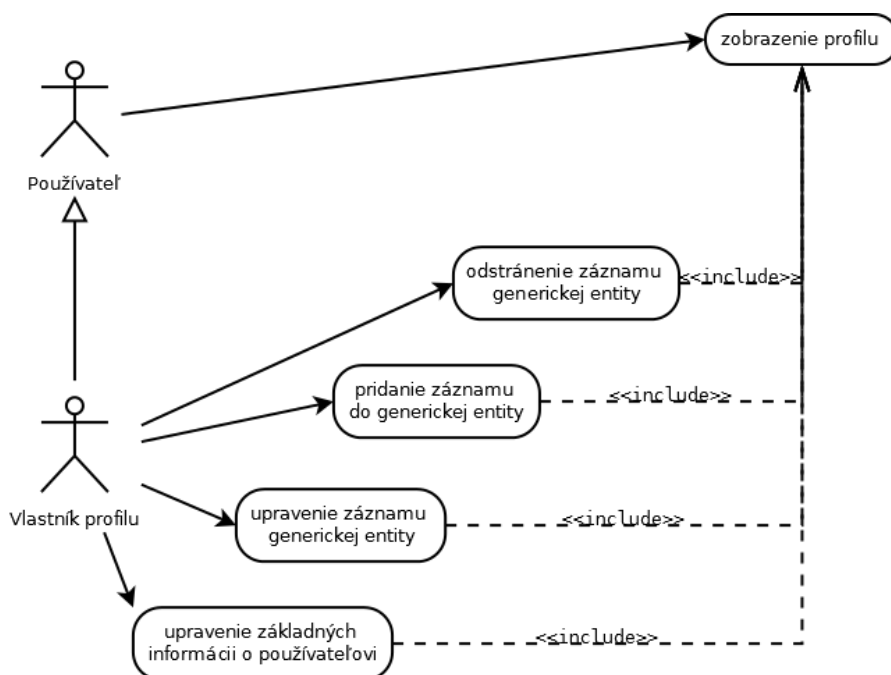
Nefunkcionálne požiadavky:

- grafické prevedenie zladené so štýlom systému
- dostupnosť - minimalizovať čas odozvy systému medzi akciami
- použiteľnosť - jednotlivé akcie musia byť jasne viditeľné

Návrh

Kategórie zostanú zachované podľa dátových entít:

- všeobecné informácie (and. Basic info)
- vzdelanie (and. Education)
- skúsenosti (and. Skills)
- jazyky (and. Languages)



Obr. 19: Diagram prípadov použitia

- zamestnania (ang. Positions)

Dátový model nie je ovplyvnený, zachovaná štruktúra informácií.

Diagram prípadov použitia: Diagram opisuje základné funkcionálne prvky profilu. Generická entita je jedna z dopredne určených entít s dynamickým obsahom. Základné sú odvodené vyššie. Úpravy profilu vyžadujú zobrazenie profilu. Profil si môže zobraziť ktokoľvek ale editovať ho môže len vlastník profilu.

Pre zabezpečenie požiadavky štruktúrovanosti je profil rozdelený do dvoch stĺpcov. Jednotlivé kategórie majú vlastný panel s názvom a informačným obsahom. Podľa typu uložených informácií sa tlačítko s akciou pre úpravu bude nachádzať, buď pri celej kategórii alebo jednotlivej položke. Týmto je zabezpečená požiadavka použiteľnosti - v prípade veľkého množstva záznamov používateľ edituje len ten, ktorý potrebuje.

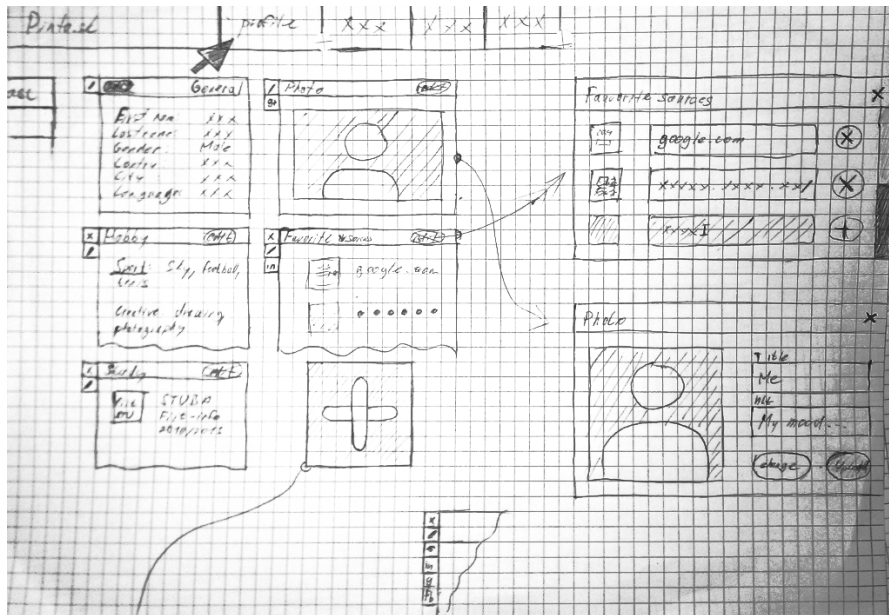
Low fidelity návrh rozloženia funkcionálnych prvkov:

Implementácia a testovanie

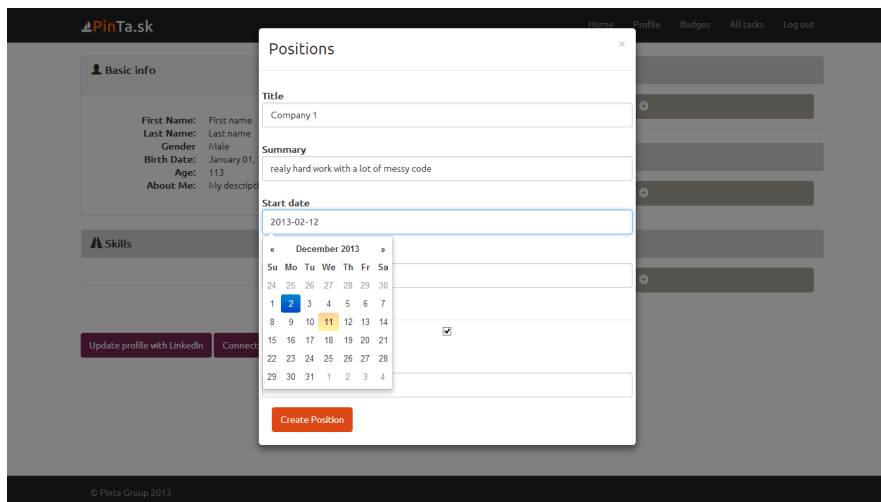
V implementácii je kladený dôraz na použiteľnosť a dostupnosť. Zvolili sme asynchrónnu metódu. Základná štruktúra zodpovedá pôvodnému low fidelity návrhu. Z dôvodu vyššej komplexity niektorých prvkov a nízkej priority boli v implementácii opomenuté. Zamerali sme sa na hlavné funkcionálne prvky a ostatné sme odsunuli do neskorších fáz vývoja.

Rozhranie modálneho okna je zobrazené na nasledujúcom obrázku. Obsah je získaný asynchrónnou požiadavkou na server. Formulár je štandardný pre rails-ovské aplikácie s podporou asynchrónneho odovzdania dát. Modálne okno je ako komponent abstrahované pre ďalšie použitie v aplikácii.

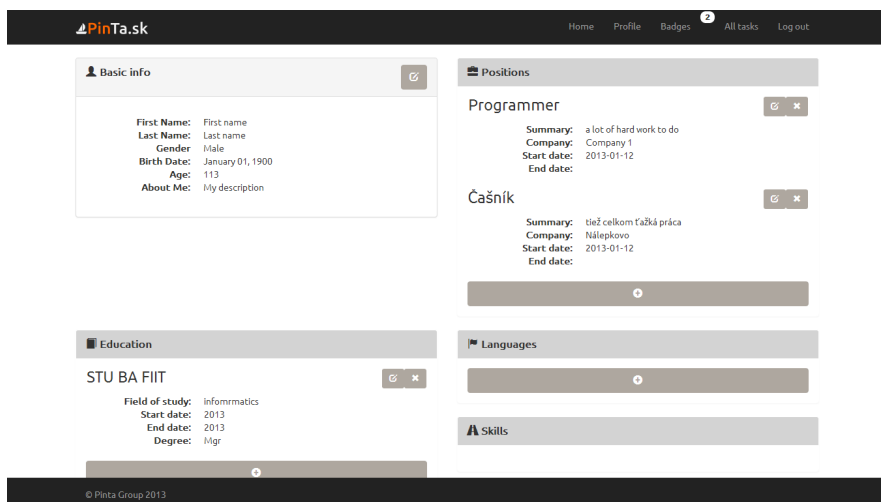
Výsledná štruktúra profilu je zobrazená na nasledujúcom obrázku. Rozdiel medzi pohľadom na cudzí profil a vlastný je v absencii ovládacích prvkov pre pridanie a úpravu odstránenie informácií.



Obr. 20: Low fidelity návrh



Obr. 21: Modálne okno



Obr. 22: Návrh nového profilu

Na základe požiadavky zachovať štýl celého systému, boli použité triedy štýlov definované použitým GUI frameworkom Bootstrap.

Z dôvodu náročnosti automatizovaného testovania asynchrónneho vykonávania boli jednotlivé prvky testované len manuálne.

3.8 Šprint #7

3.8.1 Zarovnanie dizajnu produktovej aplikácie s novou identitou tímu

Autor: Bc. Martin Toma

V minulom týždni prebehol rebranding a navrhli sme produktové logo, tímové meno a hlavné produktové farby. Keďže aplikácia mala v tej dobe iba mierne upravený základný bootstrap dizajn, rozhodli sme sa zaviesť miernu konzistenciu aj v tomto ohľade.

Vytvoril sa teda nový CSS dizajn a použili sa pokročilejšie bootstrap techniky a hlavne sa aplikovali farby a fonty navrhnuté v minulom šprinte pri rebrandingu.

3.8.2 Survey task

Autor: Bc. Filip Bednárík

Ďalším user story, ktorý sme riešili bol user story v ktorom zadávateľ vyžadoval od pracovníkov vyplňovať dotazníky. Preto sme v tomto šprinte implementovali novú kategóriu úloh a tou je Survey (dotazník). Zmenili sme atribúty podúloh a to tak, že sme pridali typ vstupu a typ výstupu. Typ vstupu môže byť v súčasnosti:

- Obrázok z URL
- Text
- Facebook stránka

Dostupné typy výstupu:

- Text
- Číslo
- Možnosti (oddelené čiarkou)

Zadávateľ má možnosť vyskladať si vlastný dotazník z týchto elementárnych otázok. V ďalších šprintoch som sa venoval zmene GUI CSS atď. Searchbox som spravil aby fungoval AJAXovo spolu s willpaginate.

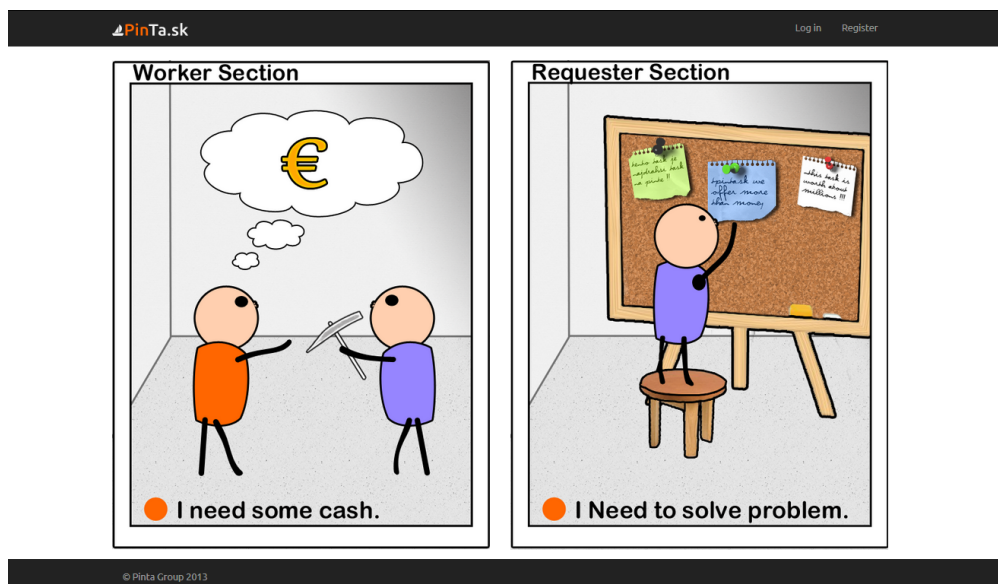
3.8.3 Vytvorenie grafických návodov pre landing page

Autor: Bc. Marek Lenčoš

Na úvodnej stránke aplikácie sa pri návšteve objaví rozcestník pre pracovníka, alebo pre zadávateľa úlohy, podľa výberu sa mu zobrazí jednoduchý obrázkový návod, ako asi funguje vypracovanie/zadanie úloh v systéme Pinta.sk.

3.8.4 Integrácia so sociálnou sieťou Twitter

Autor: Bc. Miroslav Molnár



Obr. 23: Výber sekcie na landing page

Analýza

Pre možnosť získania viacerých zdrojov informácií o používateľovi nášho systému sme vytvorili integráciu so sociálnou sieťou Twitter. Druhým významom je biznis logika, pomocou ktorej sa snažíme získať väčšie množstvo používateľov. Čím viac používateľov systém bude používať, tým bude väčšia pravdepodobnosť získania kvalitnejších výsledkov a väčšie množstvo úloh.

Špecifikácia požadaviek

Vytvorenie integrácie používateľa s jeho účtom na sociálnej sieti Twitter. Používateľ v profile pomocou tlačidla bude môcť sa zintegrovat' so sociálnou sieťou. Po úspešnom prepojení s účtom na Twittri, bude môcť používateľ pokračovať v činnosti na webovej stránke projektu.

Návrh

Po dôkladnom preštudovaní možností prepojenia so sociálnou sieťou treba vybrať tú najbezpečnejšiu pre používateľa a zároveň najefektívnejšiu pre náš projekt. Je potrebné dbať na obmedzujúcu podmienku času, vyhradeného na vývoj. Vytvorené tlačidlo v profile webovej aplikácie projektu po stlačení presmeruje používateľa na stránku sociálnej siete Twitter. Tu sa prihlási pomocou svojich prihlasovacích údajov, ktoré má na sociálnej sieti. Po úspešnej registrácii, bude používateľ presmerovaný na webovú stránku v profile, kde bude uvedený o úspešnej, alebo neúspešnej integrácii s účtom.

Pre umožnenie integrácie používateľa s našou webovou stránkou projektu je nevyhnutné na stránke Twittru <https://dev.twitter.com/>, registrovať našu aplikáciu. Pre registráciu je nevyhnutné mať účet na sociálnej sieti Twitter. Po úspešnej registrácii aplikácie, budú vygenerované špeciálne hash kľúče slúžiace na identifikáciu pri komunikácii s Twittrom

Implementácia a testovanie

Do profilu bolo doimplementované tlačidlo s funkcionalitou pre komunikáciu s Twittrom. Zvolili sa OAuth a Twitter gemy. Pomocou týchto gemov sa zaoberá komunikácia so sociálnou sieťou. Pre naviazanie komunikácie sa využívajú hash kľúče vygenerované pri registrácii našej aplikácii na stránke Twitter. Po naviazaní komunikácie nám Twitter pošle request token, z ktorého si získame dáta potrebné na presmerovanie používateľa z našej stránky na stránku Twitter. Po úspešnom prihlásení presmeruje používateľa na našu stránku. V parametri URL, server Twittru, odovzdá vygenerované hash kľúče integrujúceho používateľa. Tieto kľúče uložíme do tabuľky profilu používateľa. Bola implementovaná metóda `show_twitter_profile` v kontroleri profilu, ktorá využije hash kľúče používateľa už integrovaného s účtom na Twittri a získa dáta z jeho profilu. Funkcia má ukazovať správnu komunikáciu so sociálnou sieťou a spracovanie prijatých údajov.

3.9 Šprint #8

3.9.1 Zmena dizajnu badges

Autor: Bc. Miroslav Molnár

Analýza

Jednou z možností ako motivovať pracovníka a zadávateľa ku kvalitnejšej práci v našom systéme je dostávanie badges. Badges sa dostávajú za špeciálne procesy splnené v systéme. Takýto štýl motivácie je veľmi prínosný a lákavý pre používateľov. Pre jeho zlepšenie bola navrhnutá jeho grafická úprava.

Špecifikácia požiadaviek

Navrhnite lepšiu grafickú reprezentáciu badges v systéme. Použiť už implementovaný zdrojový kód. Navrhnite ďalšie možné typy badges.

Návrh

Pre reprezentáciu badges bola navrhnutá grafická podoba v tvare guľatých lesklých odznakov, farebne rozlíšených pre pracovníka a zadávateľa. Pri nedosiahnutí badges sa zobrazia tieto grafické podobizne čiernobielo Obr.1 V pôvodnom návrhu bola tabuľka so všetkými badges a informáciami o nich. Tento typ bol pre používateľa hneď po prvom zobrazení veľmi chaotický. Prílišné množstvo informácií spôsobovalo negatívnu dezorientáciu.

Z tohto dôvodu bol význam dát prehodnotený a nepotrebné informácie sa nezobrazujú používateľovi, ako je napríklad ID badges. Ostatné informácie sa zobrazia až keď si ich používateľ vyžiada. Týmto prístupom sa stalo grafické rozhranie intuitívnejšie a príjemnejšie pre používateľa. Príklad už upraveného užívateľského rozhrania môžeme vidieť na Obrázku 25

Celková reprezentácia dát ostala v tabuľke. Bola však vytvorená nová šablóna „.css“ súboru, ktorá nemá zbytočné čiary a tak je voľnejšia. Pre lepšiu čitateľnosť sa riadky farebne odlišujú. V tabuľke sa zobrazuje iba názov badges, pre bližšie informácie používateľ musí kliknúť na tlačidlo nachádzajúce sa napravo. Po stlačení tlačidla sa zobrazí modálne okno so všetkými informáciami o konkrétnom badge.



Obr. 24: Zobrazenie grafickej reprezentácie badges

Implementácia

Reprezentácia obrázkov bola nakreslená a zmodifikovaná do rôznych farieb. Modálne okno bolo prevzaté z profilu a upravené pre potreby použitia. Boli doimplementované potrebné preklady jazykov. Bol doplnený parameter worker typu boolean pre detekciu či sa jedná o badge pre zadávateľa, alebo pracovníka. Výsledná schéma bola otestovaná.

3.9.2 Vytváranie záznamov z externej aplikácie

Autor: Bc. Filip Bednárík

Analýza

Pri user story, v ktorom zadávateľ vytvorí úlohu, kde vyžaduje od pracovníka nainštalovanie nejakej aplikácie, ktorú vyvinul a jej používanie je vhodné aby mal možnosť zadávateľ sledovať pracovníkovu aktivitu v jeho aplikácii, odmeňovať ho za používanie, prípadne mu prístupňovať nové úlohy na základe jeho aktivity.

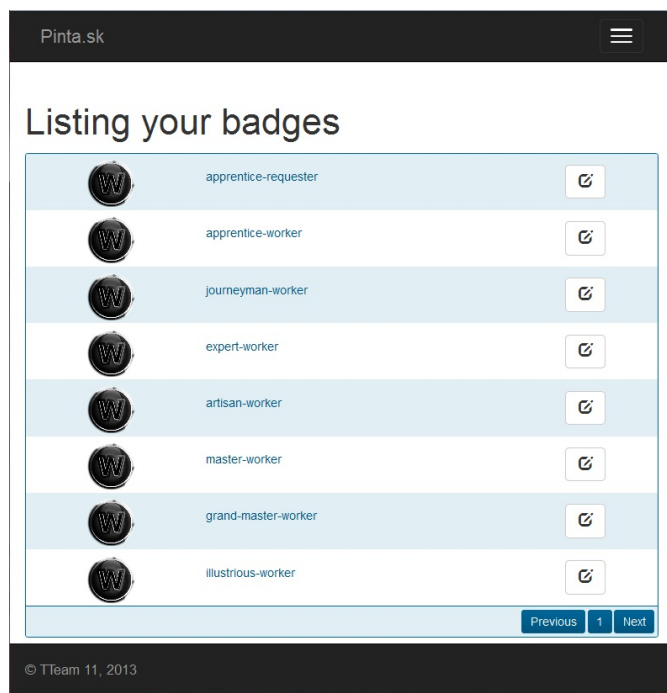
Špecifikácia požiadaviek

Pre tento účel potrebujeme API pre logovanie (vytváranie záznamov) z externej aplikácie do nášho systému. Logovanie musí prebiehať čo najjednoduchšie zavolaním URL s parametrami. V parametroch musí byť identifikovateľný zadávateľ prípadne úloha ku ktorej sa záznam vzťahuje, pracovník, ktorý používa danú aplikáciu, nejaká značka podľa ktorej si zadávateľ vie filtrovať záznamy a nakoniec samotná hodnota záznamu.

Návrh

Navrhlí sme jednoduchý CRUD, kde môže zadávateľ svoje záznamy spravovať. Taktiež sme navrhli formát adresy ktorú zadávateľ zavolá zo svojej externej aplikácie pre vytvorenie záznamu.

`http://www.pinta.sk/log/*API-KEY*/*USER-TAG*/*TAG*/*VALUE*`



Obr. 25: Zobrazenie stránky s badges

Listing Logs - API Key: 31b8face-08c7-4845-9ef7-b01c51816c0d

Tag	Value	User Tag	Actions
tag1	value	drndos	Edit Destroy
TaG	sadhsadhasdsad	drndos	Edit Destroy
TaG	sadhsadhasdsad	drndos	Edit Destroy
TaG	sadhsadhasdsad	drndos	Edit Destroy

New Log

Obr. 26: CRUD

- ***API-KEY*** je jedinečný kľúč zadávateľa do nášho systému, ktorý sa mu vygeneruje akonáhle prisúpy na stránku záznamov.
- ***USER-TAG*** je identifikátor používateľa systému
- ***TAG*** je značka, podľa ktorej si vie zadávateľ záznamy filtrovať
- ***VALUE*** je samotná hodnota záznamu

Implementácia a testovanie

Pri implementácii sme využili generátor RoR pre CRUD. Nastavili sme routes tak aby pri navštívení URL daného formátu sa vyvolal controller pre vytvorenie nového záznamu z URL.

Pri generovaní API sme využili generátor jedinečného kľúča poskytovaný frameworkom.

3.9.3 Úloha s pridávaním tzv. Facebook lajkov

Autor: Bc. Róbert Černý

Analýza

Jednou z najnavštevovanejších sociálnych sietí v súčasnosti je Facebook. Okrem regulárnych používateľov sa na tejto sieti nachádzajú aj stránky, ktoré prezentujú spoločnosti, významné osobnosti, médiá a iné skupiny, ktoré bojujú o zvyditeľnenie a popularitu. Tá sa meria počtom hlasov (tzv. lajk). Automatizovanie procesu pridávania hlasov k želaným stránkam na facebooku je ďalšou možnosťou ktorú náš systém poskytne. Nakoľko je Facebook rozšírený, predpokladá sa, že táto úloha bude veľmi obľúbená nakoľko je nenáročná a jednoduchá.

Špecifikácia požiadaviek

Vytvorenie novej kategórie úlohy Facebook lajk, prostredníctvom ktorej automatizujeme proces hlasovania na sociálnej sieti Facebook. Tento proces vyžaduje od pracovníka iba jednoduché kliknutie na tlačidlo lajku konkrétnej stránky.

Návrh

Zadávateľ úlohy poskytne zoznam URL stránok pre lajkovanie. Zobrazenie úlohy pre pracovníka je veľmi jednoduché. Jednotlivé URL sú zobrazené v prehľadnej tabuľke pričom je ku každej priradené tlačidlo pre hlasovanie. Pracovník vždy vie, či za danú stránku už hlasoval. Táto úloha prebieha výlučne na našej stránke, nieje teda potrebné navštíviť jednotlivu každú URL zo zoznamu. Pre implementovanie tejto úlohy je nutné priradiť pracovníkovi autorizačný token, aby bolo jasné, kto hlasuje. Túto funkcionality rieši integrácia so sociálnou sieťou Facebook (kapitola ...). Po obdržaní autorizácie s príslušnými privilégiami máme možnosť používať Graph API, ktoré poskytuje facebook. Pomocou tohto API dokážeme dopytovať servery sociálnej siete a získavať tak dáta o pracovníkovi, ktorý práve hlasuje. Pre optimalizáciu a rýchlosť budeme využívať FQL (Facebook Query Language). Úloha skončí úspešne, ak bude každá zo stránok lajknutá.

Implementácia a testovanie

Pre úlohu vytvoríme možnosť novej kategórie “facebook_like”. Pri implementácii formulára pre novú úlohu vytvoríme možnosť pre vkladanie jednotlivých URL, čo bude jediná špecifická povinnosť zadávateľa úlohy. Príkladom takejto URL je: <https://www.facebook.com/amazing.weird>.

Jednotlivé tlačidlá pre lajkovanie je možné implementovať pomocou jednoduchého HTML kódu kde premenná “url” reprezentuje URL stránky.

```

1 <div class="fb-like"
2   data-href="<%= url %>"
3   data-layout="button_count"
4   data-action="like"
5   data-show-faces="false"
6   data-share="false">
7 </div>

```

Metóda “pages_liked?” kontroluje, či boli všetky stránky lajknuté.

Výsledkom bude nová inštancia triedy “solution”, ktorá má v atribúte “data” celkový počet lajknutých stránok nakoľko iný výstup úloha nepožaduje.

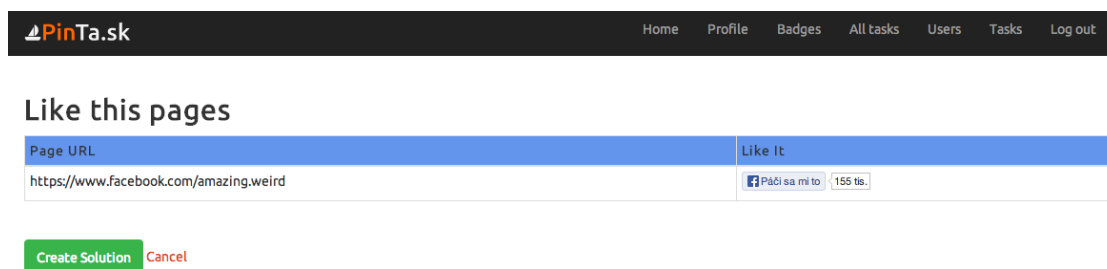
Chyby ktoré môžu v tomto procese nastať zachytávame pomocou výnimiek, ktoré ošetrujeme. Medzi ne patria:

1. pracovník nieje prepojený s facebookom, teda nemá autorizačný token
2. niektorá zo stránok neexistuje
3. pracovník nelajkol všetky stránky

Takýmto jednoduchým dopytom zistíme, či pracovník lajkol stránku.

```
SELECT page_id FROM page_fan WHERE uid=me() AND page_id=page_act['id']
```

Výberáme teda identifikátor stránky z tabuľky “page_fan” ktorá patrí aktuálnemu pracovníkovi. Musí platiť podmienka, že identifikátor stránky z tabuľky sa zhoduje s identifikátorom stránky z úlohy. V prípade, že pracovník lajkol stránku, výsledný dopyt vráti hodnotu identifikátora stránky. Tento proces opakujeme pre každú stránku z čoho vieme určiť správnosť vypracovania úlohy.



Obr. 27: Lajkovanie stránok

3.9.4 Intenacionalizácia (SK, ENG)

Autor: Bc. Róbert Černý

Analýza

Žiadna webová aplikácia používaná medzinárodne sa nezaobíde bez jazykových prekladov. Identifikovali sme požiadavku pre implementáciu frameworku, ktorý bude poskytovať možnosť jednoduchým spôsobom meniť jazyk aplikácie. Pre účely testovania sme sa rozhodli pre jazyky slovenčina a angličtina.

Špecifikácia

Základnou požiadavkou bolo, sprístupniť používateľovi možnosť meniť jazyk aplikácie. Táto funkcionality sa bude nachádzať priamo na úvodnej stránke po prihlásení. Kliknutím na zástavku sa automaticky zmení jazyk bez nutnosti akéhokoľvek iného úkonu. Požiadavkou samotného vývojového tímu bolo zdefinovať čo najjednoduchší spôsob ako pri programovaní definovať reťazec znakov určený na preklad. Najefektívnejším spôsobom bude používať volanie metódy. Príkladom definovania takéhoto reťazca je: `f('tento text sa prekladá')`. Takýmto spôsobom ušetríme čas a zefektívime proces vývoja.

Implementácia a testovanie

Súčasťou Ruby on Rails je API L18n, ktoré nám dovoľuje jednoducho a efektívne meniť jazyk. Jazyk je vhodné nastaviť v súbore `application_controller.rb` z ktorého dedia všetky ostatné kontroleri. Samotný jazykový preklad sa nachádza v súboroch `en.yml` a `sk.yml` so špecifickou syntaxou. Výraz “Description” by sme zadefinovali takto:

V súbore `en.yml` pridali:

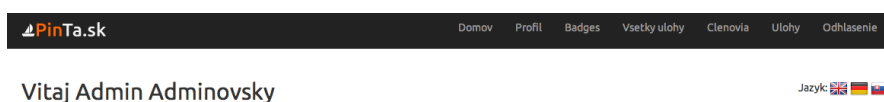
```
1 description: Description
2           sk.yml pridali
3 description: Popis
```

V zdrojovom kóde stačí je možné použiť túto syntax a jazykové API automaticky substituujú výraz za správnu hodnotu, podľa nastaveného jazyka `<%=t "description"%>`

Zmena jazyka prebieha pomocou kontrolera `locale.rb`, ktorý umožňuje zmenu jazyka po kliknutí na zástavku.



Obr. 28: Substitúcia pri anglickom jazyku



Obr. 29: Substitúcia pri slovenskom jazyku

3.9.5 Zobrazovanie výsledkov

Autor: Bc. Róbert Černý

Analýza

Zobrazovanie a vyhodnocovanie výsledkov pracovníkov je dôležitou súčasťou systému pre sprostredkovanie práce. Zadávatelia úloh prichádzajú do systému aby obdržali relevantné výstupy od pracovníkov. Tieto údaje budú prístupné zadávateľovi úloh, avšak okrem toho je potrebné ich štatisticky vyhodnocovať. Zadávatelovi tak pribudne možnosť prezerat výsledky svojich úloh. Prvou úlohou je vytvoriť stránku s diagramom výsledkov pre úlohu dotazník, ktorá bude prístupná zadávateľovi danej úlohy.

Špecifikácia

Pre dotazník sa vyhodnotí každá otázka oddelene. Znamená to, že na stránke pre vyhodnotenie bude rovnaký počet diagramov ako bolo otázok. Zvolíme koláčový diagram, ktorý je na tento typ výsledkov najvhodnejší. Názov diagramu bude znenie otázky a jednotlivé odpovede budú zobrazené v legende. Každú odpoveď bude znázorňovať jedna farba. Po kliknutí na úsek v diagrame sa nám zobrazia informácie o počte hlasov a percentuálnom podiele, ktoré daná odpoveď získala. Takýmto spôsobom vytvoríme jednoduchý framework pre zobrazovanie výsledkov.

Implementácia

Pri implementácii bol použitý gem Highchart, ktorý ponúka diagramy rôznych typov v graficky zaujímavom prevedení. Za účelom zobrazovania výsledkov bol vytvorený kontroler, ktorý sa bude v budúcnosti rozširovať o ďalšie štatistické zobrazenia prípadne filtrovanie výsledkov podľa atribútov pracovníkov. Diagramy je možné definovať pomocou jQuery takto:

```

1 <script>
2     $(function () {
3         var chart;
4         $(document).ready(function () {
5             // Build the chart
6             $('#<%= element.__id__ %>').highcharts({
7                 chart: {},
8                 title: {},
9                 tooltip: {},
10                plotOptions: {
11                    pie: {}
12                },
13                series: [
14                    {
15                        type : ""
16                        name: ""
17                            data: []
18                    }
19                ]
20            });
21        });
22    });
23 </script>

```

V tomto skripte je potrebné doplniť typ diagramu, ktorý chceme zobraziť, znenie otázky a dataset obsahujúci dvojicu hodnôt - (odpoveď,počet). Tento dataset sa naplní v samotnom kontrolery, kde sa pre každú podúlohu danej úlohy vytvorí pomocný dataset s informáciami o znení otázky, možných odpovediach, pričom sa pre každú odpoveď pomocou dopytu vyhľadajú hlasy a uchová sa ich počet. Výsledný dataset vyzerá nasledovne:

„question“=> „Aky mobil používate?“, „answers“=>[„Nokia“, „Samsung“, „HTC“, „Sony Ericsson“, „iPhone“], „Nokia“=>0, „Samsung“=>0, „HTC“=>1, „Sony Ericsson“=>2, „iPhone“=>0

„question“=> „Aky browser používate?“, „answers“=>[„Firefox“, „Chrome“, „IE“, „Safari“], „Firefox“=>1, „Chrome“=>1, „IE“=>0, „Safari“=>2

Po otvorení stránky pre výsledky úlohy sa zobrazia tieto diagramy. Údaje korešpondujú s datasetom, ktorý je spomínaný vyššie. Tento prístup zabezpečuje jednoduché pridávanie nových diagramov a štatistických zobrazení.



Obr. 30: Grafy z gemu Highchart

3.10 Šprint #9

3.10.1 Vytvorenie survey kampane

Autori: Bc. Róbert Černý, Bc. Filip Bednárík

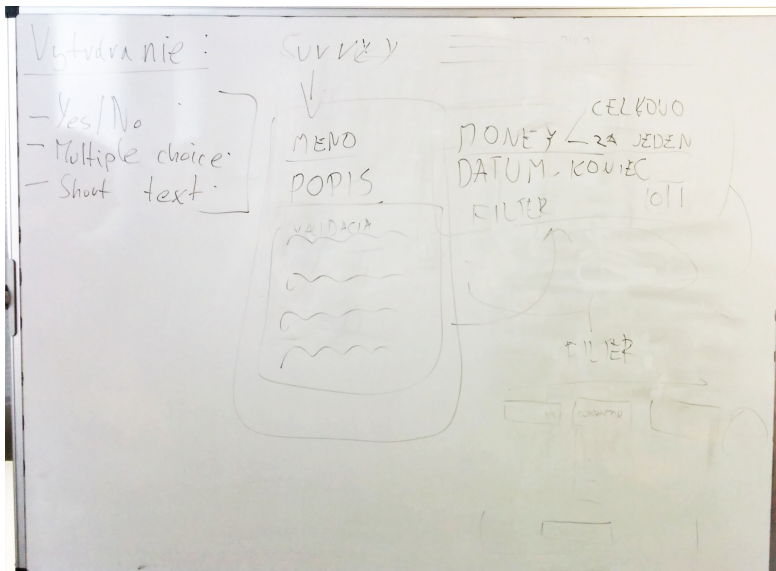
Analýza

V pôvodnom návrhu sme vytvorili používateľské rozhranie pre pridávanie nových úloh typu Survey (dotazník). Časom sme sa zhodli, že toto rozhranie nie je príliš prívetivé pre používateľa a tiež málo intuitívne pri plánovanom pridávaní nových typov podúloh. Preto sme sa rozhodli rozhranie zjednodušiť, inšpirovali sme sa online nástrojmi pre tvorbu dotazníkov Typeform¹, ktorý má jednoduché, intuitívne ovládanie s možnosťou pridávania otázok štýlom drag&drop a preto je oveľa používateľsky prívetivejší aj pre ľudí, ktorí nie sú príliš technicky zdatní.

Návrh

V treťom kroku procesu vytvárania kampane pridáva zadávateľ jednotlivé otázky dotazníka pomocou drag&drop, pričom má na výber z niekoľkých možných typov otázok menovite: áno/nie otázka, multi-choice, otvorená otázka. Tieto typy patria medzi základné otázky pričom v budúcnosti chceme dopĺňať nové typy otázok. Náš návrh spĺňa podmienky pre zjednodušenie používateľského rozhrania pre používateľa. Low fidelity návrh pre zjednodušené rozhranie je znázornený na Obrázku 31

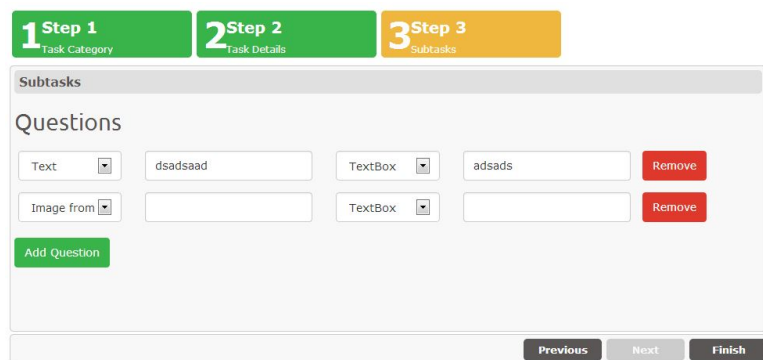
¹<http://www.typeform.com/>



Obr. 31: Low fidelity návrh UI pridávania úloh

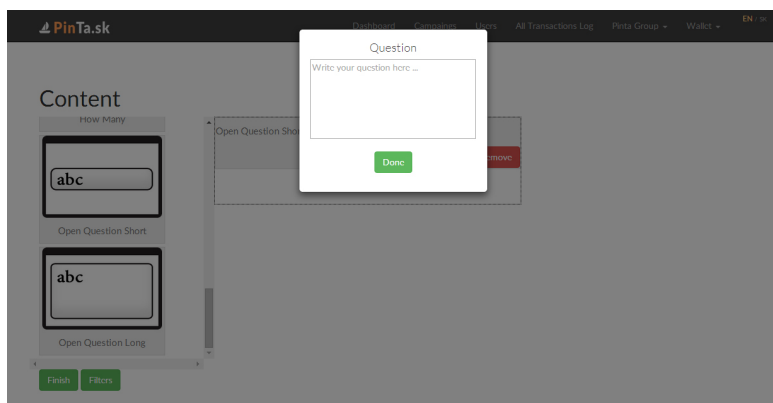
Implementácia a testovanie

Pred: implementované pomocou Smart wizard.



Obr. 32: Krok 3 pri vytváraní úlohy

Po: drag and drop a modálne okná



Obr. 33: Krok 3 nové rozhranie

3.10.2 Oprava klíčových chýb v systéme

Autor: Bc. Róbert Černý

Počas testovania systému sme zistili logické chyby vo fungovaní aplikácie. Niektoré z chýb boli natoľko závažné, že nám bránili vo vývoji novej funkcionality. Väčšina z nich sa nachádzala v procese vytvárania nových kampaní, nakoľko je tento proces komplikovaný vzhľadom na nové typy kampaní. Chyby sme zaznamenali do systému JIRA. Počas prvého šprintu sme vyriešili tieto chyby v systéme:

- pracovník bol schopný vyplniť kampaň ľubovoľný počet-krát cez link,
- na dashboarde chýbala možnosť zmeniť jazyk (vlajky boli skryté),
- po prihlásení bol používateľ presmerovaný na domovskú stránku (správne má však vidieť dashboard),
- po prihlásení sa používateľovi zobrazila správa o úspešnom odhlásení,
- výsledky kampane boli globálne prístupné cez linku,
- zadávateľ vedel vytvárať kampaň aj bez kreditu,
- link pre registráciu smeroval na starý formulár (nový bol riešený modálne priamo z homepage),
- pri vytváraní kampane mohol zadávateľ zadať menší kredit ako cenu jednej otázky (vytváralo problémy v DB, nebolo možné implementovať transakčný systém),
- v otázke typu multi-choice bolo možné zadať prázdny reťazec,
- navigačný panel mal nepresné názvy, ktoré neboli jednoznačné a samovysvetľujúce.

Vyššie popísané chyby sme odstránili.

3.10.3 Zmena layoutu pre zobrazenie výsledkov. Zovšeobecnenie dátového modelu pre existujúce typy úloh a ich podtypy (survey)

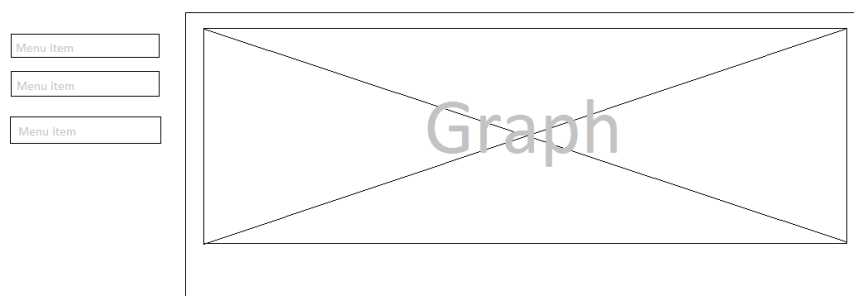
Autor: Bc. Miroslav Vojtuš

Analýza

Pôvodná implementácia z #8 šprintu zimného semestra bola len základnou implementáciou založenou na plug-ine HighChart. Zobrazené výsledky nekorešpondovali s designom. Museli sme prispôbiť vzhľad aby korešpondoval so používanou šablónou Bootstrap. Taktiež výsledky boli použiteľné len pre jeden typ podúlohy. Museli sme zohľadniť aj novopribudnuté podúlohy a možnosť pridania ďalších podúloh. Grafy sa zobrazovali všetky na jednej stránke čo pôsobilo rušivo pre používateľa, najmä v prípade rozsiahlejšej kampane.

Návrh

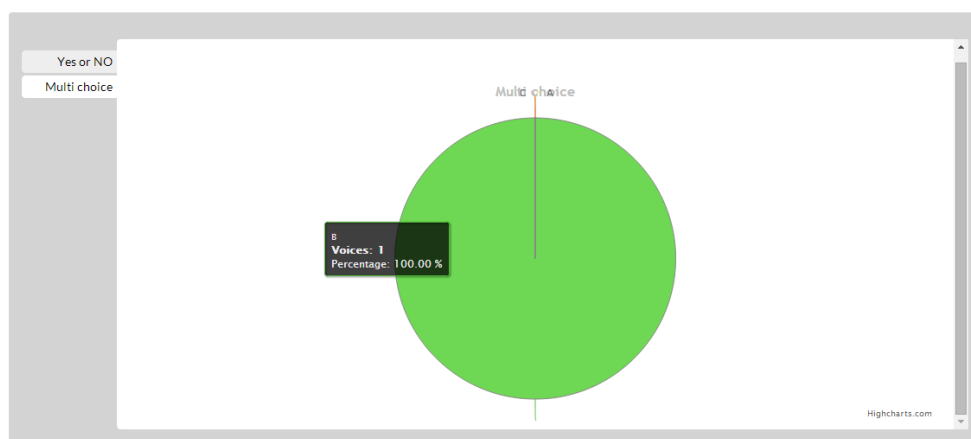
Aby sme sprehľadnili výsledky, rozdelili sme ich na základe podúloh. Každá podúloha má vytvorenú položku v ľavom ovládacom paneli. V pravom paneli sa zobrazí graf podľa zvolenej položky. Tento návrh je znázornený na Obrázku 34.



Obr. 34: Návrh zobrazenia grafov výsledkov

Implementácia a testovanie

Pre implementáciu sme využili základnú kostru pre HighCharts z minulého semestra. Dátový model sme čiastočne pozmenili aby vyhovoval rozdeleniu grafov do viacerých obrazoviek. Dáta sú zoskupené podľa jednotlivých podúloh. Výsledná obrazovka je znázornená na Obrázku ??.



Obr. 35: Výsledný graf

3.10.4 Vytvorenie prezentačnej stránky aplikácie

Autori: Bc. Patrik Štrba, Bc. Martin Toma

Analýza

V rámci webovej aplikácie je potrebné samotnú aplikáciu prezentovať. Preto sme potrebovali vytvoriť prezentačnú stránku aplikácie, kde by návštevníci mohli vidieť čo robíme a v čom by im aplikácia mohla pomôcť. Taktiež z prezentačnej stránky je vstup, prihlásenie a registrácia do aplikácie.

V rámci analýzy sme si najskôr museli ujasniť hlavné časti aplikácie a jej prínosnosť pre ľudí.

Návrh

V návrhu sme museli najskôr navrhnuť rozhranie ako by sme chceli aby táto stránka vyzerala. Vzhľadom na to, že sa jedná o prvú stránku čo návštevník uvidí, tak sme chceli aby bola stránka zaujímavá a zaujala čo najväčšie percento návštevníkov.

Implementácia a testovanie

Prezentačnú stránku sme vytvorili ako jednu stránku, ktorá sa skladá z viacerých častí. Na tieto časti sa je možné dostať z navigačného menu, ktoré sa nachádza navrchu stránky.

Stránku sme naprogramovali s využitím HTML5, CSS a pomocného frameworku s názvom Bootstrap. V rámci implementácie sme museli viackrát upravovať obsah na stránke, aby bol čo najvhodnejší a najviac vystihoval samotnú aplikáciu a zároveň bol jednoduchý.

3.10.5 Refaktoring spojenej úlohy

Autor: Bc. Miroslav Molnár

Analýza

Pri práci na projekte sa často našli chyby v zdrojovom kóde pri schvaľovaní zmien. Následne sa tieto zmeny neposlali na spoločný repozitár, ale boli vrátené programátorovi na opravu. Schvaľovanie zmien trvá aj niekoľko dní. Programátor ale musel už vyvíjať ďalšie veci a nemohol si dovoliť čakať až kým sa predošlé zmeny schváli a až potom pracovať na ďalšom zadaní. Takýmto spôsobom sa vytvorila skupina zmien, ktorá bola veľmi veľká, ťažko kontrolovateľná na chyby a ešte horšie spojitelná s ostatným kódom v repozitári.

Návrh

Pre jednoduchšie spracovanie zmien museli byť odstránené všetky konflikty s aktuálnym repozitárom. Následne sa museli povytvárať nové inštancie aktuálnej verzie aplikácie z repozitára do ktorých sa presúvali jednotlivé menšie zmeny. Takto menšie časti zmien sa museli znova pretestovať. Testovala sa správna funkcionálna. Následne sa postupne jednotlivé menšie časti pridávali do repozitáru. Po pridaní jednej časti sa museli prekontrolovať všetky ostatné inštancie či nevznikli konflikty ak áno museli sa odstrániť a až potom sa mohla pridať do repozitáru ďalšia časť. Pri ďalšom vývoji sa každé zadanie vytváralo už na zvlášť vymedzenej inštancii repozitára. Toto riešenie je zdĺhavé pre vývojára, pretože musí udržiavať viaceré inštancie repozitára stále aktuálne, ale vyhne sa tomu, že sa zmeny budú postupne spájať do jednej inštancie lokálneho repozitára.

3.10.6 Proces zefektívnenia tvorby zápisníc - automatický generátor šablóny

Autor: Bc. Marek Lenčes

Analýza

V priebehu stretnutia je náročné zachytiť všetky myšlienky, ktoré nás napadnú pri riešení príbehov a zároveň ich vpisovať do pripravenej šablóny zápisníc vytvorenej v zimnom semestri v typografickom systéme L^AT_EX, navyše ak sa to počas stretnutia nepodarí, je potrebné zápisnicu písať dodatočne a to zaberá zbytočný čas. Preto sme sa rozhodli zefektívniť tvorbu zápisníc automatickým generovaním priamo na stretnutí.

Návrh

Generátor zápisníc bude obsahovať jednoduché rozhranie pre vepisovanie potrebných údajov do zápisnice tak, ako sú štandardne postupne vyplňané a po vyplnení vygeneruje hotový naformátovaný súbor so zápisnicou zo stretnutia.

Implementácia a testovanie

Generátor zápisníc je vytvorený v programovacom jazyku Java a pre generovanie používa prenosnú verziu balíka MiKTeX s L^AT_EX kompilátorom pdfLatex, generátor automaticky vyplní potrebné údaje do šablóny, čo značne urýchľuje proces písania zápisníc. V prvej verzii je potrebné ešte vygenerovanú šablónu doupravovať, pridať jednotlivé úlohy členom a vygenerovať hotový dokument. Používateľské rozhranie je znázornené na Obrázku 36.

Obr. 36:

3.11 Šprint #10

3.11.1 Vytvorenie prototypu notice úlohy

Autor: Bc. Patrik Štrba

Analýza

V systéme sa nachádzalo viacero druhov úloh, avšak my sme potrebovali úlohu, kde by zadávateľ mohol zadať úlohu, respektíve zanechať oznam a pracovník, ktorý by túto úlohu vykonával, by vybral či má záujem splniť čo je napísané v ozname alebo nie. Jedná sa o veľmi jednoduchý typ úlohy, ktorá ponúka viaceré možnosti využitia.

Návrh

V aplikácií už bolo navrhnuté rozhranie, ktoré umožňovalo jednoducho pridať nový typ úlohy. Tento typ úlohy bol ale špecifický tým, že po vykonaní sa automaticky nepripočítavali kredity pracovníkovi. Preto bolo potrebné navrhnuť systém, ktorý umožní prenos kreditov až po schválení úlohy zadávateľom.

Implementácia a testovanie

V rámci implementácie bolo potrebné vykonať veľa zmien. Pre implementovanie nového typu úlohy sme museli spraviť dve veci a to postup pri založení úlohy a jej vyplňanie. Založenie úlohy bolo dosť naviazané na typ úlohy dotazník a preto bolo potrebné zrefaktorovať kód a vytvoriť abstrakciu, ktorá výsledný kód spraví prehľadnejší a jednoduchšie udržiavateľnejší. V rámci tejto abstrakcie sa spravil celý postup pri vytváraní úlohy, ktorá má tri kroky. V rámci týchto troch krokov bolo potrebné vytvoriť medzivrstvu, ktorá by sa dokázala prispôbiť aj na ďalšie typy úloh, ktoré by v budúcnosti mohli vzniknúť.

Taktiež bolo potrebné spraviť úpravy v šablónach a triedach obsluhujúcich vyplňanie úloh. Tento typ úlohy je jednoduchý tým, že bolo treba spraviť len jednu šablónu na vyplnenie či používateľ danú úlohu splní alebo nie.

Táto úloha sa odlišovala správaním pripočítavaním odmeny. Tieto zmeny viedli na vytvorenie transakcií, ktoré budú neskôr popísané. Taktiež bolo potrebné implementovať rozhranie, kde bude zadávateľ potvrdzovať vykonané úlohy používateľov a tým im pripočítavať kredity za splnenie úlohy.

3.11.2 Návrh a implementácia zobrazovania úloh

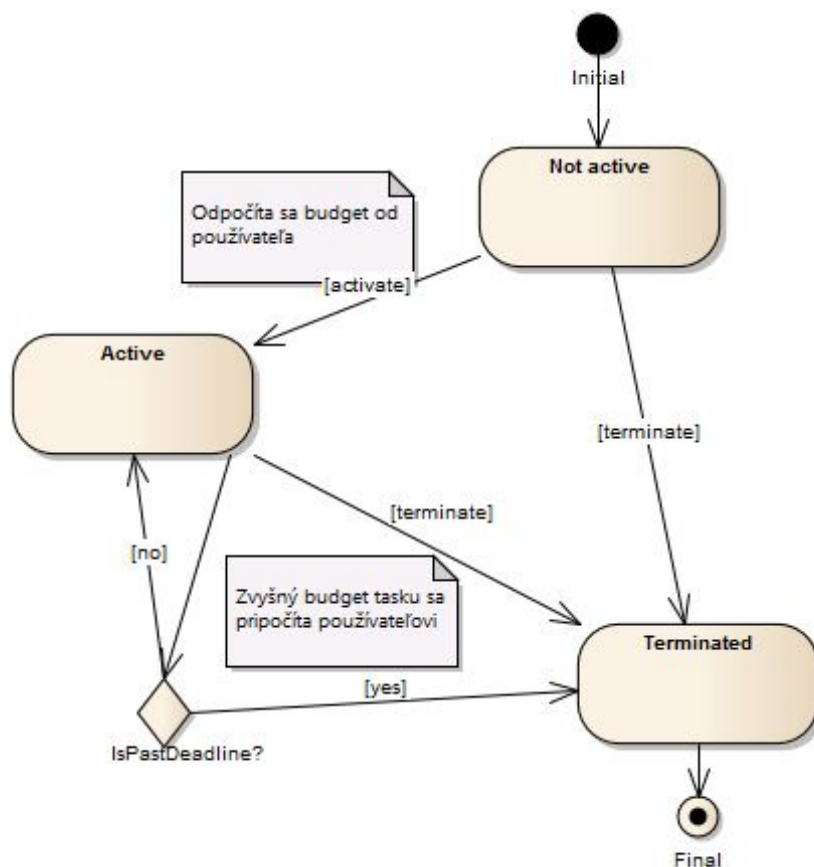
Autor: Bc. Filip Bednárík

Analýza

V systéme potrebujeme rozlíšiť, ktoré údaje by mali byť viditeľné zadávateľom úlohy a ktoré potencionálnym riešiteľom úlohy. Potrebujeme vedieť aké procesy by mali prebiehať v životnom cykle úlohy, aké operácie je možné s ňou vykonávať a vykonávať ich jednotne ako pre zadávateľa tak pre pracovníkov. Potrebujeme tiež sledovať štatistiky jednotlivých zadaných úloh, nakoľko sú splnené alebo rôzne detaily ako dátum ukončenia úlohy alebo cena úlohy.

Návrh

Pri návrhu využijeme existujúce vlastnosti úloh s pridaním ďalších, pre zobrazovanie jednotlivých úloh budú použité sekcie podobne ako v profile, sekcie budú obsahovať základné informácie o úlohe, akcie, ktoré možno vykonať s úlohou, štatistiky úlohy, výsledky, filtre. Základné úpravy úlohy bude možné vykonávať priamo v detailnom zobrazení úlohy. Zadávatel úlohy môže spúšťať akcie nad úlohou v závislosti od jej aktuálneho stavu. Možné stavy úlohy sú znázornené na Obrázku 37. V stave neaktívnej úlohy existuje možnosť upravovať úlohu alebo aktivovať úlohu.

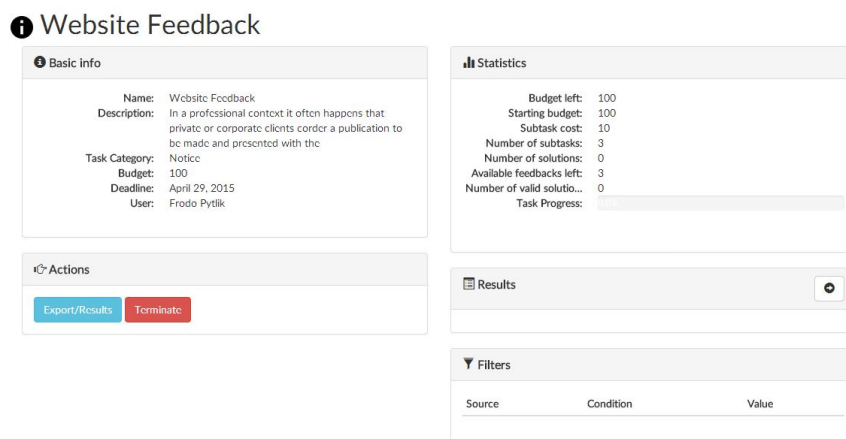


Obr. 37: Stavový diagram úlohy

V stave aktívnej úlohy existuje možnosť ukončiť úlohu a exportovať výsledky. V stave terminovanej úlohy existuje možnosť exportovať výsledky.

Implementácia a testovanie

Implementované zobrazenie úlohy je znázornené na Obrázku 38.



Obr. 38: Detailné zobrazenie úlohy

3.11.3 Návrh API rozhrania

Autor: Bc. Miroslav Molnár

Analýza

Pre rozšírenie a uplatnenia aplikácie sme navrhli API rozhranie pomocou ktorého by si mohli tretie strany implementovať svoje vlastné aplikácie. Tieto aplikácie by boli schopní manipulovať s naším systémom. Výhodou je, že tretie strany nemusia byť závislé od vytvorenia webovej aplikácie, ale môžu vytvoriť aplikáciu pre desktopové systémy. Rovnako je možné rozšírenie aj na mobilné zariadenia. Hlavnou myšlienkou je voľnosť pre implementáciu vlastných rozhraní pre konkrétny typ používateľov. Tretie strany implementujúce toto rozhranie sú si schopné pridávať a rozširovať funkcionality, ale rovnako vložiť svoju implementáciu už do existujúcich projektov. Pomocou týchto výhod môže vzniknúť viacero aplikácií komunikujúcich s naším systémom. Používateľ si následne môže vybrať medzi viacerými aplikáciami a zvoliť si tú ktorá jemu bude najviac vyhovovať.

Návrh

Rozhranie API je potrebné oddeliť od funkcionality zvyšného kódu. Je potrebné aby sa nevytvárala duplicita kódu a využívali sa už implementované metódy pre spracovanie požiadaviek a následných odpovedí. Tieto dva prvky sú základnou myšlienkou pre vytvorenie návrhu API rozhrania.

Implementácia a testovanie

Pri samotnej implementácii bola funkcionality API oddelená od ostatnej funkcionality systému a vložená do špeciálneho adresára v kontroléri s menom „api“. Všetka funkcionality API je uložená do tohto adresára a tým pádom je oddelená od ostatnej funkcionality. Pre zabránenie duplicity kódu sme mali na výber dve možnosti. Jednou z možností bolo využitie helper class do ktorých by sa metódy zdieľané s API mohli vložiť. Druhou možnosťou, ktorá bola implementovaná je posielanie špeciálnych požiadaviek na metódy s typom návratového formátu JSON.

Následne používateľ dostane JSON dáta v príslušnom formáte, ktoré si môže spracovať.

3.11.4 Proces vytvárania kampane

Autor: Bc. Róbert Černý

Analýza

Proces vytvárania novej kampane bol riešený pomocou smart wizardu pomocou AJAX. Tento prístup sme sa rozhodli zmeniť do viacerých krokov. Dôvodom je oddelenie základných informácií kampane od špecifických. Ďalej bude možné ukladať kampaň do databázy a vyplniť ďalšie informácie neskôr. Tento spôsob umožňuje zadávateľovi ľahšiu modifikáciu vytváraných kampaní. Pri vytváraní novej kampane sa vyskytujú viaceré problémy. Zadávateľ môže kedykoľvek opustiť proces vytvárania kampane. Znamená to, že aktuálny stav kampane sa musí dočasne uložiť do databázy. Dôležitým aspektom je aj bezpečnosť pri vytváraní novej kampane, nakoľko sa pracuje s kreditmi zadávateľa.

Návrh

Návrh spočíva vo vytvorení novej kampane v 4 logických krokoch, prvým krokom je výber kategórie kampane z dostupných kategórií, ďalej nasleduje vyplnenie základných informácií o kampani, nastavenie ceny a dátumu ukončenia ako aj popisu a názvu kampane. Tretí krok je špecifický pre typ zvolenej úlohy. Posledným krokom je vytvorenie filtrov na nasmerovanie kampane správnym respondentom, dôraz na jednoduché a logické pridávanie nových filtrov. Po vytvorení kampane je potrebné kampaň aktivovať aby bola viditeľná (naznačiť túto skutočnosť aj zadávateľovi). Proces vytvárania kampane je možné tiež prerušiť po druhom kroku a dokončiť vytváranie neskôr.

Implementácia a testovanie

KROK 1. - Výber typu kampane

Create new campaign



Obr. 39: Výber typu kampane

KROK 2. - Zadávane základných informácií

New Survey

Name

Write description here ...

Budget

Solution amount

Solutions count

Subtask cost

Cost of subtask

Deadline - 9:10

Obr. 40: Zadávane základných informácií

KROK 4 - dashboard pre kampaň s možnosťou výberu filtrov, filtre boli implementované pomocou gemu nested fields

Task was successfully created.

Name: Task
Description: Description
Number of people: 10
Deadline: 2013-12-11 20:42:00 UTC
Task category: Survey
User: Jason Derulo

:

Sourcetype

Conditiontype

Value

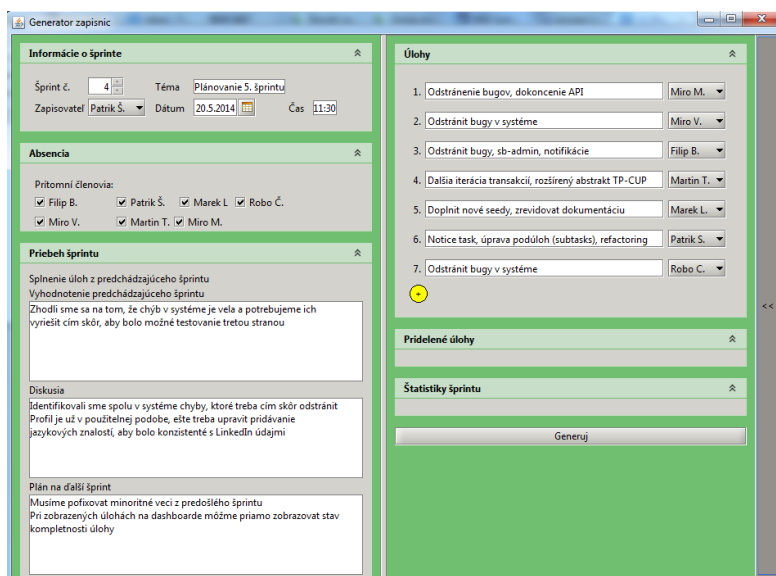
Obr. 41: Dashboard pre kampaň

3.11.5 Generátor zápisnic v.2 a úprava vzhľadu

Autor: Bc. Marek Lenčes

Implementácia

Generátor zápisnic sme doplnili o možnosť automatického pridávania úloh jednotlivým členom tímu, po stlačení tlačidla Generuj sa už vygeneruje hotový PDF dokument so zápisnicou zo stretnutia. Týmto sa značne zrýchli proces písania zápisnic. Tiež sme pridali do zápisnice logo tímu. Finálnu podobu generátora môžeme vidieť na Obrázku 42.



Obr. 42: Generátor zápisnic v.2

3.11.6 Rozšírenie výsledkov o ďalšie typy grafov. Rýchla zmena bez znovu načítania dát

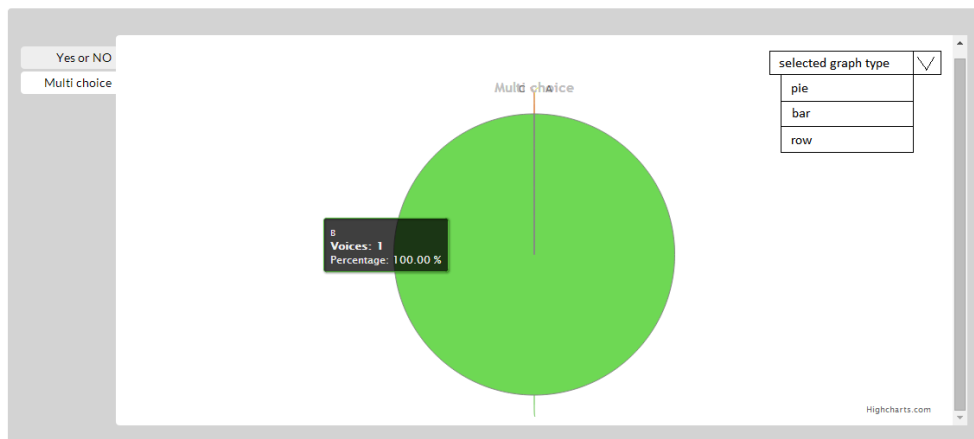
Autor: Bc. Miroslav Vojtuš

Analýza

Po odsúhlasení implementácie kolektívom sme sa rozhodli rozšíriť bázu poskytovaných grafov o nové pre spiestrenie výstupu. Hlavným úmyslom bolo umožniť používateľom vybrať si spôsob zobrazenia dát podľa požiadaviek.

Návrh

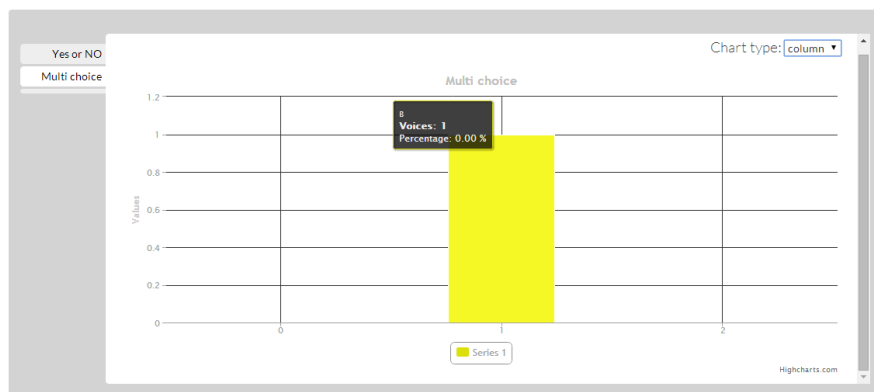
Keďže samotné získanie dát zo serveru je náročný výpočtový úkon, pretože dáta musia byť transformované medzi rôznymi dátovými modelmi (db vs HighCharts), rozhodli sme sa zovšeobecniť dátový model tak aby bolo možné meniť zobrazované dáta na strane klienta. Na klientovej strane sú všetky informácie potrebné pre zmenu typu grafu. Výsledná obrazovka mala zodpovedať návrhu na Obrázku 43.



Obr. 43: Výber typu grafu z dropdown menu

Implementácia a testovanie

Funkcionalita bola implementovaná pomocou JavaScript-u. Transformácie prebiehajú na klientovej strane, pretože každý graf potrebuje čiastočne odlišné dáta. Na Obrázku 44 je obrazovka so stĺpcovým grafom. Zmena typu grafu prebieha výberom z menu v pravom hornom rohu.



Obr. 44: Stĺpcový graf

3.11.7 Zoznam úloh

Autori: Bc. Filip Bednárík, Bc. Róbert Černý

Analýza

Pre jednoduchú manipuláciu s úlohami potrebujeme pridať akcie priamo do zoznamu všetkých úloh bez nutnosti zobrazenia detailov úlohy, napríklad pridanie tlačidla na zmenu stavu úlohy. Stĺpce, ktoré používateľ nemusí vidieť zostanú skryté. Namiesto názvu kategórie úlohy budeme zobrazovať kvôli lepšej prehľadnosti a intuitívnemu ovládaniu ikonu typu úlohy. Názov kampane môže odkazovať na jej dashboard. Tiež tlačidlá akcií by bolo vhodné kvôli prehľadnosti zmeniť na ikony a zobrazovať iba tie akcie, ktoré môže aktuálny používateľ vykonať. Tlačidlo na vytváranie novej kampane by bolo vhodné tiež presunúť na vhodnejšie miesto.

Návrh

Pred: Po:

Your current tasks list

Create new task

Id	Name	Description	People count	Subtask cost	Budget	Deadline	Category	User	Created	Action
1	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
2	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
3	test		100	1	1		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
4	test2		100	1	2		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
5	test		100	1	0		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
6	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
7	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
8	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
9	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
10	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete

Obr. 45: Zoznam pred implementáciou

My Campaings

Create new campaing

Category	Name	Subtask cost	Budget	Deadline	Created	Status	Action
?	BioStore e-shop feedback	2	198	April 29, 2015	April 28, 2014	ON	⌵ ⌶ ⌷
?	Rate TroX vacuum Cleaner	5	20	April 29, 2015	April 28, 2014	ON	⌵ ⌶ ⌷
?	Medium Task	5	50	April 29, 2015	April 28, 2014	ON	⌵ ⌶ ⌷
?	RoNav GPS Feedback	10	100	April 29, 2015	April 28, 2014	ON	⌵ ⌶ ⌷
f	Like our e-shop BuyMcTux.com	10	100	April 29, 2015	April 28, 2014	ON	Budget ⌵ ⌶ ⌷

Obr. 46: Zoznam po implementácii

Implementácia a testovanie

Podľa návrhu sme vypracovali nový vzhľad zobrazovania úloh, odstránili sme niektoré nepodstatné stĺpce a na prepínanie stavu úlohy priamo v zozname sme použili JQuery, ajax a bootstrap switch.

3.11.8 Flow s riešením úloh

Autor: Bc. Matin Toma

Analýza

V našom systéme je riešenie úloh jedným z najdôležitejších procesov. Preto sme sa chceli viac zamerať na tento proces a vytvoriť z neho jednu z najväčších výhod nášho riešenia. Analyzovali sme všetky potrebné údaje, ktoré by mali byť pre používateľa zobrazené v procese riešenia úlohy. Inšpirovali sme sa existujúcimi riešeniami, pričom najviac nás zaujalo riešenie <http://www.typeform.com/>.

Návrh

Navrhli sme teda proces pri ktorom používateľ bude postupne vyplňať jednu otázku po druhej. Tie budú zobrazované v centre obrazovky a budú optimalizované aj na obrazovky tabletov či mobilných telefónov. Na obrazovke by malo byť viditeľné

hlavne ako sa daná kampaň volá, otázka a možnosti, škály, textové oblasti a podobne. Okrem toho by bolo vhodné zobrazovať aj poradenie otázky. Ďalšie prvky sme nepotrebovali no v prípade vhodného nápadu sme sa im nebránili.

Implementácia a testovanie

Na obrázku nižšie vidíme finálnu implementáciu. Rozhodli sme sa okrem vyššie spomínaných atribútov zobraziť aj dátum ukončenia danej kampane. Experimentovali sme s veľkosťou a typom písma pre jednoduché a rýchle čítanie. Riešenie



Obr. 47:

umožňuje používateľovi sústrediť sa na danú otázku bez zbytočného rozptyľovania a scrollovania dlhého formulára. Taktiež tým podporujeme a zvýrazňujeme možnosť prerušenia riešenia kampane a neskoršieho vrátenia sa k poslednej riešenej otázke.

3.12 Šprint #11

3.12.1 Identifikácia chýb v systéme a ich odstránenie

Autor: Bc. Róbert Černý

Analýza

Počas implementácie funkcionality z predchádzajúcich šprintov sa v systéme začali objavovať chyby. Väčšina z nich súvisí s procesom vytvárania novej kampane. Preto sme museli identifikovať tieto chyby a zaviesť ich do systému JIRA. Po evidovaní chýb sme postupne vytvárali riešenia.

Chyby v systéme vznikli nedostatočným testovaním implementovanej logiky, čo sa odstráni zavedením testovania. UI prvky nefungujú na všetkých webových prehliadačoch rovnako. Tento fakt spôsobuje hneď niekoľko chýb v modálnych oknách v prehliadači Firefox.

Návrh

Pre identifikáciu chýb musíme prejsť základné procesy v aplikácii a identifikovať chyby, použijeme pritom rôzne internetové prehliadače (Chrome, Firefox, Safari).

Identifikované chyby vložíme do systému JIRA, zameriame sa aj na zlepšenie UI a použiteľnosť. Musíme taktiež odstrániť možné spôsoby zavedenia chyby do databázy a teda musíme identifikovať chyby v seedoch. Tiež je potrebné optimalizovať proces vytvárania kampane dotazník a vyriešiť všetky identifikované chyby v systéme.

Implementácia a testovanie

V procese sme objavili a reportovali celkovo 12 chýb z rôznou vážnosťou. Všetky tieto chyby sme odstránili. Väčšina z nich sa týkala procesu vytvárania novej kampane typu dotazník. Zistili sme taktiež neúplnosť seedov a následne sme ich doplnili.

3.12.2 Vylepšenie zobrazenia výsledkov. Pridanie filtrov. Csv výstup dát.

Autor: Bc. Miroslav Vojtuš

Analýza

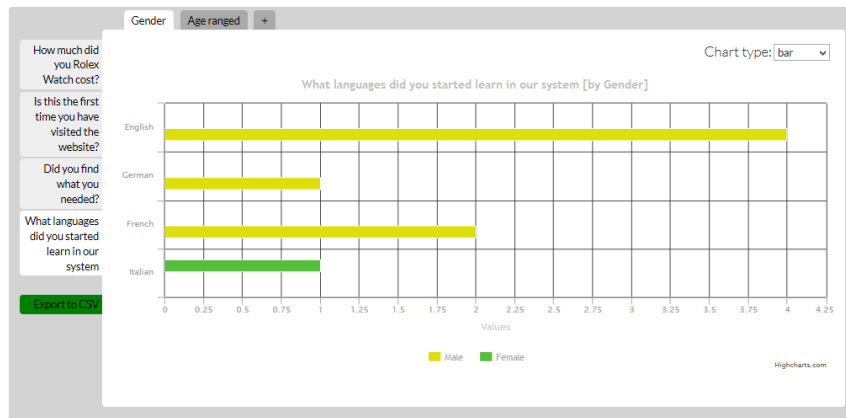
Používateľ zvyčajne potrebuje rôzne uhly pohľadu na dáta. Častým pohľadom je napríklad vekové rozloženie respondentov. Prípadne rozlíšenie medzi pohlaviami. Tieto dáta treba rozumne vyfiltrovať z databázy a premietnuť do výsledkov vhodnou formou.

Návrh

Plne korešponduje s analýzou. Nakoľko ešte nemáme spätnú väzbu od potenciálnych klientov. Navrhujeme len základnú sadu filtrov. V našom prípade je to filter podľa veku a filter podľa pohlavia. Vekový filter bude pevne ohraňovaný. Po získaní spätnej väzby môže byť parametrizovateľný. V prípade záujmu môžeme v budúcnosti navrhnúť generickú verziu filtrov, kde si klient špecifikuje konkrétne parametre pre rozlíšenie výsledkov.

Implementácia a testovanie

Pridali sme záložky nad grafmi ktoré umožňujú zvoliť požadovaný filter. Tento je príslušnému kontroleru dodaný ako parameter. Kontroler vytvorí vhodný dopyt do databázy a získa požadované dáta. Tieto potom transformuje do dátového modelu vhodného pre HighChart modul. Výsledná obrazovka je zobrazená na Obrázku 48. Dáta je taktiež možné exportovať do CSV formátu (Comma Separated Values). Prototypová verzia exportu neumožňuje konfiguráciu. Dáta použité vo výsledkoch sú totožné s dátami použitými na zobrazenie grafov. V budúcnosti by bolo užitočné exportovať aj dáta o používateľoch, ktorí danú úlohu riešili pre filtrovanie výsledkov vo vlastnej réžii.



Obr. 48: Grafy z výsledkov otázok s viacerými odpoveďami

3.12.3 Implementácia HTTP overovania používateľa pre API

Autor: Bc. Miroslav Molnár

Analýza

Kvôli zabezpečeniu systému je nutné aby používatelia využívajúce API rozhranie boli riadne overení a nemohlo sa stať, že dáta budú ľubovoľne získané inou osobou ako osobou ktorá má na tieto dáta skutočný nárok. Preto sa prístupové práva overujú predtým ako sa požiadavka skutočne vyrieši.

Návrh

Pre overenie prístupových práv používateľa je treba použiť autentifikáciu. Od používateľa sa vypýta prihlasovacie meno a heslo. Okrem toho je nutné aj autorizovať používateľa na konkrétne typy dát, z dôvodu zabráneniu získania informácií na ktoré nemá používateľ právo. Napríklad osobné informácie vyplnené v profile.

Implementácia a testovanie

Pre implementáciu používateľa sme použili http autentifikáciu. Používateľ musí zadať svoje prihlasovacie meno a heslo, ktoré sa uloží do session a následne posielajú ďalšími požiadavkami. Pre implementáciu sme priamo použili metódu `authenticate_or_request_with_http_basic`, ktorá zabezpečuje bezpečné overenie http autentifikácie používateľa. Ukážka kódu nižšie:

```

1 def authenticate_api
2   authenticate_or_request_with_http_basic('Solution') do |login_name, password|
3     if !(@act_user = User.find(:first, :conditions => {:email => login_name})).
4       nil? && @act_user.valid_password?(password)
5       @current_user = @act_user
6       true
7     else
8       redirect_to(root_path) and return false
9     end
10  end
end

```

Pre autorizáciu bolo nutné overiť práva používateľa na konkrétne dáta. To bolo docielené overením používateľa a jeho práv na získanie konkrétnych dát v systéme. Používateľ mohol získať iba svoje vlastné dáta. Najprv je potrebné zistiť vlastníka dát ak existuje ak neexistuje dáta sú verejné. Ak existuje porovná sa vlastníka dát

s prihláseným používateľom. Ak vlastník dát aj prihlásený používateľ sú rovnaký tak sa dáta pošlú inak sa vráti chybová hláska o neoprávnenom prístupe.

3.12.4 Vytvorenie prvej verzie transakcií

Autor: Bc. Martin Toma

Analýza

V systéme bola navrhnutá možnosť získania odmeny za vyriešenie kampane. Táto odmena (kredit) musí byť nejakým spôsobom reprezentovaná a jej výmena medzi používateľmi kontrolovaná. Rátali sme v budúcnosti aj s možnosťou výmeny za reálne peniaze, čo iba zvyrazňuje potrebu nejakej formy evidencie týchto výmien, transakcií.

Návrh

V tomto šprinte sa jednalo o prvú verziu transakčného systému. Nakoľko je tento systém veľmi dôležitý a je pre nás kľúčová aj jeho bezpečnosť a robustnosť, rozhodli sme sa ho implementovať inkrementálne vo viacerých fázach. V prvej fáze sme navrhli jednoduchý dátový model. Navrhli sme pridať iba jednu tabuľku, transactions, ktorá by mala obsahovať referencie na všetky entity v systéme ktoré budú súvisieť s danou transakciou.

Implementácia a testovanie

Po implementácii tabuľka transactions obsahovala identifikátor kampane, otázky používateľa ktorý získava a rovnako používateľa ktorý odovzdáva kredity. Záznam do databázy o novej transakcii sa vykonával vždy po každej odpovedi na otázku (bez čakania na ukončenie celej kampane). To bolo z dôvodu možnosti prerušenia vykonávania kampane. Takto sme mohli zaznamenávať čas odoslania každej odpovede.

3.12.5 Optimalizácia postupu pri vytvorení úlohy a pridanie úpravy celého postupu pri vytváraní úlohy

Autor: Bc. Patrik Štrba

Analýza

Po dokončení úlohy s pridaním novej úlohy vznikol chaotický proces pri pridávaní úloh. Pridávanie bolo neprehľadné a nový používateľ by nebol schopný vytvoriť takúto úlohu. Preto sme potrebovali vytvoriť optimálnejší postup pri pridávaní novej úlohy. Taktiež bolo potrebné vytvoreniu úlohu upravovať a manipulovať s vytvorenými filtrami. Toto bolo taktiež potrebné vytvoriť.

Návrh

Aplikácia poskytuje spätnú väzbu a preto by mal byť proces vytvárania úlohy, takéhoto prostredia kvalitne spravený a taktiež by mal byť prehľadný a jednoduchý. Najskôr bolo potrebné zistiť, ktoré operácie sa vyskytujú pri takomto procese a

potom navrhnuť abstraktnú verziu, do ktorej sa následne budú len pridávať prípadné zmeny pri konkrétnej úlohe. V rámci tohoto procesu som myslel aj na proces úpravy, ktorý bol tiež potrebný.

Implementácia a testovanie

Najskôr bolo potrebné proces vytvárania úlohy zoptimalizovať a následne poskytnúť používateľom aj možnosť úpravy úlohy. V rámci optimalizácie bolo vytvorených viacero šablón. Niektoré z nich boli spoločné a niektoré boli konkrétne pre konkrétne úlohy. Týmto sa podarilo vytvoriť akúsi hierarchiu v ktorej sa jednoduchšie hľadalo a následne aj udržiaval kód.

V rámci úpravy úlohy bolo potrebné zahnúť aj úpravu filtrov. Tieto filtre bolo možné vytvoriť pri procese vytvárania úloh. Následne v rámci úpravy úlohy bolo potrebné aby bolo možné tieto filtre upravovať či pridávať alebo odstraňovať.

3.12.6 Zlepšiť zobrazovanie badges

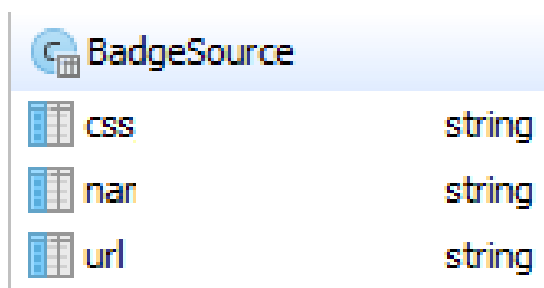
Autor: Bc. Filip Bednárík

Analýza

Ocenenia (badges) pre používateľov sú dobrým spôsobom ako poporiť snahu používateľov o aktivitu v systéme. Pre ďalšiu motiváciu potrebujeme upraviť zobrazovanie ocenení tak aby mohli používatelia sledovať svoj postup pri získaní ocenenia, t. j. koľko aktivít musia ešte v systéme vykonať aby ocenenie dostali. Ocenenia musia byť teda používateľsky prívetivejšie. Plánujeme tiež implementovať možnosť priamo sa dostať na stránku, kde používateľ získa ocenenie zo samotného ocenenia.

Návrh

Návrh databázového modelu pre ukladanie ocenení je znázornený na Obrázku 49

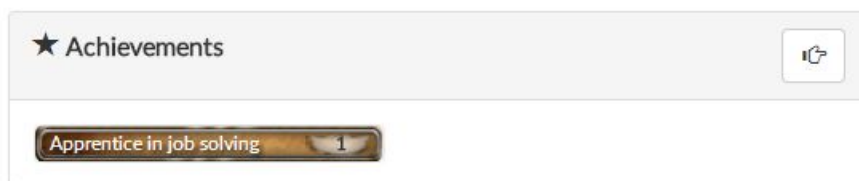


Obr. 49: DB model ocenení

Na obrázku 50, 51 a 52 je zobrazený grafický návrh ocenení, ocenenia sa dynamicky zobrazujú podľa dôležitosti formou animácie.

Implementácia

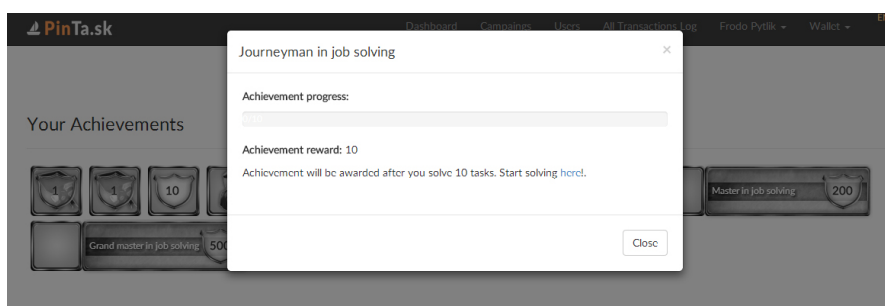
Pri implementácii podľa návrhu sme využili technológie JQuery, CSS, Freetile a modálne okná.



Obr. 50: Zobrazenie ocenení priamo v profile



Obr. 51: Zobrazenie všetkých ocenení



Obr. 52: Postup pre získanie ocenenia

3.12.7 Pridanie referral linku do profilu, pridanie zručností a zobrazenie aktuálnych úloh na dashboarde

Autor: Bc. Marek Lenčes

Analýza

Každý používateľ má po registrácii do systému k dispozícii svoj referral link - odkaz, cez ktorý sa môžu registrovať ďalší používatelia, za využitie tohto referral linku a po úspešnej registrácii dostane jeho vlastník určitý počet kreditov. Ideálne by mal mať používateľ svoj referral link prístupný v profile aby ho vedel vždy rýchlo nájsť. Tiež potrebujeme implementovať pridanie nových zručností do profilu používateľa a zobrazovanie nových, aktuálne pridaných úloh na hlavnom dashboarde aplikácie po prihlásení.

Návrh

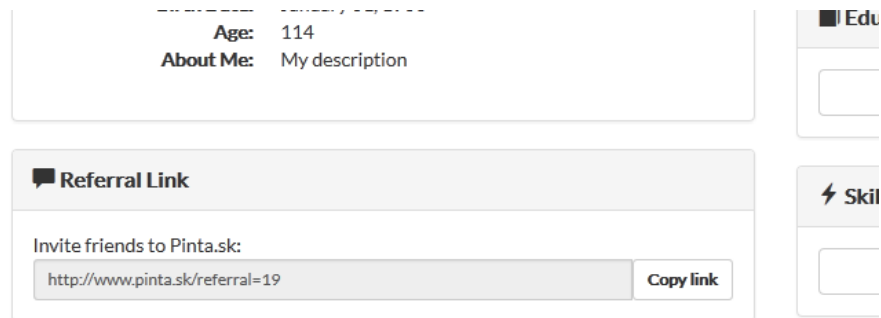
Referral odkaz by mal byť jasne viditeľný a keďže sa jedná o link, ktorý budú používatelia často kopírovať, umožňuje automatické skopírovanie do schránky po kliknutí.

Pridanie zručností bude len jednoduché pridanie textového reťazca predstávujúceho konkrétnu zručnosť.

Po pridaní úloh používateľmi sa najnovšie úlohy budú zobrazovať na hlavnej stránke aplikácie a budú tak viditeľné a dostupné pre riešenie iným používateľom.

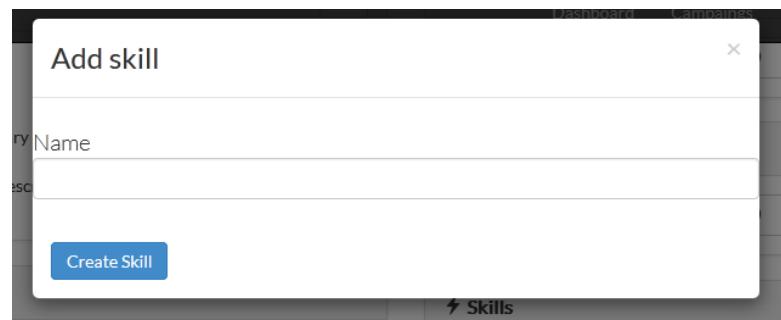
Implementácia

Vloženie referral odkazu do profilu je znázornené na Obrázku 53



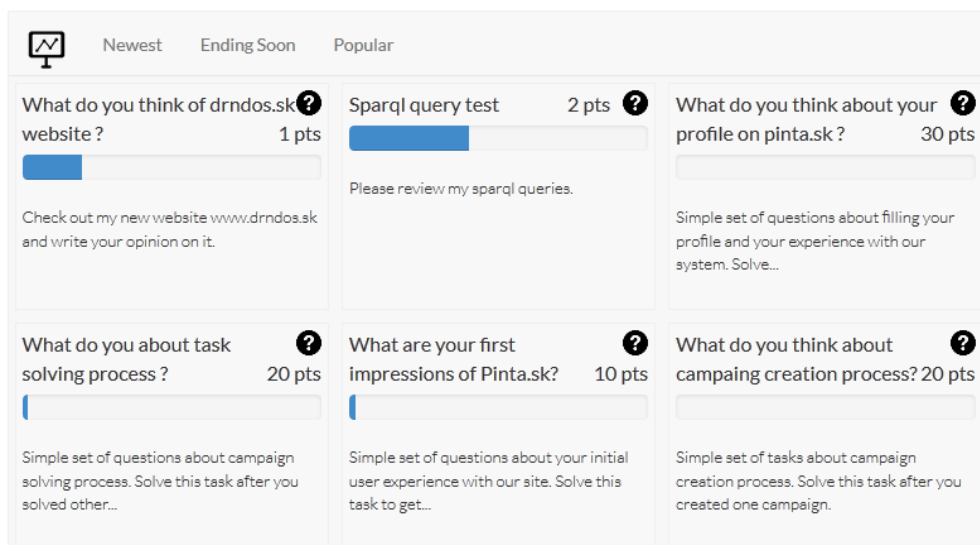
Obr. 53: Referral link

Pridávanie nových zručností je zobrazené na Obrázku 54.



Obr. 54: Pridávanie zručností

Zobrazovanie aktuálnych úloh na hlavnom dashboarde môžeme vidieť na Obrázku 55.



Obr. 55: Aktuálne úlohy na dashboarde

3.13 Šprint #12

3.13.1 Vylepšenie navigačného panelu, oprava niekoľkých chýb

Autor: Bc. Martin Toma

Analýza

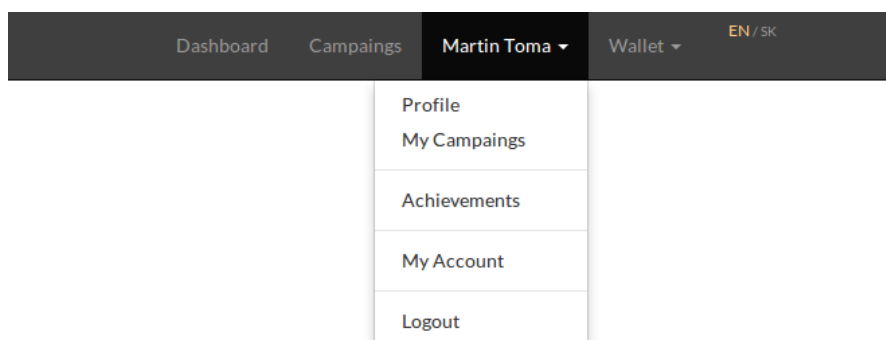
Navigačný panel je vo webovej aplikácii veľmi dôležitý. V našom prípade sme mali pocit, že sa stáva neprehľadným a ťažko tam nájdeme čo potrebujeme. Okrem toho sme potrebovali nájsť vhodné miesto a vhodný spôsob na prepínanie jazyka používateľského rozhrania. Preto musela prísť zmena dizajnu a aj obsahu navigačného panelu.

Návrh

Pri návrhu sme sa rozhodli využiť vysúvacie menu (drop-down) na zoskupenie akcií týkajúcich sa spoločnej domény. Napríklad operácie s používateľským účtom alebo informácie o transakciách a kredite. Zmena jazyku bola navrhnutá ako dve vlajčky zobrazené v pravom hornom rohu navigačného panelu.

Implementácia a testovanie

V implementácii sme využili prvky frameworku bootstrap. Vytvorili sme skupiny s akciami týkajúcimi sa používateľského účtu a profilu a informácii o kreditoch a transakciách. Finálnu verziu navigačného panelu už aj s novou verziou prepínača jazyka vidíme na Obrázku 56.



Obr. 56: Navigačný panel

3.13.2 Transakčný systém 2

Autori: Bc. Róbert Černý, Bc. Filip Bednárík

Analýza

Nakoľko náš systém pracuje s kreditmi, ktoré bude neskôr možné brať ako reálne peniaze, je stále viac potrebné všetky dôležité úkony v systéme zaznamenávať do databázy. V systéme je potrebné zaviesť transakčný systém. Prvá verzia tohto systému poskytovala iba základnú funkcionálnu a databázový model bol príliš zjednodušený. Pri transakciách je potrebné vedieť dané subjekty, ktoré manipulujú s

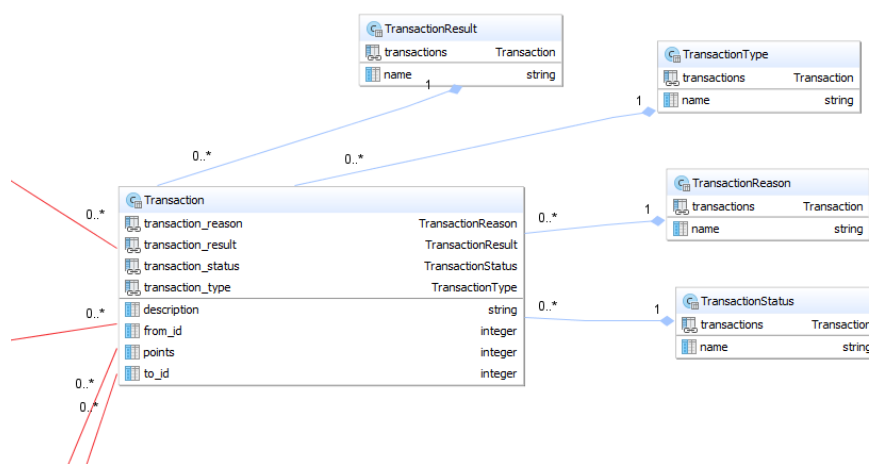
kreditmi. Tranzakcia bude mať typ a dôležitou vlastnosťou je ukladanie informácie o neúspechu ako aj bezpečný prevod kreditov z jedného konta na iný.

Návrh

vytvoriť transakčný model vytvoriť databázový model (uložiť informácie o subjektoch transakcie, počet kreditu, typ transakcie, stav transakcie, čas, popis o výsledku) vytvoriť jednoduchý konštruktor, nakoľko vytváranie transakcií bude často využívané vytvoriť stránku pre zobrazenie transakcií v systéme pre administrátora integrovať transakčné metódy do nášho systému, použitie pri vytváraní kampane, vytváraní výsledkov, terminovaní existujúcej kampane poskytnúť metódy pre: výber kreditu dobíjanie kreditu prevod kreditu zo zadávateľa na vytvorenú kampan prevod kreditu z terminovanej kampane zadávateľovi Pri transakciách je tiež dôležitý dôvod vykonania transakcie. Podľa dôvodu a typu transakcie vieme totiž namodelovať dedenie a zabezpečíme tak konzistenciu databázy. Vďaka typu transakcie vieme v zozname transakcií odkazovať používateľa na zdroj transakcie, lebo vieme, akého objektu je id použité v tabuľke. Dôvod transakcie je skôr informačného charakteru.

Implementácia a testovanie

Dátový model finálneho transakčného systému môžeme vidieť na Obrázku 57.



Obr. 57: Dátový model transakcií

Transaction sa ďalej napája na subtask, task a tabuľku User. úkážka pohľadu administrátora na zoznam transakcií v systéme.

Listing your transactions

From	To	Points	Reason	Status	Result	Description	Time
BioStore c-shop feedback	Frodo Pytlík	2	Task solving	Close	Success		2014-04-29 07:12:21 UTC

Obr. 58: Zoznam transakcií

3.13.3 Úprava načítavania úrovne jazyka z LinkedIn

Autor: Bc. Miroslav Molnár

Analýza

Bol zistený problém nekorektného načítavania dát z LinkedInu, konkrétne z načítavania úrovne jazyka. Pri preskúmaní problému bol zistený zly formát načítavania dát z tejto sociálnej siete. Pri testovaní sa taktiež narazilo na problém nevyplneného profilu. V prípade neexistencie dát aplikácia spadla.

Návrh

Pre správne fungovanie načítavania profilu z LinkedIn bolo nutné naštudovať si správny formát požiadavky na dáta. V našom prípade úroveň jazyka, ktorú používateľ používa. V pôvodnom formáte tieto dáta neboli získavané zo sociálnej siete. Ďalším problémom nutný vyriešiť bolo ošetrovanie prázdnych vstupov z LinkedIn.

Implementácia a testovanie

Prvou vecou bolo nutné zmeniť formát získavania dát z LinkedIn. V pôvodnej implementácii chýbalo získavanie úrovne jazyka. Následne získané dáta sa musia vložiť do databázy. Aby nenastával pád systému pri nevyplnenom profile na LinkedIn bolo nutné overiť existenciu dát ak dáta neexistovali tak sa nebudú spracovávať.

3.13.4 Vytvorenie dashboardu

Autor: Bc. Róbert Černý

Analýza

Dôležitou súčasťou systému Pinta.sk je zobrazovanie aktuálnych a dôležitých informácií používateľom. Základným pohľadom na stav aplikácie je možné dosiahnuť implementovaním dashboardu, kde budú používateľovi sprístupnené najdôležitejšie informácie. Hlavným cieľom je vyplňanie kampaní a poskytovanie odozvy, čo dokážeme zabezpečiť aj zobrazením zaujímavých kampaní priamo na dashboarde používateľa. Dashboard je prvá navštívená stránka po prihlásení do systému a preto musí byť prehľadná a poskytovať jednoduchý a elegantný spôsob zobrazenia jednotlivých elementov.

Návrh

zobraziť iba dôležité informácie rozdeliť stránku na bloky každý blok obsahuje informácie z jednej časti systému zobrazenie zaujímavých kampaní možnosť rýchleho filtrovania kampaní (najnovšie, skoro končiace, najpopulárnejšie) ponechať responzívny dizajn zobrazovať dosiahnuté ciele panel notifikácií zobrazenie aktuálneho kreditu používateľa

Implementácia a testovanie

Finálny návrh UI dashboardu používateľa a rozloženie elementov. Implementácia týchto elementov nebola náplňou tejto úlohy. Implementované bolo iba filtrovanie úloh a zobrazovanie kreditu používateľa.

3.13.5 Znovupoužitelnosť výsledkov v detaile tasku

Autor: Bc. Miroslav Vojtuš

Analýza

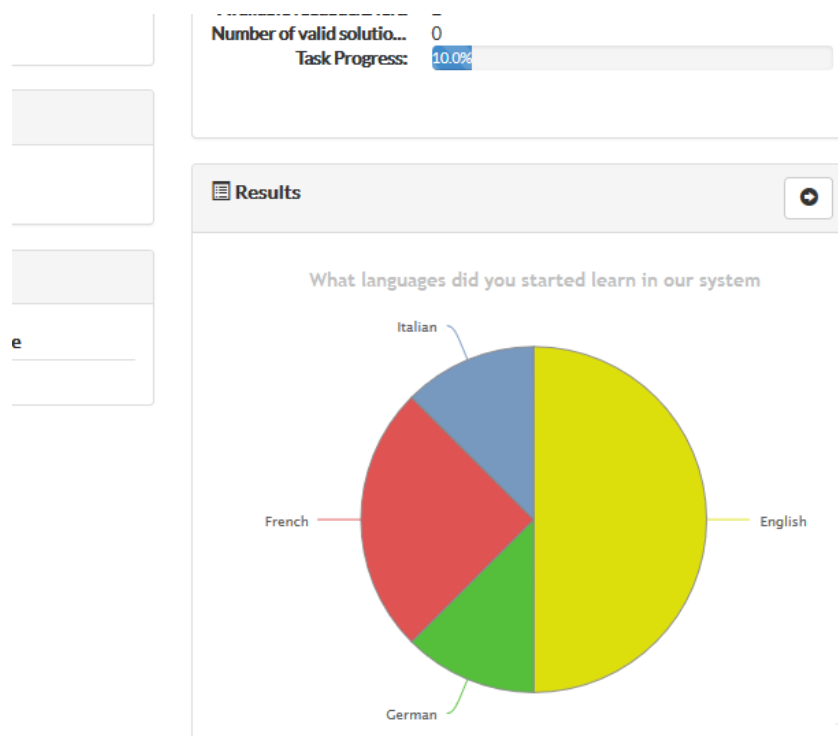
Aktuálne musí zadávateľ pri záujme o výsledk prejsť až 3-mi krokmi . Keďže počas prechádzania nie je nutné vidieť všetky možné výsledky ale len fragment pre zhodnotenie stavu riešenia. Aktuálna implementácia potrebovala zovšeobecniť pre možnosť znovu použitia. V aktuálnej verzii sú volania prepletené naprieč MVC.

Návrh

Skôr ako umožníme používateľovi vidieť viac ako je nutné, implementujeme len fragment, ktorý umožní zhliaďnúť len koláčový graf pre prvú úlohu. V prípade zvýšeného záujmu o túto funkcionality môžeme umožniť konfiguráciu zobrazených dát, prípadne slide-show variantu prezentácie výsledkov. Výsledný komponent musí byť konzistentný so štýlom ostatných, preto je nutné ho abstrahovať len na úrovni grafu. Štýlovanie ponechať na konkrétnom umiestnení.

Implementácia a testovanie

Implementácia korešponduje s návrhom. V detaile úlohy je zobrazený nový fragment s výsledkami, kde sa zobrazuje len jeden graf. Aktuálne je to napevno graf riešenej úlohy. Viď nasledujúci obrázok.



Obr. 59: Zobrazenie grafu pre viaceré výsledky

[Bug] Filtrovanie nefunkčné na produkčom databázovom serveri. Príznak: Pri pokuse o filtrovanie výsledkov na produkčnom serveri pomocou funkcie `sql juliandate()` spôsobilo chybovú hlášku.

Riešenie: Napriek zjavnej trivialite chyby bolo nutné inštalovať mysql na vyvíjanej platforme z dôvodu testovania.

Pôvodný kód:

```
1 find_filtered_solutions(["round(((julianday(Date('now')) - julianday(birth_date)))  
/ 365) BETWEEN #{from} AND #{to}"])
```

Opravený kód:

```
1 find_filtered_solutions(["round((strftime('%Y',Date('now')) - strftime('%Y',  
birth_date))) BETWEEN #{from} AND #{to}"])
```

3.13.6 Oprava chýb

Autor: Bc. Patrik Štrba V rámci tohoto šprintu bolo naplánovaná oprava chýb. Bolo potrebné vyriešiť nasledujúce chyby:

- Úprava úlohy duplikovala úlohu
- Refaktoring triedy TaskController
- Vytvorenie úlohy s typom survey nefungoval
- Integrácia so sociálnou sieťou facebook nefungovala
- Chyby pri vyplnení kedy cena za úlohy bola menšia než cena za vykonanie jednej podúlohy

Úprava úlohy duplikovala úlohu

Táto chyba sa vyskytovala ak sa používateľ pokúsil upraviť úlohu. Pri pokuse o úpravu úlohy mu vytvorilo novú upravenú namiesto toho aby upravilo starú.

Táto chyba vznikla pri spájaní kódu s ostatnými členmi tímu. Táto chyba bola opravená pridaním dodatočnej kontroly pri úprave úlohy.

Refaktoring triedy TaskController

Trieda TaskController bola obsiahla a obsahovala veľa riadkov kódu a preto bolo potrebné túto triedu refaktorovať. Refaktoringom vzniklo viacero tried, ktoré boli predtým zoskupené do jednej.

Vytvorenie úlohy s typom survey nefungoval

Pri vytváraní nového typu úlohy notice, vznikla chyba, ktorá sa vyskytovala pri nevyplnení úlohy v type úlohy survey. Táto chyba bola opravená kontrolou prítomnosti požadovaných vstupných dát. Integrácia so sociálnou sieťou facebook nefungovala Pri spájaní kódu s ostatnými členmi tímu jeden člen omylom nastavil metódu, ktorá obsluhovala túto integráciu za neprístupnú. Opravenie spočívalo v nastavení metódy na prístupnú.

Chyby pri vyplnení kedy cena za úlohy bola menšia než cena za vykonanie jednej podúlohy

Pri vytváraní alebo úprave úlohy, kedy bola zvolená cena za celú úlohu menšia ako cena za jednu podúlohu vyhadzvalo chybu. Táto chyba bola opravená pridaním chybovej hlášky v procese vytvárania a úpravy úlohy.

3.13.7 Global grid flow design

Autor: Bc. Filip Bednárík

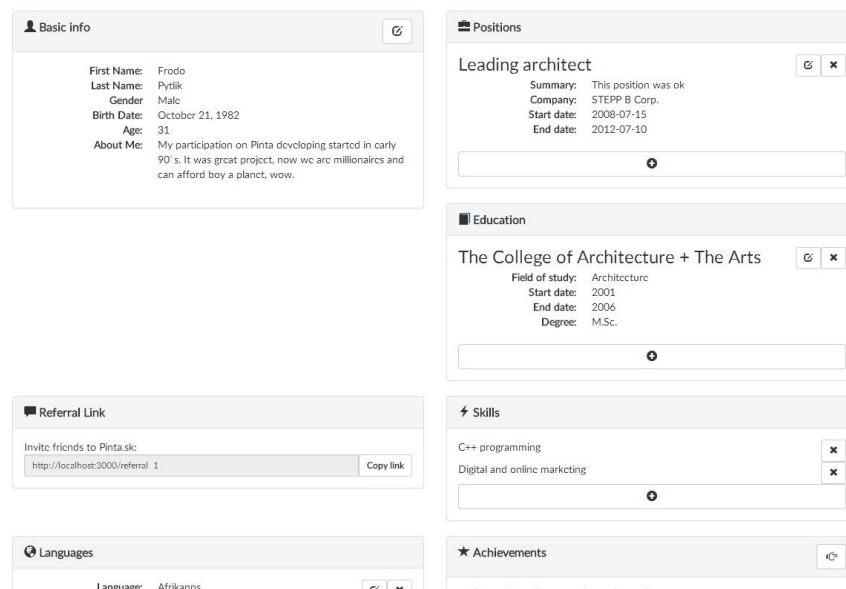
Analýza

Viacere stránky zobrazujú údaje rovnakým spôsobom - pomocou dlaždíc v mriežke, tieto stránky sú zobrazovanie ocenení, profilu a všetkých úloh. Neskôr môžu stránky so zobrazením údajov v mriežke pribúdať. Preto potrebujeme vytvoriť spoločný systém zobrazovania pre tieto stránky a tiež zabezpečiť korektné rozloženie sekcií v profile, zozname úloh a zozname ocenení.

Návrh a implementácia

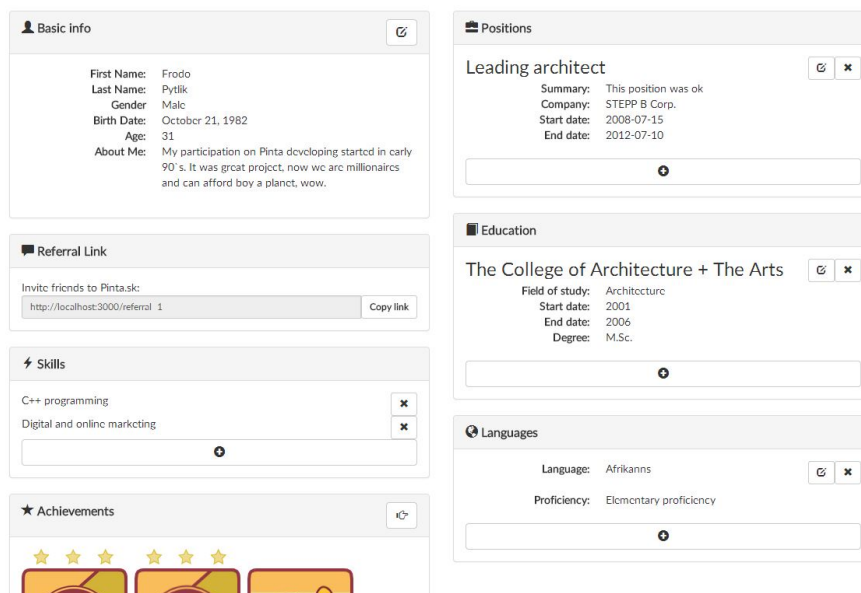
Na implementáciu sme použili masonry skript.

Ukážka pred implementáciou je znázornená na Obrázku 60.



Obr. 60: Mriežka pred implementáciou

Zobrazenie mriežky po implementácii je znázornené na Obrázku 61



Obr. 61: Mriežka po implementácii

3.13.8 Zobrazenie notifikácií na hlavnej stránke

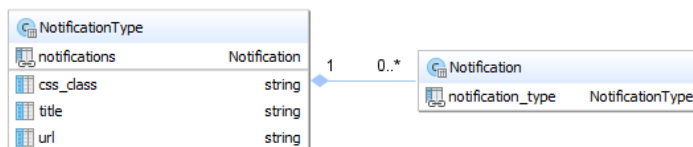
Autor: Bc. Filip Bednárík

Analýza

Pre dynamické zobrazenie aktivít používateľovi sme analyzovali možnosť vytvorenia notifikačného panelu, ktorý bude používateľovi zobrazovať udalosti v systéme, napríklad vytvorenie novej úlohy, úprava profilu, atď. Takto bude môcť používateľ sledovať čo sa v systéme deje a má možnosť priamo z hlavnej stránky sa dostať na aktivitu, ktorá je v notifikačnom paneli. Pre implementáciu navrhujeme využitie šablóny `sb-admin.css`

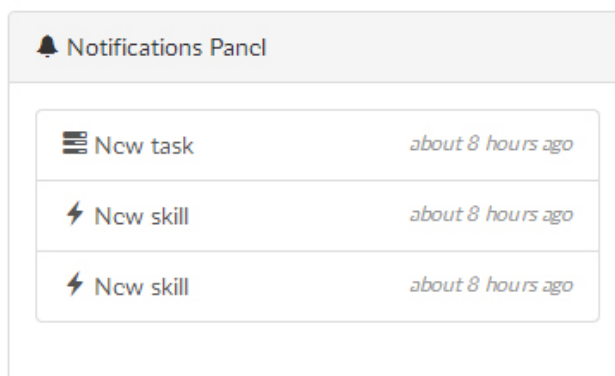
Návrh a implementácia

Návrh databázového modelu:



Obr. 62: DB model notifikácií

Grafický návrh:



Obr. 63: Grafický návrh notifikácií

Pre implementáciu sme využili nakoniec namiesto `sb-admin.css` už hotové Bootstrap triedy.

Ukážka kódu:

```

1 def notifications(limit = 0)
2   sqlString = "SELECT tasks.created_at as created_at, '1' as
      notification_type_id FROM tasks WHERE user_id="+self.id.to_s+"
3     UNION SELECT solutions.created_at as created_at, '2' as notification_type_id
      FROM solutions WHERE user_id="+self.id.to_s+"
4     UNION SELECT profile_skills.created_at as created_at, '3' as
      notification_type_id FROM profile_skills WHERE profile_id="+self.profile
      .id.to_s+"
5     ORDER by created_at DESC";
6   if (limit > 0)
7     sqlString+=" LIMIT "+limit.to_s;
8   end
9   Notification.find_by_sql(sqlString);
10 end

```

3.13.9 Doplnenie seedov do DB, tvorba dokumentácie a zrevidovanie obsahu doterajších zápisníc

Autor: Bc. Marek Lenčes

Analýza

Pre odhalenie početného množstva chýb v systéme potrebujeme mať vyplnených čo najviac dát v systéme, preto musíme vytvoriť dostatočný počet rozmanitých testovacích údajov, ktoré nám pomôžu s odhaľovaním. Tieto testovacie údaje musíme aj často meniť kvôli zmenám v databáze systému. Doplnili sme viacero testovacích údajov do databázy a zistili sme pritom viacero chýb v systéme, ktoré sme reportovali do systému JIRA.

Aktívne sme tiež tvorili dokumentáciu, opisy úloh v šprintoch a aktualizáciu časti riadenia. Zároveň sme skontrolovali obsah doterajších zápisníc a prípadné neúplné údaje alebo chyby sme opravili.

3.14 Šprint #13

3.14.1 Stretnutie s marketérkou, oprava chýb

Autori: Bc. Patrik Štrba, Bc. Martin Toma Vzhľadom na potrebu odovzdania nášho produktu v blízkej budúcnosti sme sa rozhodli konzultovať aktuálny stav s ľuďmi, ktorý by neskôr mohli byť naši potencionálni zákazníci. Náš produkt sme konzultovali s pracovníčkou marketingovej agentúry ale aj s odborníčkou na slovenskú start-up scénu a obchodné či ekonomické otázky v podnikaní.

Výsledkom týchto stretnutí bolo niekoľko podnetov a konštruktívnej kritiky, ktoré sme v ďalších šprintoch aplikovali pri plánovaní ďalších krokov. Medzi najdôležitejšie body ku ktorým sme dospeli po stretnutiach patrí:

- Treba sa zamerať na jednoznačnú cieľovú skupinu (v našom prípade študenti).
- Produkt je neskôr možné smerovať aj na malé a stredné firmy.
- Riešenie problému sliepka a vajce, formou výmenného obchodu získať prvých zadávateľov a riešiteľov. Možnosť osloviť študentov, rôzne konferencie a podobne.
- Implementovať iba základné typy kampaní a ďalšie pridávať podľa záujmu prvých zákazníkov.
- Upraviť obsah na webovej stránke aby bol napísaný jednoduchšou formou.

3.14.2 Internacionalizácia v2

Autor: Bc. Martin Toma

Analýza

Náš systém sme od začiatku cielili na širokú oblasť využitia. Aj z toho dôvodu sme sa už v zimnom semestri rozhodli zaviesť možnosť zobrazenia používateľského rozhrania vo viacerých jazykoch (slovensky, nemecky, anglicky). Zmyslom tejto úlohy bolo analyzovať nutnosť všetkých troch jazykov a následne zrevidovať použitie všade tam kde je to potrebné.

Návrh

Po analýze sme sa rozhodli úplne zrušiť podporu pre nemecký jazyk. V tejto fáze projektu nám prišlo zbytočné nákladné udržiavať tri jazyky. Rozhodli sme sa teda podporovať iba slovenský a anglický jazyk. V rámci toho sme refaktorovali prislúchajúce súbory en.yml, sk.yml. Nakolko v nich chýbala štruktúra.

Riešenie

Spomínané súbory sme prerobili tak, že každá systémová oblasť (profil, dashboard, odznaky a pod.) mala svoju sekciu a stromovú štruktúru nadpisov v slovenskej aj anglickej verzii. Následne sme v celom systéme zaviedli zobrazenie nadpisov cez funkciu `t` (translate).

3.14.3 Responzívny dizajn hlavnej stránky

Autor: Bc. Róbert Černý

Analýza

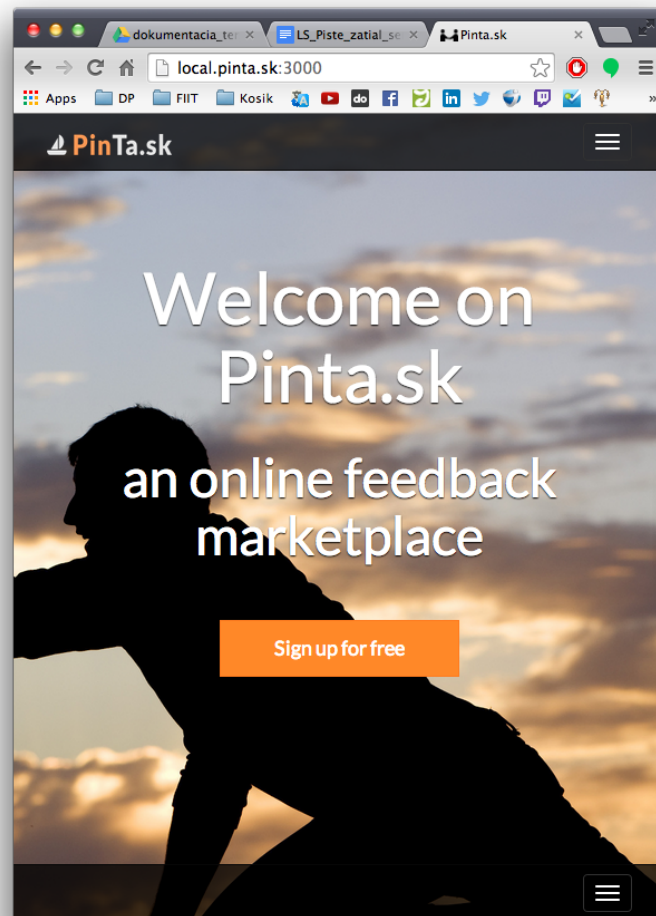
Naša aplikácia je založená na responzívnom dizajne. Takýmto spôsobom je možné vytvárať dizajn pre rôzne zariadenia naraz. Pre zachovanie konzistentnosti je potrebné prerobiť hlavnú stránku aplikácie. Jej HTML štruktúra je riešená pomocou tabuliek, čo je v dnešnej dobe považované za zlý postup. Hlavná stránka je najviac navštevovanou stránkou čo iba potvrdzuje nutnosť správneho zobrazenia na rôznych zariadeniach.

Návrh

zmeniť štruktúru html (TABLE nahradit DIV) zaviesť bootstrap a responzívne správanie

Implementácia a testovanie

Ukážka responzívneho dizajnu



Obr. 64: Responzívny dizajn

3.14.4 Unikátne dáta v profile

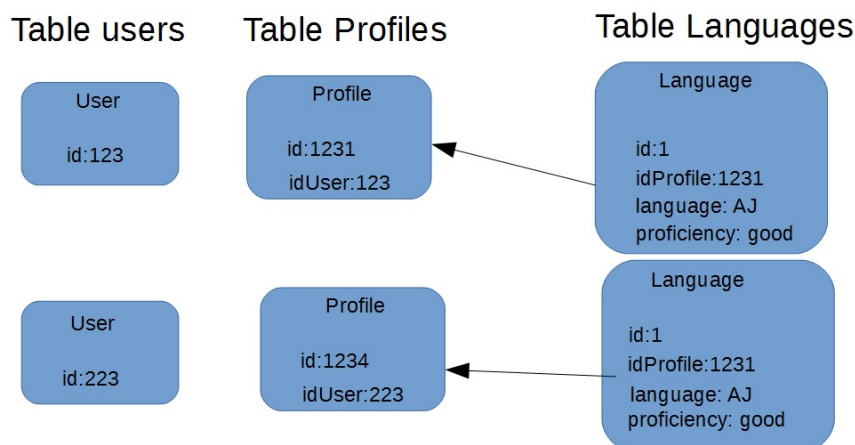
Autor: Bc. Miroslav Molnár

Analýza

Pri úprave dát z LinkedIn bolo detegované veľké množstvo duplicitných dát. Vyskytovalo sa to hlavne pri jazykoch a ich úrovniach. Rovnaký problém nastával pri zručnostiach. Na svete existuje niekoľko stoviek jazykoch, naša aplikácia by mala byť schopná udržiavať v databázy aj niekoľko stoviek tisíc používateľov. Je preto zbytočné aby sme pre každého používateľa zvlášť uchovávali jazyk a úroveň, keď očakávame, že sa v systéme budú nachádzať používatelia s rovnakým jazykom a úrovňou. Namiesto 1 záznamov v tabuľke budeme si uchovávať 1000 rovnakých jazykov. Takáto databáza je pomalá a zbytočne veľmi veľká. Rovnaký problém je aj u zručnostiach, kedy očakávame že používatelia majú rovnaké zručnosti a nie je potrebné vytvárať zbytočnú duplicitu záznamov.

Návrh

Je nutné prerobiť dátový model jazykov spolu s dátovým modelom zručností. Pôvodný dátový model bol vytvorený tak, že používateľ mal profil a každý jazyk s úrovňou zvlášť uložený ako vidíme na obrázku 65.

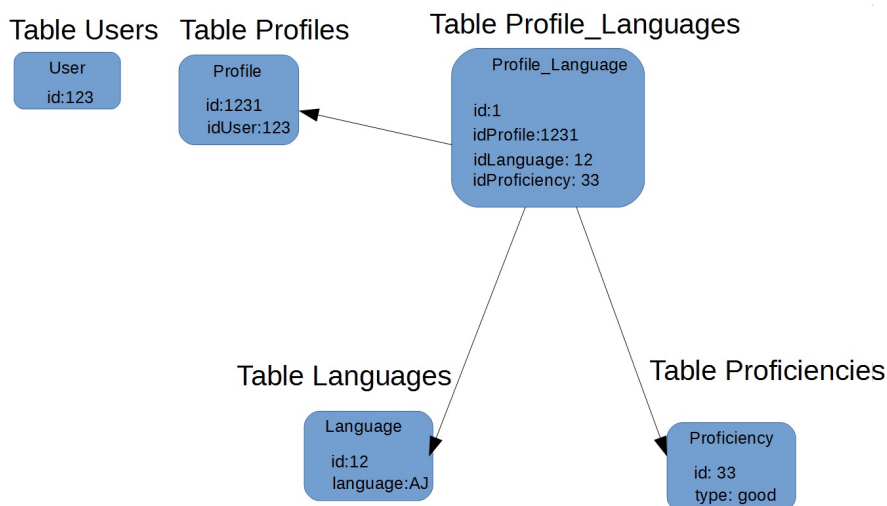


Obr. 65: Dátový model Jazykov

Ako vidíme na obrázku v tabuľke Languages boli uložené rovnaké záznamy pre rozdielnych používateľov. Takto sa vytvárala zbytočná duplicita kódu. Ďalším problémom je aj kombinácia hodnôt jazyka a úrovne. Rôzni používatelia môžu mať rovnaké úrovne a preto je nutné, aby neboli zbytočne previazané ale zvlášť v tabuľkách.

Implementácia a testovanie

Návrh nového dátového modelu môžeme vidieť na Obrázku 66.



Obr. 66: upravený dátový model Jazykov

Vytvorili sme nové tabuľky Profile_Languages a Proficiencies. Bola taktiež nutná úprava tabuľky Languages z ktorej sme zobrali hodnoty idProfile a proficiencies. Pre zmenu bolo nutné vytvoriť nad tabuľku s menom Profile_Languages zlučujúcu jazyky a ich úrovne. Používateľ sa odkazoval na túto tabuľku. Tým že sme oddelili jazyky od úrovne sme docielili ďalšiu normalizáciu celého modelu. Následne bolo nutné nastaviť unikátnosť záznamov pre jazyky a aj pre ich úrovne. To spôsobili, že sa rovnaká hodnota vyskytuje v tabuľke iba jedenkrát viac rázy nie. Pre aktualizáciu hodnôt jazykov využívame LinkedIn, z ktorej je dovolené uložiť aj jazyk, ktorý ešte v databáze nemáme. Používateľovi takúto možnosť neumožňujeme v profile. Je to kvôli ochrane údajov. Používateľ by mohol úmyselne vytvárať neexistujúce jazyky a tým pádom zapratávať databázu. Veľkou výhodou chránenia existencie dát je ich využitie pri nastavovaní riešiteľov k vyplneniu úlohy. Pri zručnostiach stačilo iba vytvoriť unikátnosť záznamu v databáze. Taktiež bolo nutné vytvoriť obmedzenie na dĺžku zadaného reťazca ktorá sa zhoduje s dĺžkou na LinkedIn. Jej hodnota je 80 znakov. Toto opatrenie tiež chráni systém pred útokmi používateľov.

3.14.5 Oprava chýb v systéme

Autor: Bc. Miroslav Vojtuš

[Bug] Csv-pridanie prípony, pomenovanie stĺpcov

Pri exporte csv súboru z výsledných grafov sa vyskytla chyba, súbor sa ukladal bez prípony .csv a stĺpce neboli riadne pomenované. Chybu sme odstránili.

[Bug] Pridávanie otázky v prehliadači Firefox. Vymazať sa dá len posledná otázka z viacnásobného výberu.

Pri pridávaní podúloh v Survey kampani sa vyskytla špecifická chyba v prehliadači Firefox, vymazať bolo možné len poslednú otázku z podúlohy Viacnásobný výber (Multichoice). Chybu sme úspešne odstránili.

3.14.6 Zlepšenie procesu vytvárania kampaní

Autor: Bc. Róbert Černý

Analýza

Pri vytváraní novej kampane sa po testovaní vyskytli niektoré problémy. Proces vytvárania má chyby v UI, ktoré používateľovi zhoršujú podmienky pri vyplňaní novej kampane. Hlavným cieľom tejto úlohy je opraviť chybu, kedy nebolo možné zrušiť vytváranie kampane, ktorá už bola uložená v databáze. Zadávatel tak nemohol vytvárať novú kampaň bez toho, aby dokončil už začatú.

Návrh

ukladať stav o nevyplnenej kampani ponúknuť zadávateľovi možnosť prerušiť vytváranie kampane odstrániť názvy polí vo formulári a použiť HTML 5 (placeholder)

Implementácia a testovanie

V prípade kampane ktorú zadávateľ nevyplnil celú sa do session dočasne uloží ID nedokončenej kampane. V prípade, že zadávateľ chce vytvárať novú úlohu, musí ukončiť predchádzajúcu alebo pokračovať v procese vyplňania.

Update this task or cancel it !

New Survey

Docasna kampan

asdas

Budget

12

Solution amount

1

Subtask cost

12

21-05-2014 14:35

Next Cancel

Obr. 67: Vytvorenie Survey kampane

3.14.7 Nové typy Survey podúloh

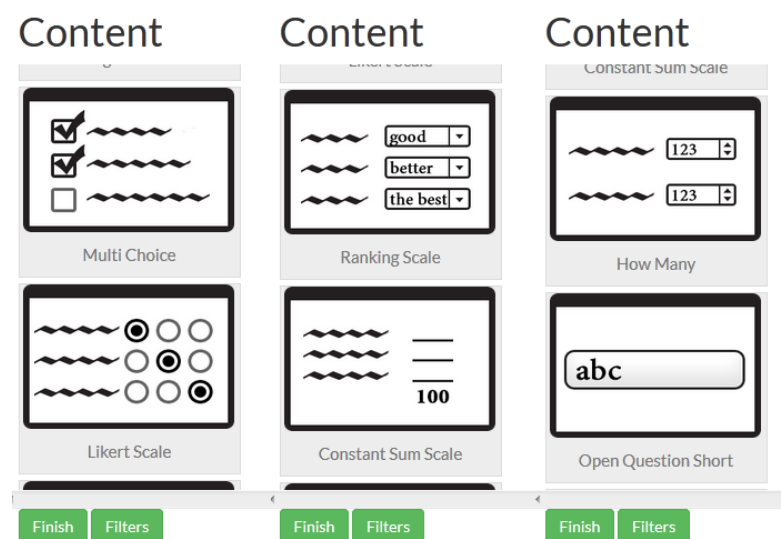
Autor: Bc. Marek Lenčes

Analýza

Pri vytváraní Survey kampane má zadávateľ k dispozícii zatiaľ 2 typy podúloh a to Otvorená otázka a Jednoduchý výber (Single-Choice). Cieľom je implementovať rôzne nové typy podúloh, ktoré by mohli zadávateľia využiť pri vytváraní svojho dotazníka. Keďže existuje veľký počet rôznych otázok, pre ktoré môže zadávateľ vyžadovať špecifický typ podúlohy, najlepšie by bolo implementovať čo najväčší počet podúloh, ktoré by zadávateľa nijako neobmedzovali pri vytváraní otázok.

Návrh

Po dôkladnej analýze a prieskume existujúcich možných typov otázok v dotazníkoch sme zistili, že existuje zopár najčastejších typov, ktoré dokážu pokryť veľké množstvo otázok, sú nimi: Viacnásobný výber (Multi-choice), Likertova škála, Hodnotenie výberom z dropdown menu (Ranking Scale), Hodnotenie súčtom pridelených bodov (Constant Sum Scale), výber číselnej hodnoty (How Many) a otvorené otázky s dlhou odpoveďou. Navrhli sme tieto typy otázok, ich grafickú podobu môžeme vidieť na Obrázku 68.



Obr. 68: Typy podúloh

Implementácia

Implementovali sme navrhnuté typy podúloh, proces ich vytvárania a pre každú sme vytvorili aj pohľad, ktorý vidí riešiteľ pri riešení úlohy. Proces pridávania otázok do konkrétnych typov podúloh je znázornený na Obrázku 69.

Question

Test question

Scale range

10

Choices

Subquestion 1

Subquestion 2

Subquestion 3

+

Done

Obr. 69: Pridanie otázok

Príklady pohľadov vytvorených podúloh sú zobrazené na Obrázku 70 a Obrázku 71.

What do you about task solving process ?

Select how easily recognizable was these part in task solving view. (5 is most easily recognizable)

Task question	1	5
Task options	1	5
Task deadline	1	5

Obr. 70: Likertova škála

3.14.8 Hall of Fame

Autor: Bc. Filip Bednárík

What are your first impressions of Pintask?

Select each process complexity

registration	easy
navigation	average
orientation	complicated

Next step

Obr. 71: Dropdown ranking

Analýza

V záujme väčšej motivácie používateľov chceme vytvoriť zoznam najaktívnejších používateľov v systéme čo sa týka počtu úspešne vyriešených úloh. Zároveň poskytneme používateľom možnosť odmien a sledovanie postupu v rebríčku.

Návrh a implementácia

Návrh rozhrania pre zobrazenie rebríčka najaktívnejších používateľov je znázornený na Obrázku 72. Je implementovaný pomocou gemu counter_cache.

#	Username	Number of solutions
1	Robert Zombie	2
2	Frodo Pytlík	1
3	Thomas Davids	0
4	Elias Schmidt	0
5	Michal Baláž	0
6	Lex Luthor	0
7	Melissa McDouglas	0
8	Sophia Reinbold	0
9	Federica Alesini	0
10	Mashito Hawakukoti	0

Obr. 72: Stĺpcový graf

3.14.9 Nastavenia pre používateľa a zobrazovanie úloh

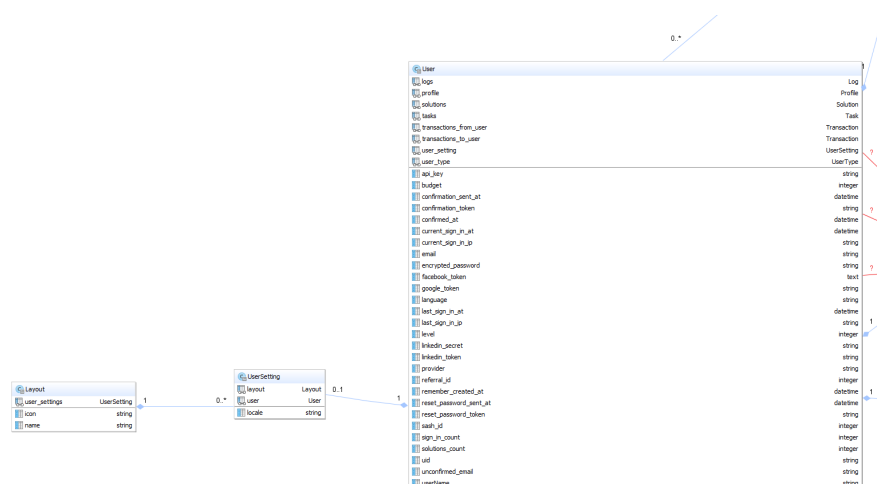
Autor: Bc. Filip Bednárík

Analýza

Používateľ potrebuje v aplikácii rôzne nastavenia. Každý používateľ má iné preferencie zobrazovania úloh. Bolo by vhodné pridať možnosť zmeniť rozloženie a personalizovať stránku podľa predstáv používateľa. Ďalšia funkcia by mohla byť možnosť jednoduchého pridávania nových layoutov.

Návrh a implementácia

Návrh DB modelu je na obrázku 73. Tlačidlo pre zmenu layoutu je globálne, zmení



Obr. 73: DB model používateľských nastavení

sa nastavenie pre konkrétneho používateľa. Možnosti zobrazenia sú buď v mriežke alebo v zozname. Každý layout je implementovaný samostatným pohľadom.



Obr. 74: Zmena layoutu

3.15 Šprint #14

3.15.1 IIT.SRC, Úprava homepage obsahu

Autor: Bc. Martin Toma

Analýza

Po stretnutí a konzultácii projektu sme dostali ohlasy, že obsah na prezentačnej stránke Pinta.sk je zložitý a nie je z neho jasné o čo sa jedná. Preto sme sa rozhodli zrevidovať tento obsah a upraviť tak aby lepšie vystihoval podstatu projektu.

Návrh

Navrhli sme znížiť počet slov v jednotlivých blokoch a použiť jednoduchšie a výstižnejšie výrazy. Rovnako je potrebné jednoznačne oddeliť ktorý workflow patrí zadávateľovi a ktorý riešiteľovi kampaní.

Implementácia a testovanie

Na obrázku nižšie vidíme finálnu verziu anglickej verzie našej prezentačnej stránky. Konkrétne ide o workflow pre zadávateľa kampaní. Tieto tri kroky by mali vysvetliť čo sa dá v našom systéme robiť. Vysvetľujeme, že zameranie cieľovej skupiny, validácia odpovedí a ich následná grafická reprezentácia je gro našej aplikácie.

Okrem toho sa v tomto šprinte pracovalo na úlohách, ktoré súviseli so študentskou vedeckou konferenciou IIT.SRC 2014. Pracovalo sa na vytvorení rozšíreného abstraktu ale aj na návrhu plagátu a podobných záležitostiach.



Obr. 75:

3.15.2 Umožniť používateľovi ovplyvniť vzhľad textu otázky

Autor: Bc. Miroslav Vojtuš

Analýza

V súčasnej verzii je otázka v úlohe typu notice zobrazená ako jednoduchý neformatovaný text. V prípade komplexnejších zadaní problému, používateľ potrebuje vkladať obrázky prípadne odkazovať na iné stránky súvisiace s riešením. Taktiež neformatovaný text pôsobí metúco a je neprehľadný.

Návrh

Pre minimalizáciu rizika injekcie do databázy, prípadne cross site scripting, sme sa rozhodli využiť technológiu mark down. Výhodou je, že zdrojový text je formátovaný len za pomoci jednoduchých značiek, ktorých počet je obmedzený podľa implementácie processora. Do markdown processora je možné doimplementovať aj vlastné značky. Výsledný dokument je formátovaný (markup - html) text. Do textu je taktiež možné vkladať obrázky, odkazy, prípadne fragmenty kódu. Výstupný text neobsahuje, žiadne tagy ktoré by bolo nutné filtrovať, prípadne povoľovať na úrovni textu.

Implementácia a testovanie

Na nasledujúcom obrázku je pôvodná verzia komplexnejšieho textu bez formátovania. Po aplikácii markdown processora:

3.15.3 Refaktorovanie jadra aplikácie

Autor: Bc. Róbert Černý

Analýza

Pri implementácii jadra našej aplikácie (task_controller a súvisiace súbory) boli naše znalosti RAILS 4 veľmi obmedzené. V súbore sa nachádza správanie, ktoré logicky nezapadá do danej oblasti. Niektoré metódy je dobré presunúť podľa konvencií, ktoré sa používajú v RAILS. Najväčším problémom je vyššie spomínaný súbor, ktorý obsahuje okrem logiky kampane aj logiku pre zobrazovanie výsledkov, nahrávanie súborov a rôzne metódy. Je nutné refaktorovať aplikáciu a aplikovať metódy ktoré sa používajú pri písaní RAIL 4 aplikácií.

Otestuj aplikáciu APP

Zo stránky <http://www.example.com/app/download.file> si stiahni aplikáciu a nainštaluje si do svojho prostredia.

Steps:

1. stiahni aplikáciu
2. rozbaľ na príslušné miesto `C://<your place>/app.file`
3. Na určené miesto vlož nasledujúci fragment kódu:

```
...  
import("app");  
app.start();  
...
```

Next step

Step: 1 / 1

Campaing deadline: May 19, 2015

Obr. 76:

Návrh

prepísať zdrojový kód v `task_controller` podľa konvencií presunúť metódy do modelu kampane minimalizovať veľkosť kódu a zlepšiť prehľadnosť zmazať nepotrebné CRUD metódy vytvoriť nový model pre typy kampaní, výsledky, prácu so súbormi

Implementácia a testovanie

Pomocou nástroja RubyMine bolo možná automatická detekcia chýb v zdrojovom kóde. Nepotrebné bodkočiarky, zmena logiky pri vetvení programu, nepotrebné zátvorky. Okrem týchto vecí sa presunula logika do nových modelov. Tím bol upozornený na písanie novej logiky do modelu a nie do kontrolera danej triedy. Výstupom bola redukcia zdrojového kódu z 472 na 113 riadkov čitateľného kódu.

3.15.4 Vytvorenie funkčných a unit testov

Autor: Bc. Patrik Štrba

Analýza

V rámci projektu sme doteraz testovanie moc nevyužívali. Projekt bolo potrebné prezentovať a preto sme potrebovali zabezpečiť že úpravy uskutočnené na aplikácií ju nepoškodili a aplikácia funguje ako má.

Preto bolo potrebné vytvoriť sadu testov, ktorými musí aplikácia prejsť aby bola pripravená na nasadenie na produkčný server.

Návrh

Najskôr bolo potrebné identifikovať dôležité operácie, ktoré sú kritické našu aplikáciu. Ako kritické operácie som zvolil vytváranie úlohy, prihlásenie a registrácia.

Otestuj aplikáciu APP

Zo stránky si stiahni aplikáciu a nainštaluje si do svojho prostredia.

Steps:

1. stiahni aplikáciu
2. rozbaľ na príslušné miesto `C://<your place>/app.file`
3. na určené miesto vlož nasledujúci fragment kódu:

```
...
import("app");
app.start();
...
```

Write answer...

Next step

Obr. 77:

Medzi dôležité operácie patrí aj vyplňanie úlohy, ale to nebolo nakoniec realizované z dôvodu nefunkčnosti.

Implementácia a testovanie

V rámci implementácie bolo vytvorených viacero prípadov, ktoré zahŕňali testy.

Implementácia a testovanie

Pridávanie úlohy V rámci testovania pridávania úlohy bol vytvorený testovací prípad, ktorý je potrebný pre úspešné zbehnutie.

Implementácia a testovanie

V rámci registrácie bolo vytvorených viacero testov. Testy zahŕňali prípady kedy boli zadávané údaje v nesprávnom formáte, nekonzistentné heslá a nakoniec aj jeden test na registráciu, kedy má registrácia skončiť úspešne.

Prihlásenie

V rámci prihlásenia boli vytvorené dva základné testy kedy sa testovalo na úspešné zadané prihlasovacie údaje a na nesprávne prihlasovacie údaje. Ďalšie dva testy boli vytvorené, kedy sa jeden z údajov nezadali.

3.15.5 Validácia nových typov úloh a ukladanie výsledkov

Autor: Bc. Marek Lenčes

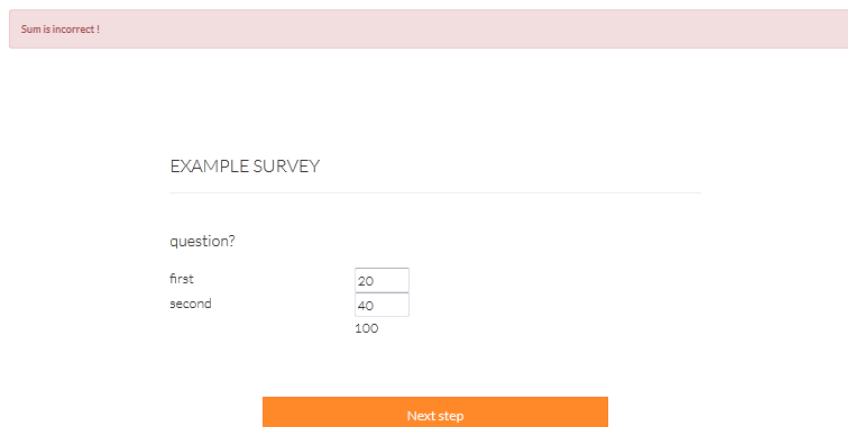
Analýza

Pri vytvorených typoch podúloh je možné vkladať nevalidné vstupy, napríklad výber žiadnej z možností, nevyplnenie otvorenej otázky, súčet súm sa nerovná celkovej sume pri podúlohe konštantnej sumy a podobne. Musíme odhaliť všetky

možnosti vloženia nevalidného vstupu a vytvoriť validačné podmienky. Následne musíme vhodne uložiť údaje pre neskoršie zobrazovanie pomocou grafov výsledkov.

Návrh a implementácia

Pre každú z podúloh definujeme podmienku, pri ktorej nesplnení sa používateľ nemôže dostať na ďalšiu otázku, zabezpečíme tým možným kolíziám v databáze a pri vyhodnocovaní odpovedí. Pri nevalidnom vstupe sa používateľovi zobrazí príslušná správa. Príklad vidíme na Obrázku 78.



Obr. 78: Stĺpcový graf

3.15.6 Autocomplete

Autori: Bc. Filip Bednárík, Bc. Miroslav Molnár

Analýza

Pre zjednodušenie vkladania údajov do niektorých textových polí v profile potrebujeme na niektoré polia použiť možnosť automatického dopĺňania obsahu, napríklad pridanie krajiny, jazyku alebo zručnosti.

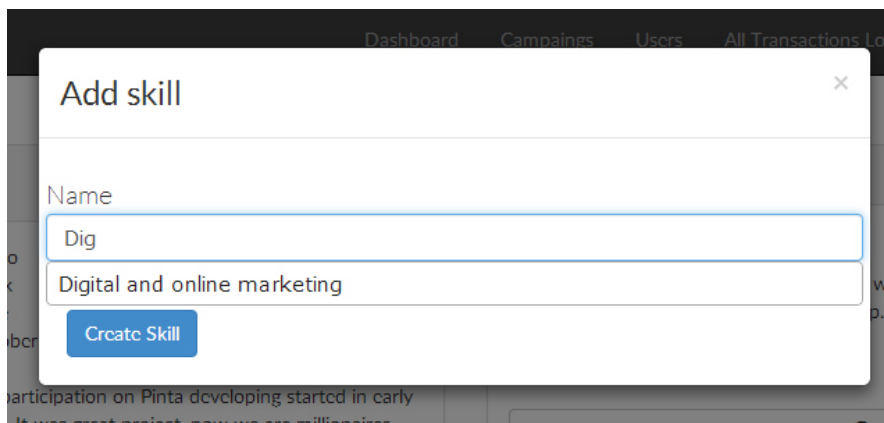
Návrh a implementácia

Používateľ pri zadávaní novej zručnosti do profilu začne písať zručnosť a následne sa mu zobrazia už existujúce zručnosti z databázy. Používateľ si môže vybrať jednu z nich alebo si vytvorí vlastnú zručnosť. Hlavnou výhodou je, že nevznikajú neustále rozdielne zručnosti, ale používatelia sa zoskupujú do jednej konkrétnej. Následne filtrovanie používateľov na základe ich zručností je jednoduchšie. Je možné pomocou jednej zručnosti osloviť viacerých používateľov. Autocomplete zabraňuje taktiež preklepom. Používateľ ak vidí, že znalosť existuje, zvolí si ju a nemusí sa pomýliť. Autocomplete bol uplatnený aj na Jazyky v profile. Návrh využitia sa ale líši. Pri nenájdení jazyku v databáze používateľ nový jazyk nemôže vytvoriť. Môže si zvoliť iba z existujúcich jazykov. Bolo preto potrebné načítať do databázy všetky používané jazyky.

Návrh automatického dopĺňania je na Obrázku 79.

Implementácia

Pri implementácii sme použili hotový gem autocomplete, ktorý ale nepodporoval modálne okná, ktorých podporu sme doplnili do systému.



Obr. 79: Autocomplete

Pre fungovanie bolo nutné upraviť hodnoty v súbore routes.rb, ako vidíme nižšie:

```
1 # Skills
2   resources :skills do
3     get :autocomplete_skill_name, :on => :collection
4   end
```

Následne bolo treba vložiť vykonávanie gemu do príslušného kontrolóra, kde sa udávalo aká hodnota v databázy sa bude automaticky prehľadávať:

```
1 autocomplete :skill, :name
```

V poslednom páde bolo treba v vložiť aj gem do html kódu príklad nižšie:

```
1 <%= f.autocomplete_field :name, autocomplete_skill_name_skills_path, {class: 'form
  -control', :maxlength => 80, :append_to=> '#skill_autocomplete'} %>
```

Takto nastavený gem vyhľadáva po zadaní začiatkových znakov automaticky existujúce hodnoty v databázy.

3.16 Šprint #15

3.16.1 Oprava identifikovaných chýb v systéme

Autor: Bc. Martin Toma

Tento šprint sme skoro celý venovali detekcii a oprave chýb v systéme. Nižšie je uvedený zoznam chýb, ich riešenie a prípadný obrázok.

Syntaktická chyba pri zobrazení preloženého nadpisu. Riešením bola oprava syntaxe. Zbytočne zobrazované tlačidlo “solve” v prípade vlastnej kampane. Riešením bola náhrada za “not eligible” tlačidlo. Nezobrazovanie všetkých v systéme implementovaných typov kampane. Riešením bolo korektné zobrazenie všetkých typov.

Obr. 80:

You are not able to solve your own task !

Active campaigns

Name	Description	Category	Points for solving	Deadline	Action
Rate TreX vacuum Cleaner	Please take our quick survey to assess your satisfaction with our new product TreX vacuum cleaner. We look forward to reviewing your responses.	Survey	5	2014-04-15 12:00:00 UTC	Solved
Testdsadasdsds	Testdsdaads	Survey	10	2014-04-15 00:10:00 UTC	Solve

Obr. 81:

3.16.2 Automatické uzamykanie kampaní (CRON)

Autor: Bc. Róbert Černý

Analýza

Pri vytváraní novej kampane je možné zadať dátum ukončenia kampane. Po tomto dátume je potrebné kampaň uzamknúť a znemožniť tak jej ďalšie riešenie.

Návrh

použiť gem WHENEVER ktorý zjednodušuje prácu s RAILS CRON vytvoriť metódu pre terminovanie vypršaných kampaní automatické spúšťanie každých 30 minút ukladať výstupy do logu

Implementácia a testovanie

úkažka kódu ktorý rieši danú úlohu, využitý bol gem WHENEVER

```

1 set :environment, 'production'
2 set :cron_log, '/log/cron_log.log'
3 every 30.minutes do
4   runner 'Task.terminate_expired'
5 end

```

3.16.3 Rozdelenie seedov

Autor: Bc. Róbert Černý

Analýza

Pri rozširovaní dátového modelu vznikla potreba rozdeliť seedy našej aplikácie na dve množiny. Základné seedy, ktoré sú potrebné pre produkčnú verziu systému a seedy pre vývojársku verziu.

Návrh

rozdelenie seedov do 3 súborov common.rb (typy používateľov, krajiny, typy kategórií, filtre, seedy pre profil, jazyky ...) development.rb (testovací používatelia, kampane, otázky dotazníka, profily ...) production.rb (vytvorenie administrátora) zabezpečenie načítania konkrétnych seedov pre prostredie produkčné a vývojárske prepísanie seedov do súboru do ktorého patria

Implementácia a testovanie

Ukážka zdrojového kódu ktorý zabezpečí načítanie základných seedov a seedov pre konkrétne prostredie.

```
1  
2 load(Rails.root.join( 'db', 'seeds', "common.rb"))  
3 load(Rails.root.join( 'db', 'seeds', "#{Rails.env.downcase}.rb"))
```

3.16.4 Oprava chýb v systéme

Autor: Bc. Filip Bednárík

[Bug] Verejný profil používateľa

Problém:

Verejný profil používateľa zobrazuje viac ako je vhodné

Username:
Password:
Email: drndos@drndos.sk
Firstname: Filip
Lastname: Bednárík
Country: Slovak Republic
[Back](#)

Obr. 82: Verejný profil pred

Riešenie:

Odstránenie informácií z pohľadu, aplikovanie bootstrap tried na pohľad.



Firstname: Frodo
Lastname: Pytlík
[Back](#)

Obr. 83: Verejný profil po

Terminating inactive task shouldn't give user points and Changing task budget of not active task should have affect on my credit balance

Problém:

Terminácia nekatívnej úlohy by nemala pripočítat body používateľovi.

Riešenie:

Rozhodli sme sa, že body sa budú odpočítavať až po aktivovaní úlohy. Pridal som podmienky do transakčného systému.

Activation of task with no subtasks

Problém:

V systéme bolo možné aktivovať úlohu, ktorá neobsahovala žiadne podúlohy.

Riešenie:

Pridal som podmienku, aby to nebolo možné.

After click to Show tasks from table representation to Dashbord return "nil can't be coerced into Float"

Problém:

Pri zobrazovaní dashboardu úlohy sa vyskytovala chyba.

Riešenie:

Chyba sa vyskytovala kvôli chybným vstupným datam. Opravili sme seedy.

Filters: I can destroy filter of active task.

Problém:

Aktivovaná úloha by sa nemala dať upravovať. Po aktivovaní úlohy by sa nemali dať upravovať filtre úlohy.

Riešenie:

Pridal som podmienku aby nebolo možné pristupovať do filter controllera ak je task v inom ako neaktívnom stave.

Profile: age update

Problém:

Fails on unknown formatting. Used datepicker changes date format. System allows to save it, but is not abble to show it (FAILS)

Riešenie:

Zmena metódy strftime na metódu, ktorá funguje aj s MySQL databázou.

Search button in all tasks problem

Problem:

Nefungovalo vyhľadávanie pomocou AJAXu.

Riešenie:

Pri refactoringu niekto vymazal súbor s javascriptom obsluhujúci pole pre vyhľadávanie. Obnovil som tento súbor.

Notifications panel is flowing ower screen.

Problem:

Pri väčšom počte notifikácií tieto pretekali mimo obrazovky.

Riešenie:

Obmedzil som počet zobrazovaných notifikácií na 10.

Workers age filter with non-numeric value

Problém:

Filtre nekontrolujú hodnotu.

Riešenie:

Prerobenie systému filtrov, v ďalšom šprinte.

Homepage register is not working

Problem:

After filling registration form and submit, nothing happens

Riešenie:

Chyba v povinných poliach používateľa. Odstránil som polia a opravil tým chybu.

Task description in user dashboard is overflowing

Problem:

Pri väčšom počte znakov v popise úlohy popis pretekal v dashboarde.

Riešenie:

Malá úprava css, pridanie truncate pre popis.

Pri testovaní aplikácie som identifikoval väčšie množstvo chýb. Tu sú prehľadne napísané v tabuľke:

User firstname and lastname handling
Public Profile of user is kinda bad
User shouldnt see solve button on his own task
Terminating inactive task shouldn't give user points
Deadline clears after error message is shown
Education action buttons are little off
En localization bug in languages
If position doesn't have End date position is never saved
Autocomplete suggestion box is on the bottom of the page at first
After user is successfully registered he is not notified about it
Skill name should be unique and use autocomplete
Languages table is not correctly designed
Achievements modal windows are not working
Changing task budget of not active task should have affect on my credit balance
Edit in task dashboard is link to bad edit method
Creating notice still doesn't work
We should allow destroying task only for admin other can only terminate task
Buttons in profile edit page are confusing
CSV file doesn't contain headers, hard to tell which column is what
CSV file doesn't have .csv extension
Results are not showing in task dashboard
Image for category is missing
Seeds are still not presentable
Task description in user dashboard is overflowing
Footer is overlapping the content
responsive design is not working on homepage
Solved and success tasks in profile are not counting

3.16.5 Oprava chýb

Autor: Bc. Miroslav Vojtuš

Hlavnou náplňou tohoto šprintu bolo identifikovanie chýb a oprava najzávažnejších. Tieto chyby som identifikoval. Následne nám boli pridelené Martinom T.

Transactions: Double points removal or double check
Transactions: Activating task doesn't affect budget
Campaign: with deadline passed can be solved.
Error messages: AWFUL/INCONSISTENT
Can create campaign in past
Filters: I can destroy filter of active task.
Campaign solve: Error if i try to solve inactive task.
Results: filtering broken, obsolete plus button
Results: newly created subtasks
Profile: update button
Unlimited textarea.
Profile: age update
Trasnsaction log: NoMethodError
Hall of fame: every one has zero tasks solved
Useless actions in wallets dropdown.
No ui element available to return on home page.
Footer links lost
Forbid showing/solving of own campaigns.
Notifications panel is flowing ower screen.
Account details cancelation behaves wierd
Notice subtask title in results
Budget is not affected even after campaign solved.
Wierd continue buton on already solved task
Continue button clicked and received NoMethod Exception
User shouldn't be able to return to make new solution to subtask by pressing back button.
Step is not incremented
Trimming of titles in campain grid layout.
Focus of cursor on after login modal shown.
Hide edit buttons if task in active mode.
Pagination is not bookmarkable.
Budget is not affected even after task activation.
Linked in - 404

3.16.6 Nasadzovanie aplikácie na produkčný server a riešenie chýb s tým spojených

Autor: Bc. Patrik Štrba

V rámci tohto šprintu bolo potrebné nasadiť všetky nové úpravy nasadiť na produkčný server.

Nasadenie prebiehalo vo viacerých krokoch, kedy bolo potrebné upravovať vzniknuté chyby.

Nasadenie prebiehalo spojením v repozitári z vetvy devel, kedy sme zmeny prevedli do vetvy master. Následne repozitár oznámil produkčnému serveru, že ob-

sahuje novú verziu a produkčný server si túto verziu stiahol a nasadil.

Pri nasadzovaní vznikali chyby, ktoré sa z rôznych príčin na vývojovej verzii nevyskytovali. Jednou z najčastejšou chýb bola tá, že na vývoj sme používali databázu SQLite a na produkčnom serveri databázu MySQL.

Ďalšie chyby, ktoré bolo potrebné opraviť boli najčastejšie spôsobené načítavaním skriptov, ktoré sa inak načítavajú lokálne a inak z produkčného servera.

V rámci nasadenie bolo často potrebné vyprázdniť databázu a vytvoriť nanovo. Toto bolo spôsobené nevhodnými úpravami dátového modelu a zasahovanie do migračných skriptov počas vývoja.

3.16.7 Oprava chýb

Autor: Bc. Miroslav Molnár

Našli sme a opravili nasledujúce chyby v systéme:

Task info special valur "TASK Progress"
After click to Show tasks from table representation to Dashbord return "nil can't be coerced into Float"
Dashboard miss icon
Bad localization SK in Moj ucet breaking button layout
Bed version Rails problem with GEM TZinfo integer valuer represented Date in Education.
Tasks missing icon in task notice campane
Change buttons possitions in profile languages.
Worker work task created notice where is missing textbox of Data.

3.16.8 Oprava chýb

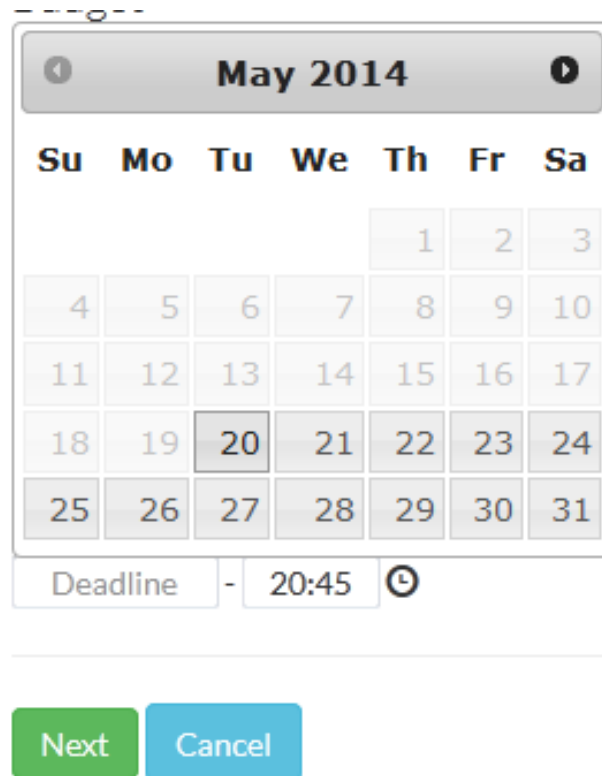
Autor: Bc. Marek Lenčes

Našli sme v systéme nasledujúce chyby a odstránili sme ich.

Time field is editable
Referral text field is editable
Profile windows float by their will
Cost of subtask can be higher than whole assigned budget
All tasks grid view broken
Empty notice task can be created
Text overflows browser window in Profile
Language Proficiency label is mistyped
Skill name can be 1 billion chars long, overflowing when too long
Update button for Positions and Education in profile doesn't work
User can create blank Position/Education/Skill/Language
User can set earlier end date than start date in Profile Positions and Education
Deadline is always set to current date, no matter what date requester selects
After trying to update task, Deadline field contains Date + Time, should be only Date

Pri pridávaní hocijakej úlohy sme zistili závažný problém, že používateľ môže zadať starší dátum začiatku úlohy ako je dátum momentálneho dňa. Preto sme museli

pridať validáciu na časové pole. Pre implementáciu validácie sme použili gem *timeliness*, ktorý umožňuje validovať dátumové polia a zistiť tiež sice syntakticky správne, no neexistujúce dátumy (30 Feb.). Tiež sme pridali nový DatePicker a TimePicker, ktoré umožňujú pohodlnejší výber dátumu a času. Nový DatePicker je znázornený na Obrázku 84.



Obr. 84: DatePicker

Na Obrázku 85 vidíme v poslednom riadku, že validácia prebehla nad dátumovým polom a nedovolila zadať skorší dátum ako dátum vtedy aktuálneho dňa.

New Survey

- Name can't be blank
- Name is too short (minimum is 5 characters)
- Description can't be blank
- Description is too short (minimum is 5 characters)
- Solution amount can't be blank
- Subtask cost can't be blank
- Budget can't be blank
- Deadline must be after 2014-05-19 00:00:00

Name

Write description here ...

Obr. 85: Validácia dátumu a času

3.17 Šprint #16

3.17.1 Robime.IT článok, UI vylepšenia a bugfixy

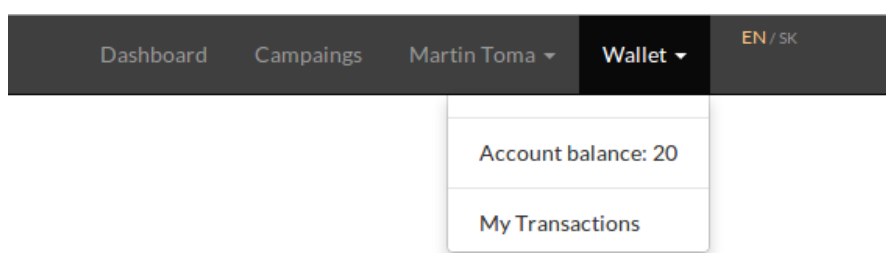
Autor: Bc. Martin Toma

Analýza

V systéme sa vyskytovalo veľa nekonzistentných názvov, akcií, farieb a fontov. Okrem toho bol aktuálny spôsob prepínania nastavenia jazyka v kontraste s celkovým dizajnom. Rovnako z neho (anglická a slovenská vlajka) nebolo jasné, ktoré nastavenia je aktívne. V menu peňaženka bolo viacero nejednoznačných odkazov. V okne úprava profilu bolo zle riešené rozmiestnenie niektorých elementov.

Implementácia a testovanie

V prípade názvov sme zvolili tie najvystížnejšie. Farby a fonty sme si určili a od tejto chvíle sme používalie práve tie. Prepínač jazyka sme zmenili tak, že vlajky boli nahradené skratkami EN/SK s tým, že aktuálne aktívne nastavenie bolo zvýraznené. Novú verziu panelu aj s revidovaným menu peňaženka (wallet) vidíme na obrázku 86.



Obr. 86:

Oprava chýb

- Oprava nefunkčného zrušenia úpravy používateľského účtu.
- Automatická aktualizácia LinkedIn profilu po jeho spojení.
- Statický blok informácií o prihlásenom používateľovi zobrazovaný len v debug móde.
- Nastavenie maximálnej dĺžky niektorých nadpisov a položiek v profile.

Okrem toho bolo tento šprint potrebné vytvoriť prezentáciu aktuálneho stavu projektu a jeho cieľov do budúcnosti pre webový portál Robime.it. Článok je dnes dostupný na adrese <http://robime.it/tp-cup-pinta-sk-ziskajte-spatnu-vazbu-na-cokolvek/>

3.17.2 Pridanie nových typov achievementov. Vytvoriť používateľsky prívetivejšie obrázky

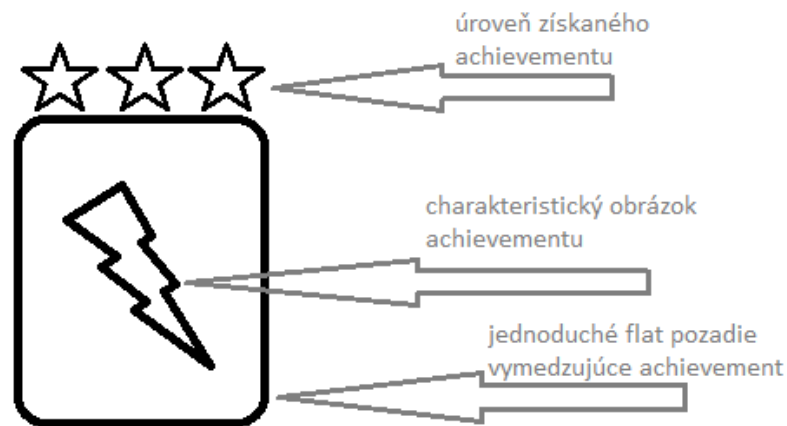
Autor: Bc. Miroslav Vojtuš

Analýza

Pôvodná verzia achievementov v ranných šprintoch tohoto semestra bola prezentovaná potenciálnim používateľom. Konkrétne marketingovým študentkám a pracovníčkam. Podľa nich boli achievementy príliš odlišné od designu a taktiež ťažko pochopiteľné. Odporučili nám zjednodušiť ich a využiť princípy bežne používané naprieč internetom.

Návrh

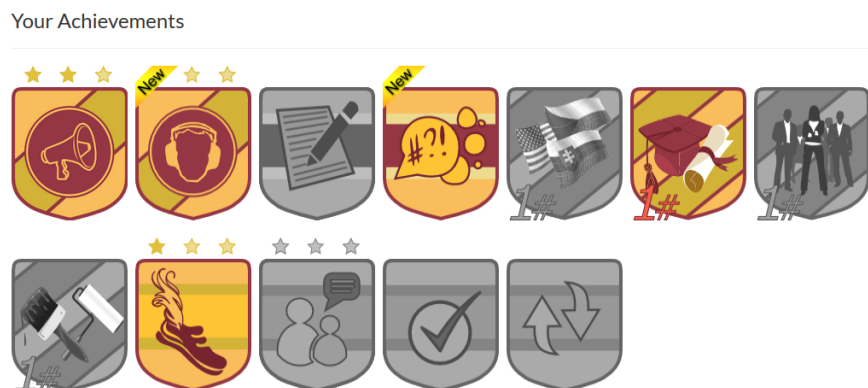
Pri návrhu sme sa opierali o pohľad potenciálneho používateľa a už existujúce riešenie. Aktuálny design postavený na frameworku Bootstrap sa dá charakterizovať ako flat. TO znamená, že elementy majú minimálne množstvo detailov a temer žiadne farebné prechody, tvoriace dojem priestorovosti. Flat - plochý. Z tohto predpokladu sme vychádzali. Základná štruktúra aždého achievementu musela zodpovedať nasledovnému návrhu.



Obr. 87: Návrh nových ocenení

Implementácia a testovanie

Podľa návrhu sme vytvorili vlastné achievementy ktoré spĺňali predom špecifickú farebnú škálu. Aktívny achievement je farebný, neaktívny čiernobiely. Hviezdičky sú postupne vyplňané podľa množstva získaných achievementov. Výsledná implementácia je znázornená na nasledujúcom obrázku.



Obr. 88: Výsledná implementácia ocenení

3.17.3 Vytvorenie statických stránok

Autor: Bc. Marek Lenčes

Špecifikácia požiadaviek

Základné informácie pre používateľa, ako sú sručná charakteristika aplikácie, na čo slúži, aký je jej cieľ a hodnota pre používateľa, návod na použitie aplikácie, kontaktný formulár a zoznam noviniek je základnou súčasťou každej webovej aplikácie, umožňuje používateľom lepšie sa na stránke orientovať a dozvedieť sa všetky potrebné informácie. Tieto údaje budú dostupné používateľom vo forme statických stránok.

Návrh

Statické stránky budú obsahovať jednoduché informácie, stránka O nás obsahuje charakteristiku aplikácie, jej účel a opis funkcií. Stránka Kontakt obsahuje kontaktný formulár pre odoslanie dodatočných otázok alebo postrehov. Stránka novinky obsahuje informácie o novej pridanej funkcionalite v aplikácii, zmenách, alebo opravených chybách. Stránka Pomocník má za účel poskytnúť používateľom návod na použitie aplikácie.

Implementácia a testovanie

About Pinta.sk

PinTa.sk
an online **feedback** marketplace

ČO JE Pinta.sk ?
Pinta.sk je moderné webové rozhranie na sprostredkovanie mikro-práce. Nazov je odvodený z anglického "pin task", príjmi úlohu. Pinta.sk sa od ostatných konkurenčných aplikácií líši moderným a prepracovaným používateľským rozhraním, podporou veľkého množstva typov úloh a zábavným a pútavým spracovaním, ktoré Vás núti "pracovať" viac a viac. Pre pracantov máme pripravené :

- prepracované a príjemné používateľské rozhranie,
- kreditný systém a možnosti výmenného obchodu,
- systém evolúcie (vývoja používateľa),
- karma systém (oblíbenosť úloh, zadávateľov aj pracantov),
- sociálnu integráciu a možnosť súperit s priateľmi.

Zadávateľa sa môžu tešiť na :

- množstvo nažhavených pracantov,
- rozsiahle možnosti definovania úloh s vysokou mierou automatizácie.

©2014 Pinta.sk, All rights reserved.

Obr. 89: Statická stránka O nás

Send A message to Us

Name

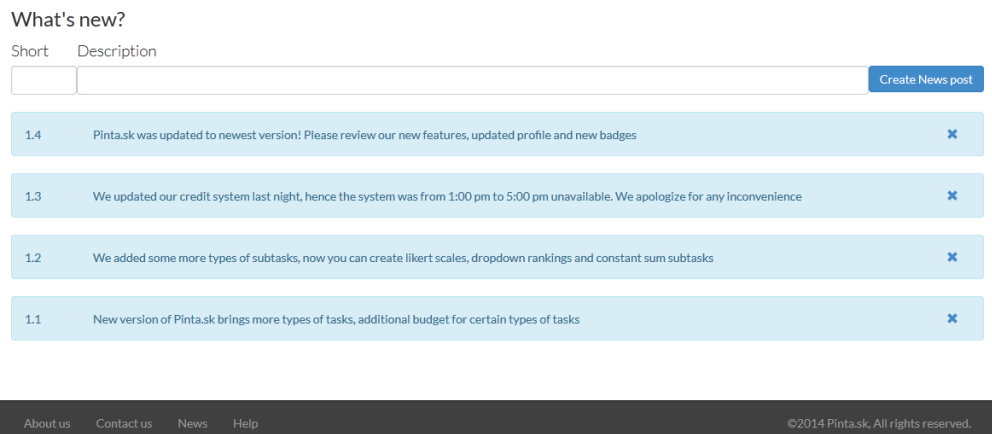
E-mail

Your message..

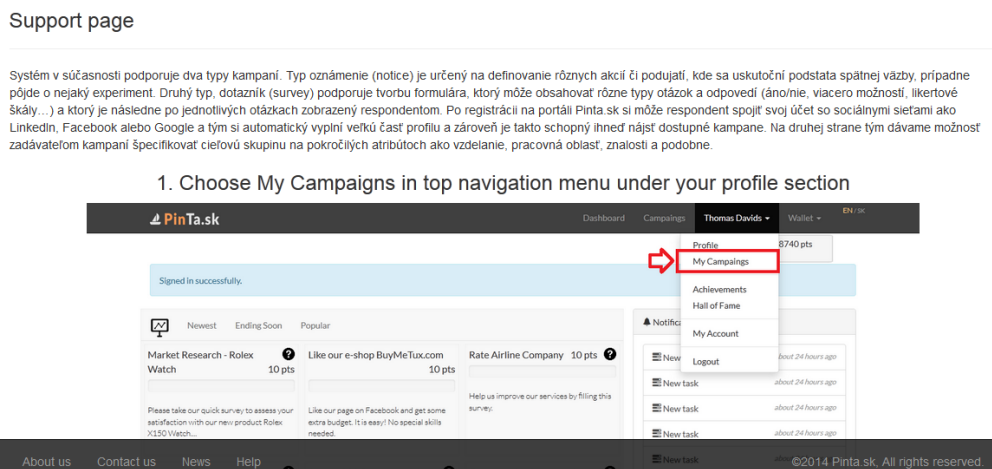
Send message

©2014 Pinta.sk, All rights reserved.

Obr. 90: Statická stránka Kontakt



Obr. 91: Statická stránka Novinky



Obr. 92: Statická stránka Pomocník

3.17.4 Doplnenie transakčnej logiky do notice kampane

Autor: Bc. Róbert Černý

Pre úspešne dokončenie kampane typu notice bolo potrebné implementovať systém na spracovanie transakcií. Po tomto úkone bolo možné dovoliť zadávateľovi potvrdiť výsledok pracovníka a vykonať tak prenos kreditu medzi subjektami. Implementovanie úlohy boli jednoduché nakoľko bol systém pre transakcie navrhnutý korektne. Po akceptovaní výsledku zadávateľom prebehne transakcia, ktorá prevedie kredit.

3.17.5 Automatické počítanie ceny kampane

Autor: Bc. Róbert Černý

Analýza

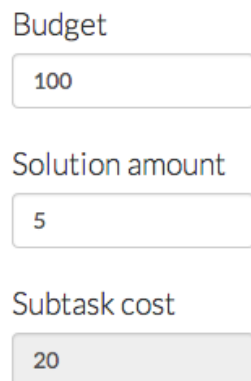
V procese tvorby novej kampane bude jednoduchšie umožniť zadávateľovi zadať celkový kredit ktorý je ochotný investovať a počet výsledkov, ktorý pre jeho potreby stačí. Aplikácia potom dopočíta, množstvo kreditu, ktoré sa prevedie pracovníkovi za úspešné vyriešenie celej kampane.

Návrh

premenovať atribút `subtask_task` na `solution_count` pomocou JS dopočítavať hodnotu počtu výsledkov pomocou zadaného kreditu a počtu otázok kampane.

Implementácia a testovanie

Na ukážke sa v okne `Subtask cost` dopočíta počet možných riešení kampane.



The image shows a form with three input fields. The first field is labeled 'Budget' and contains the value '100'. The second field is labeled 'Solution amount' and contains the value '5'. The third field is labeled 'Subtask cost' and contains the value '20'. The fields are arranged vertically and have a light gray border.

Obr. 93: Dopočítanie ceny jednej podúlohy

3.17.6 Vytvorenie prezentačného videa

Autor: Bc. Filip Bednárík, Bc. Róbert Černý

Analýza

Pre úvodné oboznámenie sa s aplikáciou potrebujeme prezentovať projekt používateľom bez toho, aby sa museli registrovať. Umožníme im tak zoznámenie sa s hlavnými funkciami systému. Vďaka návodu ako pracovať so systémom lepšie pochopia princíp a účel aplikácie. Najjednoduchším štýlom prezentácie je startupové video.

Návrh

Navrhli sme úvodnú scénu v 3D modelovacom programe Blender, kde na výkonnom počítači sa táto scéna renderovala 5 hodín, čo značí o kvalite tohoto intra. Výber hudby bol veľmi dôležitý, vo videu by mala byť ukludňujúca hudba ale s dobrou a chytlavou melódiou so stupňujúcim rázom a zároveň kvôli nízkemu rozpočtu projektu bolo vhodné vybrať hudbu bez poplatku za licenciu. Vybral som najdôležitejšie funkcie systému a tie som heslovito opísal a ukázal ako sa na nich orientovať a ako pracovať so systémom

Implementácia

Video sme vytvorili v prostredí modelovacieho programu Blender a zostrihali pomocou nástrojov DVDVideoSoft a Windows MovieMaker.

3.17.7 Nové funkcie do dashboardu

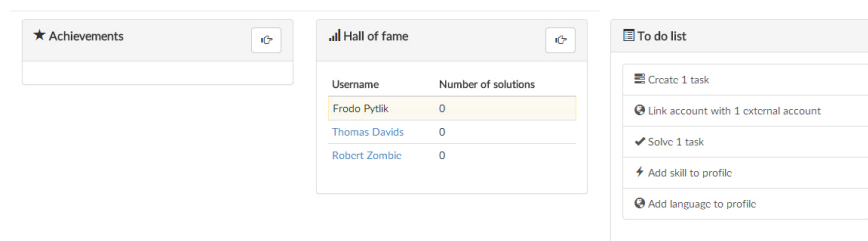
Autor: Bc. Filip Bednárík

Analýza

Naším cieľom je čo najviac motivovať používateľa aby riešil úlohy a preto chceme zobrazovať prehľad na jednej stránke. Vďaka tomu aj nový používateľ vie čo má robiť a nestratí sa v systéme.

Návrh a implementácia

Používateľovi zobrazíme +-2 používateľov z Hall of fame. ďalej úlohy pre splnenie. Pre implementáciu sme využili achievements, zoradené od najjednoduchšieho po najťažší. Z databázy sme vybrali používateľov s podobným počtom riešení ako má aktuálny používateľ. Výsledok vidíme na Obrázku 94.



Obr. 94: Nové funkcie na hlavnom dashboarde

3.17.8 User management + transakcie

Autor: Bc. Filip Bednárík

Analýza

V systéme potrebujeme spravovať používateľov a poskytovať možnosť zobrazenia zoznamu všetkých používateľov a tiež možnosť pridávať používateľom body manuálne.

Návrh a implementácia

Pridali sme nový typ transakcie typu *bankToUser*, čiže pridanie kreditov z „b“anky používateľovi a tlačidlo pre pridanie bodov do rozhrania zoznamu používateľov. Výsledok vidíme na Obrázku 95.

Listing Users

Email	Budget	Firstname	Lastname	Country	
admin@example.com	4688	Frodo	Pytlík	Canada	Add 100 pts Show
test@example.com	8740	Thomas	Davids	Cuba	Add 100 pts Show
rob.zombic@gmail.com	380	Robert	Zombic	Finland	Add 100 pts Show

Obr. 95: Manuálne pridanie kreditov

3.17.9 Filtre

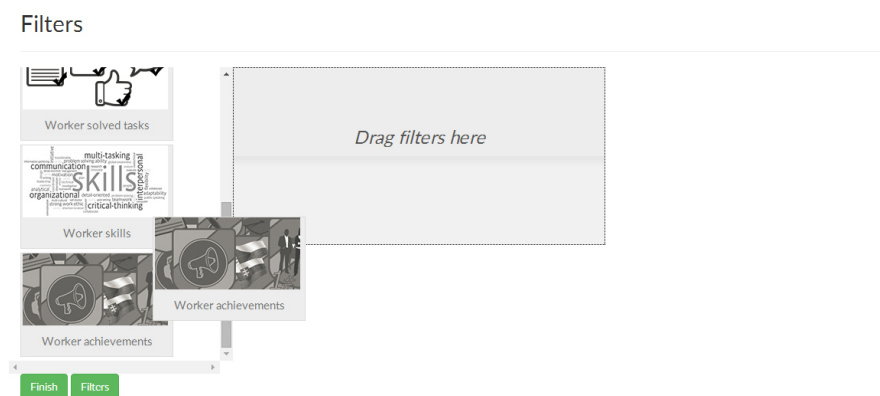
Autor: Bc. Filip Bednárík

Analýza

Zistili sme, že v procese vytvárania úlohy poskytujeme nekonzistentné filtre, ktoré boli používateľsky neprívetivé a zle reprezentované. Preto sme sa rozhodli vylepšiť pôvodný návrh grafickým zobrazením filtrov a pridávaním pomocou drag&drop.

Návrh a implementácia

Grafický návrh filtrov vidíme na Obrázku 96.



Obr. 96: Prerobené filtre

3.17.10 Welcome tasky pre IIT.SRC

Autori: Bc. Filip Bednárík, Bc. Martin Toma

Analýza

Pre potreby prezentácie na študentskej vedeckej konferencii IIT.SRC 2014 sme potrebovali dať návštevníkom možnosť získať kredity v našom systéme a následne byť schopný vytvoriť vlastné kampane. Rovnako sme chceli získať spätnú väzbu na niektoré elementy a procesy v našom systéme.

Návrh

Navrhli sme preto vytvoriť kampane pod menom Pinta.sk, ktorých zmyslom bude ohodnotiť proces registrácie, orientáciu na webe, či pochopiteľnosť nadpisov a akcií. Ako odmena za vyplnenie úloh boli poskytnuté kredity v našom systéme. Takto sme mi získali spätnú väzbu a zároveň ten kto nám ju poskytol získal možnosť vytvoriť vlastnú kampaň.

Implementácia a testovanie

Tieto kampane sme implementovali v našom systéme pomocou nami implementovaných mechanizmov. Vytvorili sme jednu single choice, jednu multi choice a zopár likert scale kampaní na úkážku možností v našom systéme. Na obrázku nižšie je vidieť zoznam kampaní, dostupných novému používateľovi po zaregistrovaní sa na našom portáli. Dôležité je, že používateľ postupne rieši kampane a filtre v našom systéme sme nastavili tak, aby na podstatu danej otázky vedel používateľ

Name	Description	Task Category	Points for solving	Deadline	Action
What do you think about campaign creation process?	Simple set of tasks about campaign creation process. Solve this task after you created one campaign.	Survey	10	2014-06-30 09:40:00 UTC	Not eligible
What are your first impressions of Pintask?	Simple set of questions about your initial user experience with our site. Solve this task to get points, so you can create your own tasks.	Survey	10	2014-06-30 09:40:00 UTC	Solve
What do you think about task solving process?	Simple set of questions about campaign solving process. Solve this task after you solved other task to get some more points.	Survey	10	2014-06-30 09:40:00 UTC	Not eligible
What do you think about your profile on pintask?	Simple set of questions about filling your profile and your experience with our system. Solve this task after you filled at least one skill and one language in your profile.	Survey	10	2014-06-30 09:40:00 UTC	Not eligible

Obr. 97: Zoznam vytvorených kampaní

odpovedať, musí teda napríklad najprv vytvoriť riešenie aby mohol tento proces ohodnotiť. Týmto sme chceli ukázať silu filtrov implementovaných v systéme.

3.17.11 Úprava aplikácie a oprava chýb a nasadenie do produkcie

Autor: Bc. Patrik Štrba V rámci tohto šprintu bolo potrebné vykonať úpravy a opraviť vzniknuté chyby aby sme mohli aplikáciu prezentovať na vedeckej konferencii.

Nasledujúce úpravy bolo potrebné vykonať:

- Pridanie popisných vyskakovacích okien
- Pridanie metódy, ktorá bude kontrolovať či zvolený používateľ vytvoril úlohu
- Automatické prihlasovanie pri registrácii
- Úprava filtrácie

Pridanie popisných vyskakovacích okien V rámci procesu vytvárania alebo úpravy úlohy boli pridané vyskakovacie okná pri označení myšou na otázniky. Toto bolo realizované cez knižnicu tipsy.

Pridanie metódy, ktorá bude kontrolovať či zvolený používateľ vytvoril úlohu Pridanie metódy isCreator do modelu úlohy a použiť na zvolených miestach (refactoring).

Automatické prihlasovanie pri registrácii Pri úspešnej registrácii bolo potrebné upraviť potrebné obsluhujúce skripty aby bol používateľ prihlásený.

Úprava filtrácie Preddefinované filtre nefungovali správne. Pri úpravách vznikli chyby, kedy zvolené filtre nefungovali alebo fungovali iným spôsobom akým by mali. Toto bolo opravené úpravami v modeli úlohy, kde bola opravená metóda, ktorá kontrolovala či daný používateľ má prístup k danej úlohe alebo nie.

Nakoniec bolo potrebné opakovane nasadenie do produkcie a opraviť všetky vzniknuté chyby.

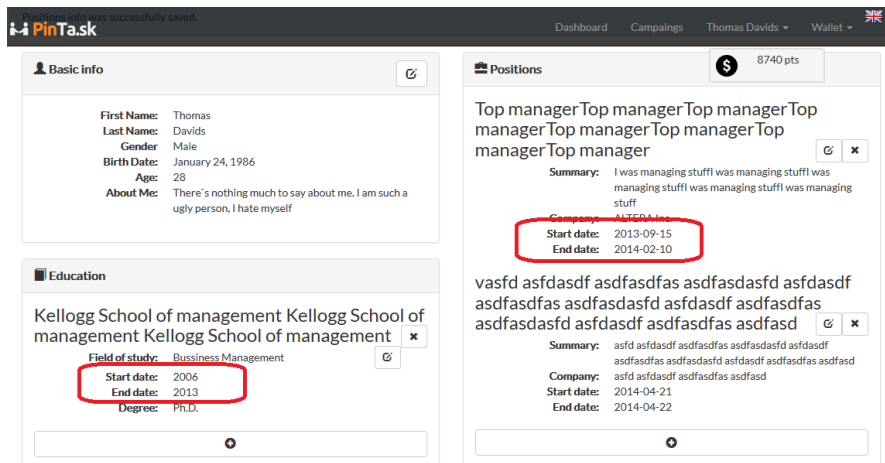
Hľadanie chýb v systéme a ich odstraňovanie

Analýza

Systém pre nasadenie na produkčný server a následnú prezentáciu cieľovým zákazníkom musí byť dobre funkčný, preto je nutné aby sa celý systém pretestoval a boli odhalené všetky chyby. Následne po odhalení chýb je nutné dobre chyby v systéme zdokumentovať.

Návrh

Nájdene chyby boli zapísané do JIRA a bola im určená priorita s akou sa majú vypracovávať. K popisu bol vytvorený screenshot napomahajúci chybu pochopiť. V prípade potreby k screanshote bolo problémové miesto aj podrobnejšie zvýraznené, viac obrázok 98. Pred pridelením však bolo potrebné chybu preveriť či naozaj ide



Obr. 98: Screenshot k problému

o chybu, alebo zlé testovacie prostredie. Po overení chyby bolo nutné opätovné prehodnotenie priority. Takto zdokumentované a preverená chyby sa následne dali do zásobníku, z ktorého sa podľa priority prideliť členovi tímu podľa skúsenosti s danou časťou kódu, kde sa chyba vyskytovala.

V systéme boli odhalené viaceré typy chýb s rôznou úrovňou. Najčastejšie chyby s nízkou úrovňou sa týkali zlého prekladu textu, chýbajúcich obrázkov v html kóde po načítaní webovým prehliadačom, zle zobrazenie vyplnených dát v profile, zle zobrazené výsledky v tasku. Medzi vážnejšie chyby patrili nefunkčné verzie gemov, pády systémy pri prvom spustení. Niektoré chyby boli spôsobené neposlaním všetkých zmien na repozitár. Tak vznikali chyby, ktoré v skutočnosti sa už odstránili. Testovanie prebiehalo na viacerých prehliadačoch.

3.17.12 Povolenie správnych hodnôt v Education a Positions

Autor: Bc. Miroslav Molnár

Analýza

Boli detegované problémy so správnym načítaním dát z LinkedIn a nesprávna funkcionálna začiatková a ukončovacia časť v systéme týkajúceho sa hodnôt v Education a Positions. Pre úpravu týchto hodnôt je treba byť konzistentný s LinkedIn odkiaľ berieme údaje o používateľoch systému. Je potrebné aby všetky dáta z LinkedIn boli úspešne uložené do nášho dátového modelu. To znamená, že naše obmedzenia dátového modelu musia byť rovnaké alebo miernejšie ako na LinkedIn.

Návrh

Našli sa ďalšie chyby v obmedzeniach dĺžok záznamov, ktoré LinkedIn sociálna sieť nemá v sebe implementované do takej miery ako sme ich mali implementované my

v svojom systéme. Problém nastal v tom, že korektne vyplnený profil nebol korektný pre náš dátový model a teda dáta z profilu LinkedIn neboli uložené v našej databázy. Kôli tomuto sa museli dôkladne naštudovať a otestovať minimálne a maximálne hodnoty aké môže používateľ nastaviť v LinkedIn profile pri Education a Positions. Ďalším problémom je, že náš systém bol schopný uložiť dáta v Education a Positions pri hodnotách kedy používateľ zadal začiatok neskoršie ako je aktuálny stav. Rovnako pri Positions nebolo umožnené aby používateľ zadal hodnotu ktorá ešte trvá dodnes.

Implementácia a testovanie

Na začiatku boli implementované minimálne dĺžky zadaných znakov z 5 na 1 ako to má LinkedIn, následne bolo nutné dať aj obmedzenia na hodnoty začiatočného a koncového dátumu. Pre Vzdelanie sú odlišné. Používateľ si nemôže dať do profilu školu, ktorú ešte neskončil nakoľko ak ju neukončí nemá žiadne osvedčenie, že nadobudol dané vedomosti a zručnosti. Pri Positions to je možné. Ak používateľ zvolí, že na danej pozícii je až do teraz tak sa mu pozícia uloží s príznakom, že ešte stále trvá. Používateľovi sa následne zobrazí v mieste dátumu ukončenia hláška I currently work here. Bolo nutné aj obmedzenie začiatočného času ktorý nemohol byť skoršie zadaný ako je čas aktuálny. To vymedzilo možnosti používateľom zadávať Education alebo Position, ktorú ešte len plánujú.

3.18 Závěrečný šprint

Autori: Všetci členovia

Po prezentácii na súťaži TP-CUP sme dostali mnoho nových nápadov a postrehov k našej aplikácii. Tie nám pomohli zistiť, čo nám všetko ešte chýba a ktoré časti systému by sme potrebovali vylepšiť. Následne sme sa v poslednom šprinte venovali dotváraniu výslednej dokumentácie a ostatných prezentačných materiálov našej aplikácie a opis celkového pohľadu.

4 Celkový pohľad

4.1 Opis prototypu

Súčasný stav aplikácie pracuje v testovacom režime. Beta verzia aplikácie sa nachádza na www.pinta.sk a je prístupná verejnosti. Webová aplikácia má plne upravené grafické rozhranie, ktoré je prispôsobené téme lode, ktorá vyplýva z názvu projektu pint. Po príchode na stránku je používateľ informovaný o účele webovej aplikácii. Aplikácia umožňuje prihlásiť sa pod testovacími heslami vytvorených, ako pre zadávateľa, tak pre pracovníka. Aplikácia pracuje v dvoch základných módoch:

- mód určený pre zadávateľa,
- mód určený pre pracovníka.

Funkcionalita módov sa líši. V móde zadávateľ používateľ môže vytvoriť novú úlohu pre pracovníka. V aplikácii sa môžu aktuálne vytvoriť nasledovné úlohy:

detekcia informácií z obrázka - používateľ môže nahrať konkrétny obrázok, alebo dokáže nahrať archivačný súbor, ktorý je aplikácia schopná rozbaľiť a vybrať z neho obrázky.

Dynamické vytváranie testovacích formulárov: používateľ je schopný vytvoriť otázku a k nej odpovede. Takýmto spôsobom sa dajú vytvoriť celé testovacie zadania. Pracovník s testovacím formulárom pracuje tak, že poznačuje správne odpovede k zadaným otázkam.

Pridanie like na facebook- používateľ vloží url na facebook, pomocou ktorej pracovník sa dostane na stránku a potvrdí like.

Po vybratí typu úlohy aplikácia umožňuje zadať aj obmedzujúce podmienky pre pracovníka. Vytvorenú úlohu môžu vykonať iba pracovníci, ktorí tieto podmienky spĺňajú. Zadávatel ďalej môže určiť a odmenu pracovníka za vypracovanie úlohu. Pracovník je takto motivovaný k lepšej a efektívnejšej práci v našej aplikácii. Aplikácia umožňuje zadávateľovi taktiež pohľad na aktuálny stav úlohy. Výsledky úlohy dokáže reprezentovať pomocou grafov.

V móde pracovník, používateľ nemá možnosť vytvárať úlohy, ale rovnako ako zadávateľ môže si úlohy prezerat. Pracovník po začatí vypracovávania úlohy môže úlohu prerušiť. Pracovníkovi sa zobrazujú iba tie úlohy, pri ktorých spĺňa podmienky zadávateľa.

Okrem odmen za vypracovanie úlohy sa nachádza v aplikácii aj odmeňovací systém badges, ktorý oceňuje pracovníkov za špeciálne procesy splnené v systéme. Je to ďalší proces motivácie používateľov a je ním sledovaná väčšia chuť pracovať a pracovať kvalitne na svojich úlohách.

Aplikácia umožňuje používateľom pripájať sa na sociálne siete Twitter, Facebook, LinkedIn, Google. Prepojenie so sociálnymi sieťami aplikácia využíva na doplnenie profilu. Profil si používateľ vie sám upravovať a meniť podľa potreby. Je možné vyplniť krajinu, región, meno, priezvisko, prezývku, email, dátum narodenia, z ktorého sa generuje vek používateľa, jazykové zručnosti, dosiahnuté vzdelanie. Plne vyplnený profil je pre pracovníka výhodou, lebo zadávateľ môže zadať obmedzujúce podmienky na hodnoty z profilu.

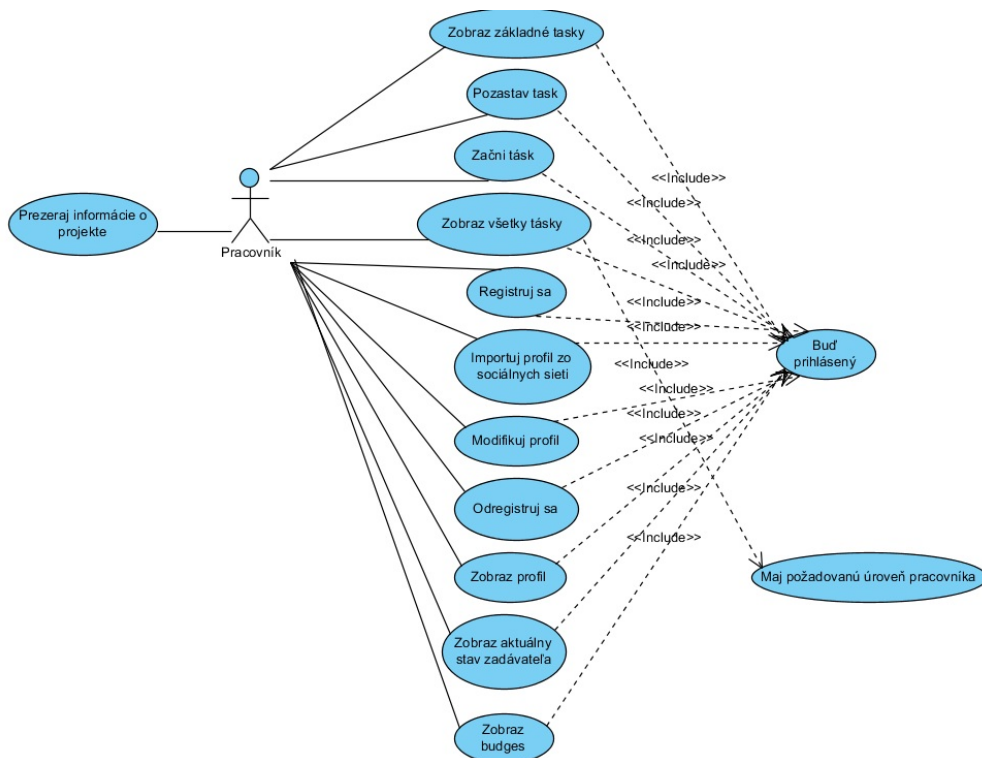
4.2 Funkcionálne požiadavky

Z funkcionálneho hľadiska sa nám podarilo naimplementovať webovú aplikáciu umožňujúcu registrovať a následne sa prihlásiť používateľom. Aplikácia umožňuje

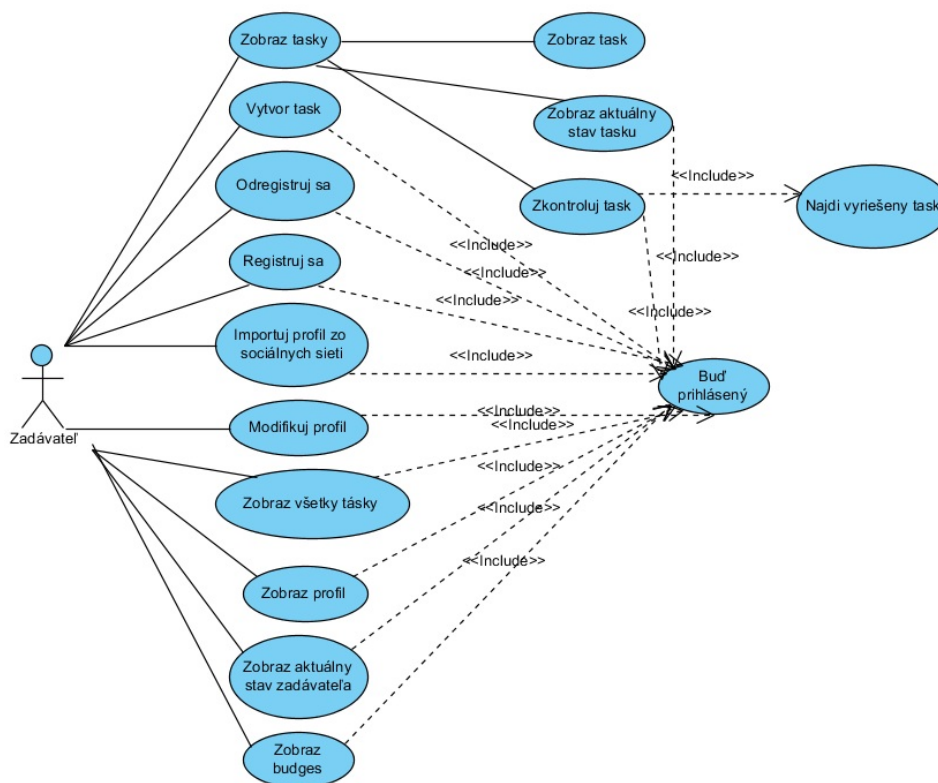
vytvárať rôzne typy úloh s rôznymi nastaveniami nárokov na vypracovanie, ale aj na pracovníka. Je možné taktiež zobrazovať grafickú podobu výsledkov. Na strane pracovníka je možné úlohy prerušovať a ľubovoľne meniť poradie ich vypracovania. Jednou z významných funkcionalít je práca a profilom, ktorý si môže používateľ vyplňať a upravovať. Do profilu je možné načítavať informácie z profilov na sociálnych sieťach po integrácii s účtom. V profile je možné nastavovať najrôznejšie veci od mena, priezviska až po vzdelanie a pracovnú kvalifikáciu.

4.3 Prípady použitia

V systéme máme dve hlavné roly a to zadávateľ a pracovník. Pracovník vypracováva úlohy zadané zadávateľom. Je mu umožnené prezerat si zadania základných úloh. Po registrácii sa môže prihlásiť do systému. Po prihlásení do systému pracovník môže úlohy, ktoré sa mu zobrazia aj vypracovať. Taktiež si môže prezerat a dopĺňať profil a tým zvyšovať svoje šance na zobrazenie lepších úloh. Úlohy môžu byť zadané s rôznymi požiadavkami. Jedným z tipov požiadaviek môžu byť špecifické hodnoty z profilu. Pracovník môže taktiež úlohu prerušiť. Je možné si prezerat aj zatiaľ dosiahnuté výsledky ako môžeme vidieť na Obrázku 99



Obr. 99: Zobrazenie prípadu použitia pracovníka

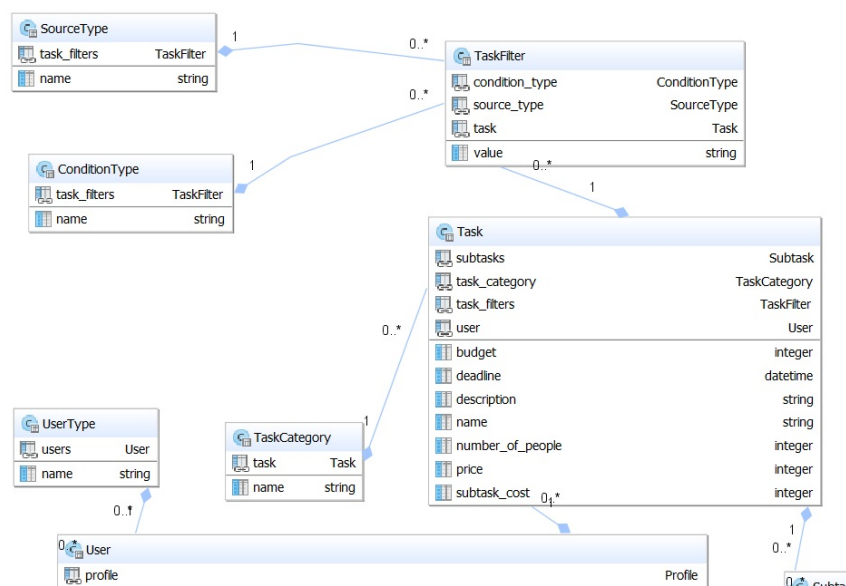


Obr. 100: Zobrazenie prípadu použitia zadávateľa

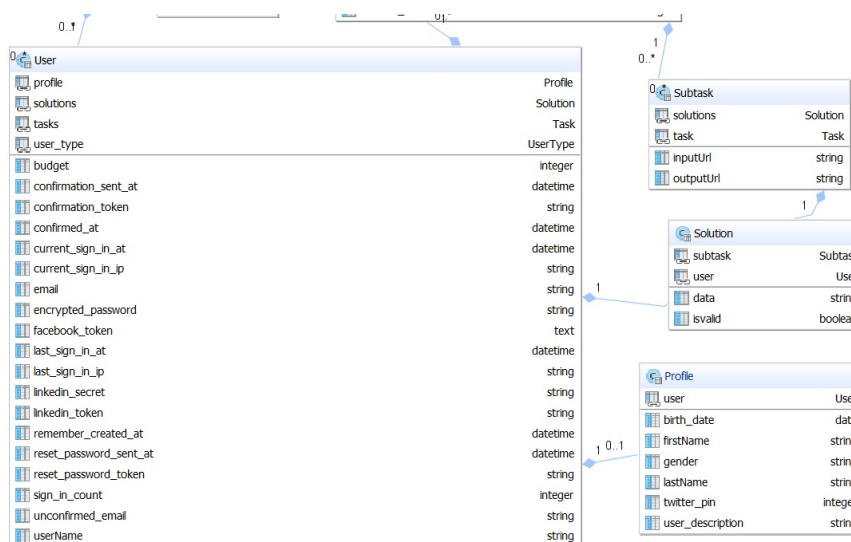
Zadávateľovi, a pracovníkovi sú pred prihlásením zobrazené tie isté informácie a majú možnosť pracovať s rovnakou funkcionalitou v aplikácii. Zmena nastáva až po prihlásení, kedy Zadávateľ má iné možnosti práce z aplikáciou a je schopný vytvárať úlohy. Týmto úlohám môže priradiť rôzne kritéria na vlastnosti pracovníka. Následne sa úlohy zobrazia iba pracovníkovi, ktorý tieto kritéria spĺňajú. Ako môžeme vidieť zobrazené na prípade použitia obr 2. .Zadávateľ si taktiež môže vytvorené úlohy prezerat .

4.4 Dátový model

Dôležitými dátovými štruktúrami sú Task, Subtask a Solution. Tieto dátové štruktúry uchovávajú všetky úlohy vytvorené zadávateľmi v systéme a ich stavy. Sú dôležité pre zadávateľa, ale aj pracovníka. Oba však majú rôzny pohľad na informácie v týchto dátových štruktúrach. Hlavnou dátovou štruktúrou v projekte je User. User obsahuje informácie o používateľovi. Používateľom je zadávateľ, ale aj pracovník. Dátová štruktúra slúži na ukladanie informácií pre prihlásenie, registráciu a iných stavov používateľa v systéme. Tabuľka je prepojená s ďalšími dátovými štruktúrami, ktoré slúžia na ukladanie informácií o používateľoch.

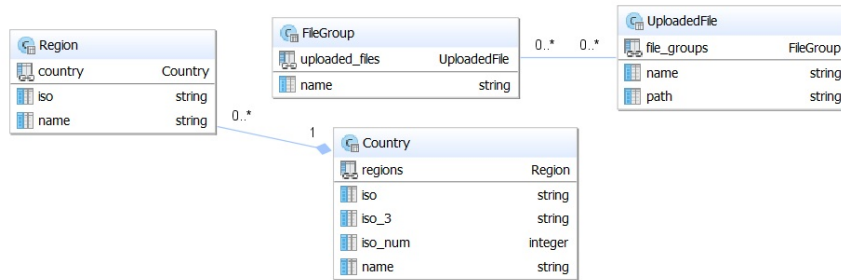


Obr. 101: Dátový model zobrazujúci štruktúru Task, Subtask a Solution.



Obr. 102: Dátový model zobrazujúci štruktúru User.

Jednou z takýchto dátových štruktúr je Profile, ktorý v sebe uchováva doplňujúce informácie dôležité pre uploadera. Profile je neoddeliteľnou súčasťou User. Po vytvorení User dátovej štruktúry sa automaticky vytvorí aj Profile dátová štruktúra. Pri zmazení User sa zmaže aj Profile. Opačne to neplatí. Pre uloženie regiónu a krajiny slúžia dátové štruktúry Region a Country.



Obr. 103: Datový model zobrazujúci datovú štruktúru Country, Region, Uploaded-File.

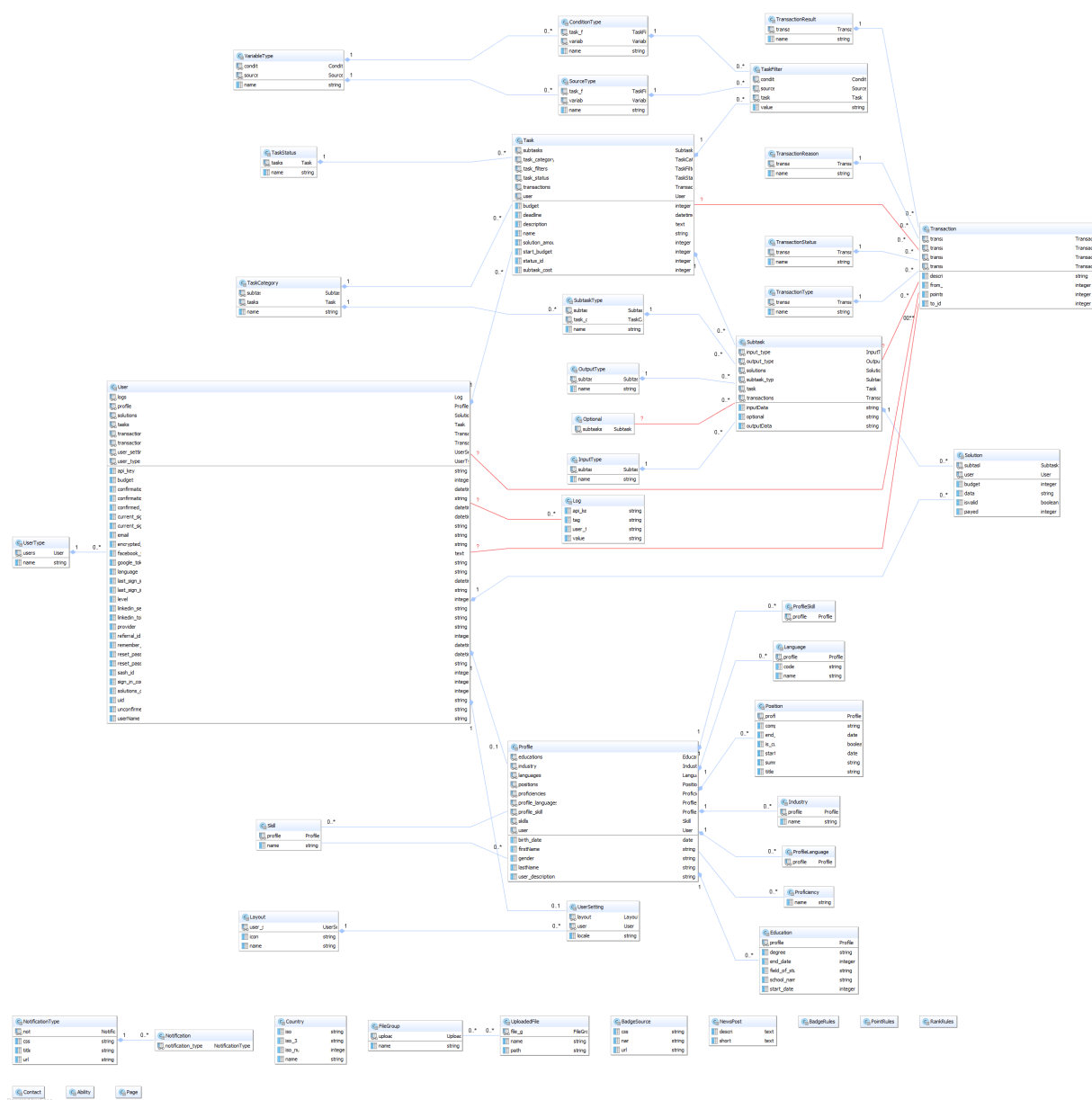
4.5 Letný semester

Projekt v aktuálnom stave ponúka viacero možností na ďalší rozvoj. Je dokončený dostatok funkcionality na pokračovanie beta testovaním a nasadenie do kontrolovanej oblasti a využitie zo začiatku v neziskových sektoroch. Projekt nebol doteraz smerovaný na konkrétny cieľ a preto sme ho vyvíjali univerzálne, preto je teda možné projekt upraviť pre konkrétnejšie potreby cieľových zákazníkov. V rámci projektu sme mali stretnutie s osobami v oblasti marketingu a startupov. Každý z týchto osôb dostal informácie o našom projekte a ponúkol nám smerovanie, prípadne čo by sme mali zlepšiť. Na základe týchto podnetov sme sa potom ďalej rozhodovali.

Najúžitečnejšiu spätnú väzbu sme získali od Kataríny Chomovej, ktorá je výskumná pracovníčka na Ekonomickej Univerzite v Bratislave. Tá nám poradila zamerať sa na cieľovú oblasť študentov. Tí by nám mohli zo začiatku slúžiť ako testujúca skupina nášho produktu s tým, že ich motivácia by mohla byť zabezpečená napríklad aj vecnými odmenami.

Ďalšie aktuálne možné zameranie je pomoc pri získavaní spätnej väzby pri bakalárskych, diplomových či dizertačných prác. Takto študenti v rámci svojich prác často potrebujú získavať spätnú väzbu od relevantného publika. Naš nástroj by bol pre takúto činnosť výborný vzhľadom na to, že u nás v systéme sa dajú výsledky filtrovať a získavať len relevantné dáta. Táto cesta má svoje výhody, ale aj nevýhody. Po úspešnej testovacej fáze, by mohlo prísť k snahe dostať produkt pred prvých platiacich zákazníkov. Ako druhú stranu nám preto poradila sa zacieliť na malé a stredné firmy. Následne by sme naštartovali marketing, ktorý by bol cielený pre študentov v lokálnom regióne a neskôr aj v rámci celého Slovenska. Taktiež by bolo potrebné získať nových zadávateľov úloh, kde by sme sa zamerali na malé a stredné firmy v rámci celého Slovenska.

4.5.1 Celkový dátový model



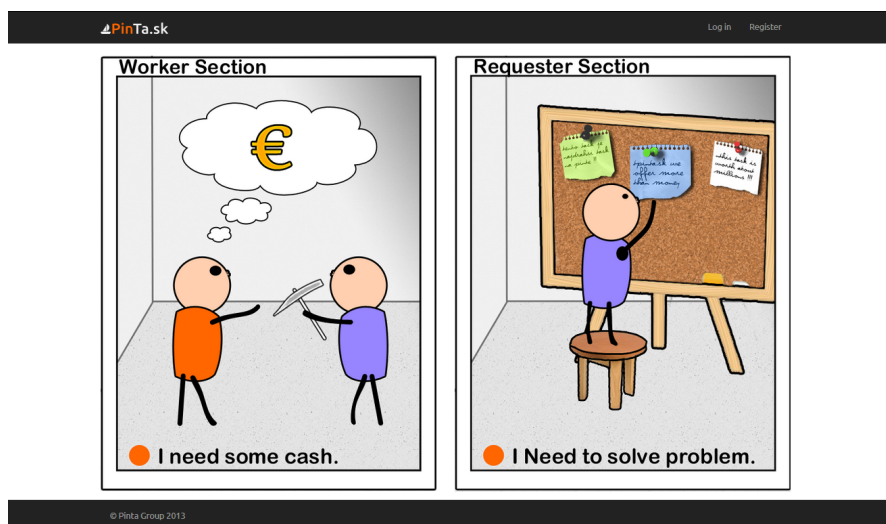
Obr. 104: Celkový dátový model aplikácie

5 Používateľská príručka

Úvodná stránka používateľovi ponúka na výber dve sekcie. Vľavo je sekcia pre “zadávateľa” a vpravo pre “pracovníka”. Každá rola má čiastočne upravené dostupné funkcie.

Role:

- Zadávateľ - zádáva úlohy, ktoré riešia pracovníci
- Pracovník - má možnosť vypracovať úlohy zadane zadávateľom



Obr. 105: .

Hlavné menu pre prihlásením ponúka možnosti:

- Log in/Sign in (prihlásenie)
- Register/(registrácia)

5.1 Prihlásenie do systému

Na úvodnej stránke si zvolte ponuku pre prihlásenie (Log in). Pre prihlásenie je potrebný email a heslo používateľa, s ktorými sa registroval. Nezabudnite pred prihlásením potvrdiť aktiváciu prostredníctvom odkazu, ktorý vám prišiel do emailu. V prípade že vám neprišiel, využite možnosť opätovného zaslania (odkaz “Didn’t received confirmation instructions?”). V prípade, že ste zabudli prihlasovacie heslo. Môžete využiť možnosť obnovy hesla (odkaz “Forgot your password?”).

Sign In

Email
worker@example.com

Password
••••••••

Remember me

Sign in

[Sign up](#)
[Forgot your password?](#)
[Didn't receive confirmation instructions?](#)

Obr. 106: .

5.2 Registrácia do systému

Na úvodnej stránke si zvolte ponuku pre registráciu (Register)
Pre úspešnú registráciu je nutné vyplniť všetky polia vrátane krajiny a regiónu.

- V poli email musí byť vyplnená platná emailová adresa.
- heslá sa musia zhodovať
- krajina aj region musí byť vyplnený - systém automaticky tieto údaje odhadne z vašej IP adresy, tieto údaje sú použité na ciele úlohy pre určité lokality.

Sign Up

Email
worker@example.com

Password
••••••••

Password confirmation
Your password again

Country
Slovakia

Region

Sign up

[Sign in](#)
[Didn't receive confirmation instructions?](#)

Obr. 107: .

5.3 Opätovné zaslanie aktivačného emailu

V prípade, že vám neprišiel email s aktivačnými inštrukciami. Môžete požiadať o znovuposlanie cez formulár prístupný cez pohľad prihlásenie a odkaz “Didn't receive confirmation instructions?”.

Resend confirmation instructions

Email

Resend confirmation instructions

[Sign in](#)

[Sign up](#)

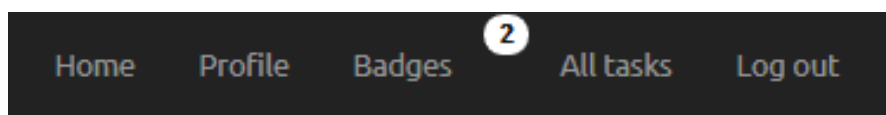
[Forgot your password?](#)

Obr. 108: .

5.4 Sekcia pre pracovníka

Hlavné menu ponúka nasledovné možnosti:

- Home - domovská stránka so základnými informáciami o používaní systému a inými detailmi podľa aktuálnej ponuky (v aktuálnej verzii novinky a základne informácie, štatistiky vypracovaných úloh)
- Profile - profil používateľa. Informácie o používateľovi, ktoré je možné prepojiť so sociálnymi službami a tým získať ďalšie výhody. Tieto informácie slúžia na cielené zadávanie úloh.
- Badges - achievement-y alebo úspechy, ktoré používateľ dosiahol/môže dosiahnúť , číslo pri odkaze značí počet nových achievement-ov
- All Tasks - zoznam riešiteľných úloh



Obr. 109: .

5.4.1 Profil

Pohľad s informáciami o používateľovi. Všetky údaje môžu byť upravené pomocou príslušného ovládacieho prvku:

- zmena/upravenie záznamu
- pridanie záznamu
- odstránenie záznamu

Tlačidlá “Connect via xxxx” slúžia na prepojenie účtu s príslušnou sociálnou službou. Po prepojení sa tlačidlo zmení na “Update profile with xxxx”, po kliknutí obnoví informácie v profile pomocou dát z príslušnej služby.

Basic info

First Name: First name
Last Name: Last name
Gender: Male
Birth Date: January 01, 1900
Age: 113
About Me: My description

Positions

Programmer

Summary: a lot of hard work to do
Company: Company 1
Start date: 2013-01-12
End date:

Čašník

Summary: tiež celkom ťažká práca
Company: Nálepko
Start date: 2013-01-12
End date:

Education

STU BA FIIT

Field of study: infommatcs
Start date: 2013
End date: 2013
Degree: Mgr

Languages

Update profile with LinkedIn Connect via Facebook Connect via LinkedIn Connect via Google

Obr. 110: .

5.4.2 Badges

Zoznam dostupných achievemento-ov (úspechov) dosiahnutých používaním systému. V budúcnosti budú použité na cieľené zadávanie úloh.

Listing your badges

id	name	title	description	requirements	granted?
1	apprentice-requester		Apprentice student of job requesting	1	F
10	apprentice-worker		Apprentice student of job solving	1	F
11	journeyman-worker		Journey man in job solving	10	F
12	expert-worker		Expert in job solving	100	F
13	artisan-worker		Artisan in job solving	500	F
14	master-worker		Master in job solving	1000	F
15	grand-master-worker		Grand master in job solving	5000	F
16	illustrious-worker		Illustrious in job solving	10000	F
20	first-language		First language defined.	1	F
21	first-education		First education defined.	1	T
22	first-position		First position defined.	1	T
23	first-skill		First skill defined.	1	F

Obr. 111: .

5.4.3 All tasks

Pohľad pre zobrazenie úloh dostupných na riešenie.

Každá riešiteľná úloha má v slípce action tlačidlo “solve”, ktoré sprístupni proces riešenia úlohy. Každá úloha je sprevádzana sprievodným textom, ktorý slúži ako príručka.

Title

Available Task: 10

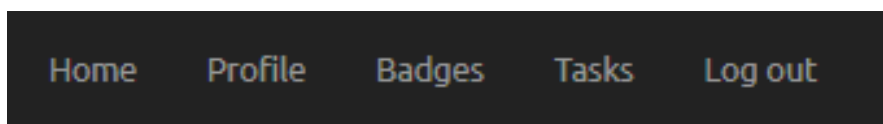
Id	Name	Description	Number Of People	Subtask Cost	Budget	Deadline	Task Category	User	Created At	Action
1	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
2	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
3	test		100	1	1		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
4	test2		100	1	2		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
5	test		100	1	0		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Closed
6	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
7	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
8	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
9	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve
10	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Solve

Obr. 112: .

5.5 Sekcia pre zadávateľa

Hlavné menu ponúka nasledovné možnosti:

- Home - domovská stránka so základnými informáciami o používaní systému a inými detailmi podľa aktuálnej ponuky (v aktuálnej verzii novinky a základne informácie, štatistiky zadaných úloh a výsledkov)
- Profile - profil používateľa. Informácie o používateľovi, ktoré je možné prepojiť so sociálnymi službami a tým získať ďalšie výhody
- Badges - achievement-y alebo úspechy, ktoré používateľ dosiahol/môže dosiahnúť, číslo pri odkaze značí počet nových achievement-ov
- Tasks - zoznam zadaných úloh
- Logout - odhlásenie zo systému



Obr. 113: .

5.5.1 Profil

V aktuálnej verzii totožný s profilom pracovníka

5.5.2 Badges

V aktuálnej verzii totožný s pohľadom pre achievement-y pracovníka

5.5.3 Tasks

Zoznam úloh v riešení. Každý task má v stĺpci action nasledovné tlačidlá:

- edit - upravenie úlohy - funkcionality analogická k funkcii vytvorenia úlohy
- results - pohľad na výsledky vo forme grafov a štatistík

- delete - odstránenie úlohy zo zoznamu riešených

Funkcionalita vytvorenia úlohy (tlačidlo “create new task”), po rozkliknutí zobrazí postupnosť krokov potrebných k vytvoreniu úlohy.

Your current tasks list

Create new task

Id	Name	Description	People count	Subtask cost	Budget	Deadline	Category	User	Created	Action
1	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
2	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
3	test		100	1	1		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
4	test2		100	1	2		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
5	test		100	1	0		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
6	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
7	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
8	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
9	test		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete
10	test2		100	1	500		Picture Tagging	Jason Derulo	2013-12-11 16:55:50 UTC	Edit Results Delete

Obr. 114: .

Postup vytvorenia úlohy

1. krok - Z ponuky vyberte typ úlohy ktorú chcete zadať.

New Task

1 Step 1 Task Category 2 Step 2 Task Details 3 Step 3 Subtasks

Task Category

Task category

Default

Previous Next Finish

Obr. 115: .

2. krok - Vyplňte informácie o úlohe

New Task

The screenshot shows the 'New Task' form at Step 2, 'Task Details'. At the top, there are three step indicators: Step 1 (Task Category) in green, Step 2 (Task Details) in orange, and Step 3 (Subtasks) in grey. The main form area is titled 'Task Details' and contains the following fields:

- Name:** A single-line text input field.
- Description:** A multi-line text area.
- Subtask cost:** A text input field with a vertical scrollbar on the right.
- Budget:** A text input field with a vertical scrollbar on the right.
- Number of people:** A text input field with a vertical scrollbar on the right.
- Deadline:** A date selection interface with dropdown menus for day (1), month (December), year (2013), and hour (18), followed by a time dropdown (02) and a minus sign.

At the bottom of the form, there are three buttons: 'Previous' (disabled), 'Next' (active), and 'Finish' (disabled).

Obr. 116: .

3. krok - Podľa typu úlohy vyplňte požadované polia a odovzdajte úlohu

New Task

The screenshot shows the 'New Task' form at Step 3, 'Subtasks'. At the top, the step indicators are: Step 1 (Task Category) in green, Step 2 (Task Details) in green, and Step 3 (Subtasks) in orange. The main form area is titled 'Subtasks' and contains a 'Questions' section with the following elements:

- Questions:** Two rows of input fields. Each row has an 'Image from' dropdown, a text input field, a 'TextBox' dropdown, another text input field, and a 'Remove' button.
- Add Question:** A green button located below the first row of questions.

At the bottom of the form, there are three buttons: 'Previous' (disabled), 'Next' (disabled), and 'Finish' (active).

Obr. 117: .

New Task

The screenshot shows the 'New Task' form at Step 3, 'Subtasks'. At the top, the step indicators are: Step 1 (Task Category) in green, Step 2 (Task Details) in green, and Step 3 (Subtasks) in orange. The main form area is titled 'Subtasks' and contains a 'Facebook URLs' section with the following element:

- Add Link:** A green button located below the 'Facebook URLs' title.

At the bottom of the form, there are three buttons: 'Previous' (disabled), 'Next' (disabled), and 'Finish' (active).

Obr. 118: .

4. krok - odovzdanie úlohy na riešenie a obmedzenie viditeľnosti

úloha je odovzdaná od ukončenia kroku 3. V záverečnej fáze je možno dodatočne upraviť viditeľnosť úlohy len pre špecifické skupiny používateľov pomocou dostupných filtrov. Filtrovateľné atribúty(source type):

- krajina pôvodu
- počet vyriešených úloh

Operátory:

- == - rovnosť
- <= - menší alebo rovný
- >= - väčší alebo rovný
- < - menší
- > - väčší
- includes - zahŕňa

Value (hodnota) : druhý operand zvolenej operácie filtra

Stlačením tlačidla “create filter” bude filter vytvorený a aplikovaný.

Task was successfully created.

Name:	Task
Description:	Description
Number of people:	10
Deadline:	2013-12-11 20:42:00 UTC
Task category:	Survey
User:	Jason Derulo

:

Sourcetype

User Country

Conditiontype

==

Value

Obr. 119: .

6 Testovanie treťou stranou

Pre overenie správneho návrhu aplikácie bolo nutné otestovať aplikáciu nezávislými používateľmi. Je dôležité aby aplikáciu boli schopní používať širšej skupine používateľov.

Aby testy prebehli úspešne a boli čo najpresnejšie stanovili sme si pevné pravidlá pre výber používateľov na testovanie:

1. V skupine používateľov sa musia nachádzať obe pohlavia.
2. Veková skupina používateľov od 18 do 45 rokov. Chceme týmto kritériom overiť použitie hlavne na skupine používateľov na našej škole, a to sú študenti od prvého ročníka bakalárskeho stupňa až po učiteľský personál, u ktorého očakávame hlavne zadávanie úloh cez náš systém študentom a ostatnej verejnosti. Týmto ohračením nevyklúčujeme možné využitie mladším a starším používateľom. Vek nie je jediným rozhodujúcim kritériom. Hlavným kritériom sú zručnosti s počítačom a internetom.
3. Ani jeden z používateľov nesmie mať skúsenosť s našou aplikáciou.
4. Všetci používatelia musia mať rovnaké podmienky na testovanie, To znamená, že používatelia nemôžu získať viac informácií, pracovať na rôznych verziách a inom technickom zariadení, byť pod nátlakom.

Aby všetci používatelia mali rovnaké podmienky vytvorili sme im jednotný popis aplikácie, ktorý si museli prečítať pred začatím testovania. Po prečítaní sme používateľom neposkytli žiadne iné informácie o aplikácii. Text určený používateľom je nasledovný:

Pinta.sk je webová aplikácia určená pre vytváranie a vypracovávanie rôznych typov úloh od dotazníkov Survey, oznámenia Notice, facebook likes až po opis obrázku Picture taggig. Tieto úlohy sa v aplikácii nazývajú kampane. Používateľ za vypracovanie kampane získa platobnú menu nazvanú budges. Za budges je schopný používateľ vytvárať svoje vlastné kampane, ktoré vypracujú iní používatelia. Používateľ má svoj profil, kde sú uložené rôzne informácie o jeho znalostiach, vedomostiach a skúsenostiach. Tieto údaje je možné použiť pri vytváraní kampane, kedy môžete zadať presné znalosti, vedomosti a skúsenosti používateľa. Následne sa kampaň zobrazí iba používateľovi splňujúcemu tieto kritéria.

Tieto údaje sme používateľom ponechali počas celého testovania. Samotný test bol uskutočnený s 5 používateľmi, splňujúcimi pravidlá popísané vyššie. Každý s používateľov dostal za úlohu vykonať zadané úlohy v aplikácii. Čas vykonávania úloh sme im merali. Zoznam informácií o používateľov môžeme vidieť nižšie v tabulke. Úlohy pre používateľov boli registrácia, vyplnenie profilu, vytvorenie kampane, vypracovanie kampane. Boli zoradené tak ako si predstavujeme, že bude postupovať reálny používateľ. Zoznam úloh s popisom aký mali používatelia môžeme vidieť nižšie.

Úlohy

Každú úlohu si podrobne prečítajte a pokúste sa ju vypracovať. Po vypracovaní každej úlohy zavolajte dohliadajúceho študenta pre overenie správnosti.

používateľ	pohlavie	vek
user1	male	18
user2	famale	20
user3	famale	21
user4	male	24
user5	male	45

Obr. 120: Tabuľka 1

1. Úloha

Registrujte sa na stránke www.Pinta.sk, ktorú máte už otvorenú vo svojom webovom prehliadači. Za prihlasovacie meno použite: test@pinta.sk Heslo: password

2. Úloha

Vyplňte profil používateľa. Profil nájdete v pravom hornom menu, po kliknutí na test@pinta.sk. Po zobrazení profilu upravte nasledujúce sekcie a hodnoty podľa pokynov nižšie. Basic info:

- Birth date: váš dátum narodenia
- Gender: vaše pohlavie
- Ostatné veci v Basic info nevyplňajte.

Education:

- School name: FIIT
- Field of study: IT
- Start date: 2012
- End Date: 2013
- Degree: 2

Language:

- Name: English
- Proficiency: Elementary proficiency

Skills:

- Name: programátor

3. Úloha

Vytvorte kampaň typu Survey. Stlačte v hornom menu test@pinta.sk, následne stlačte tlačidlo My Campaigns. Zobrazí sa vám nová stránka na nej stlačte tlačidlo Create new campaign. Následne zvolte typ kampane Survey. Zobrazí sa vám formulár vyplňte ho nasledovne:

- Meno: test používateľ
- Popis: toto je test
- Budget: 1
- Solution amount: 2
- DeadLine: 30.6.2014

Po vyplnení stlačte tlačidlo Next. Následne sa vám zobrazí okno kde budete mať na ľavej strane Contents a v strede okno. Pretiahnite Single Choice z ľavého menu do stredného okna. Následne sa vám zobrazí formulár a ten vyplňte nasledovne:

Question: Aké je dnes počasie?

Choices:

- Dobré
- Zlé

Potvrďte a stlačte tlačidlo Done a následne v ľavom dolnom rohu Finish.

4. Úloha

Vypracujte kampaň Survey s menom „Dotazník“, ktorý nájdete po stlačení v hornom menu na Campaigns. Pre spustenie dotazníka treba stlačiť na tlačidlo Solve, po ktorom sa zobrazí prvá otázka dotazníku. Po úspešnom vyplnení otázky stlačte tlačidlo Next step, takýmto spôsobom vyplňte všetky otázky v dotazníku. Po úspešnom vyplnení sa dotazník uzavrie.

Obsah dotazníku

Koľko hodín denne strávite za počítačom:

0-1 1-3 5 a viac

Koľko hodín strávite na internete:

0-1 1-3 5 a viac

Ako by ste ohodnotili prívetivosť aplikácie?

neprehľadná/zložitá dobrá/dá sa používať veľmi prívetivá/prehľadná

Robili vám niektoré zo zadaní problémy? Ak áno ktoré (zakrúžkujte aj viacej úloh)

1úloha 2úloha 3úloha 4úloha

Viete si predstaviť nejaké zadanie, ktoré by ste mohli dať niekomu vypracovať pomocou našej aplikácie?

Áno Nie

Zmenili by ste niečo na našej aplikácii?

Áno Nie

Zvolenie pevne stanovených údajov pri testovaní bolo cielené. Vyhýbame sa tomu, aby jeden používateľ zadával dlhšie údaje, alebo nad nimi dlhšiu dobu premýšľal. Cieľom testu nie je kreativita používateľa, ale meranie zručnosti a kvality systému. Rovnako je cieľom aby mali všetci používatelia rovnaké podmienky.

6.1 Výsledky testovania

Výsledky testovania pre prehľadnosť a lepšie vyhodnotenie boli upravené do nasledujúcej tabuľky:

Používateľ	Úloha1		Úloha2		Úloha3		Úloha4	
	čas trvania v minútach	popis	čas trvania v minútach	popis	čas trvania v minútach	popis	čas trvania v minútach	popis
user1	1		10		8		6	
user2	2	Pomýlil sa pri zadávaní znakov.	12		15		10	Dlho premýšľal nad odpoveďami.
user3	2		11		9		9	
user4	1		10		8		8	
user5	2		15	Zo začiatku nevedel ako pracovať s užívateľským rozhraním.	17	Problém s prenesením častí Survey do stredného panelu.	14	Pomalý a dôkladne vypracoval čítal si otázky aj niekoľko krát.
priemerné hodnoty	1.6		11.6		11.4		9.4	

Obr. 121: Tabuľka 2

Úloha 1

Podľa výsledkov a očakávaní bola najjednoduchšia. Priemerný čas jej vypracovania bol pod 2 minúty. Používatelia s ňou nemali problém. Ľahko a intuitívne našli registračné tlačidlo a registrovali sa s pripravenými údajmi. Problém niektorým robilo správne opísanie prihlasovacieho mena a hesla.

Úloha 2

V porovnaní s úlohou 1 trvala dlhšie. Používatelia však nemali problém zorientovať sa v aplikácii a nájsť v nej profil. Po nájdení profilu user5 mal problém zistiť ako sa editujú jednotlivé časti profilu. Najprv klikal na jednotlivé položky, neskôr na ich pomenovanie. Po chvíli si však všimol editačného okna a následne už nemal problém s vyplnením profilu. Taktiež najdlhšiu časť vyplňania profilu zabralo používateľom ich opísanie z papiera. Priemerný čas aplikácie trval používateľom pod 12 minút.

Úloha 3

Táto úloha sa ukázala ako najťažšia. Priemerný čas vypracovania úlohy trval pod 12 minút. Najdlhšia práca trvala user5, ktorý strávil na vypracovávaní úlohy až 17 minút. Problém bol s prenesením Single Choise do stredného okna. Príčina

problému spočíval v tom, že sa používateľ s daným prvkom nestretol pri práci na internete. Úlohu úspešne vypracovali všetci používatelia.

Úloha 4

Nájdanie kampane a následné vypracovávanie používateľom nerobil problém. Čas vypracovania sa v priemere pohyboval pod 10 minút. Používatelia museli vyplniť 6 otázok. Niektorých používatelia dlhšie premýšľali nad odpoveďami.

V prvej otázke dotazníka sme sa používateľov pýtali na počet strávených hodín za počítačom. Je potrebné brať do úvahy aj znalosť používateľa s prácou na počítači. Rýchla práca na počítači mohla v našom prípade značne vplývať na čas vypracovania úloh. Druhá otázka nám mala napomôcť zistiť do akej miery má používateľ znalosť internetu. V našej webovej aplikácii sa nachádza veľa moderných ovládacích užívateľských prvkov. Používateľ, ktorý sa s nimi ešte nestretol by mohol mať problémy. V našom prípade sa testu zúčastnili používatelia s lepšou skúsenosťou s počítačom. 3 používatelia strávia denne 5 a viac hodín a dvaja medzi 1- až 3 hodinami denne. Neprekvapilo nás ani zistenie, že dĺžka strávená za počítačom a na internete bola rovnaká. V dnešnej dobe je internet častým pomocníkom pri práci na počítači, samotná práca sa nachádza priamo na internete, alebo pripojenie k internetu je nevyhnutné pre vykonávanie práce.

Ďalšia otázka sa týkala prívetivosti aplikácie. Z testu nám vyplýva, že staršia generácia ktorá trávi na internete a počítači, rovnaký čas ako mladšia, má pocit, že aplikácia je dobrá/ dá sa používať. Preto do úspešnosť použitia na našej aplikácii vplývajú dva faktory a to skúsenosti s internetom a počítačom a vek používateľa, ktorý zasahuje do zručností. V otázke Robili vám niektoré zo zadaní problémy? Až dvaja používatelia označili úlohu číslo 3. Na tento fakt mohlo pôsobiť hneď viacero faktorov. Táto úloha bola najkomplikovanejšia, bola zdĺhavá a textový popis mohol byť nezrozumiteľný. Na úspešnosť úlohy mohlo vplývať aj unavenosť používateľov, ktorých práca v našej aplikácii presahovala aj viac ako pol hodinu bez prestávky.

Ďalšia otázka bola Viete si predstaviť nejaké zadanie, ktoré by ste mohli dať niekomu vypracovať pomocou našej aplikácie? Na túto otázku všetci používatelia odpovedali áno. Spýtali sme sa ich po teste na konkrétne príklady. Ich odpovede môžeme vidieť nižšie.

user1	Facebook like
user2	Vytvorenie dotazníku do školy
user3	Vytvorenie dotazníku pre bakalárky
user4	Vytvorenie dotazníku na diplomovku
user5	Vytvorenie formulařu do práce

Obr. 122: Tabuľka 3

Väčšina odpovedí sa týkala typu Survey. Toto mohlo ovplyvniť fakt, že ostatné typy používatelia nemali vyskúšané. Survey sme ich nechali vytvoriť v úlohe 3. Jeden používateľ odpovedal Facebook like. Fakt, že každý používateľ by vedel najst aspoň jeden príklad využitia našej aplikácie je veľmi priaznivý.

Ďalšia otázka znela Zmenili by ste niečo na našej aplikácii? Až traja používatelia odpovedali na otázku áno. Po testovaní sme sa ich spýtali na konkrétne typy, viac tabuľka nižšie. Dvaja používatelia, u ktorých boli namerané najdlhšie časy

user1	
user2	Viac nápodved pri vytváraní kampane.
user3	Možnosť komunikácie s inými používateľmi.
user4	
user5	Viac vysvetliviek.

Obr. 123: Tabuľka 4

vypracovávaní úloh by ocenili viac nápodvedí v systéme. Jeden používateľ by rád komunikoval s ostatnými používateľmi.

Pre porovnanie nameraných priemerných výsledkov s časom, ktorý trval programátorovi, so skúsenosti s aplikáciou, viac tabuľka nižšie:

úloha	priemerný čas(min)	najkratší čas (min)	najdlhší čas (min)	čas programátora (min)
úloha1	1.6	1	2	1
úloha2	11.6	10	15	7
úloha3	11.4	8	17	4
úloha4	9.4	6	15	3

Obr. 124: Tabuľka 5

Môžeme vidieť, že čas programátora bol pri všetkých úlohách nižší. Bolo to spôsobené jeho pokročilou znalosťou aplikácie.

6.2 Vyhodnotenie testovania

Testy ukázali, že našu aplikáciu sú schopné úspešne použiť viaceré generácie používateľov od 18 až po 45 rokov s rôznymi skúsenosťami s počítačom a internetom. Testy taktiež ukázali, že používanie aplikácie si vedia používatelia predstaviť na reálnych príkladoch, ktoré boli schopný sami uviesť. Veľmi nepriaznivým faktorom boli pevne stanovené údaje, ktoré používatelia opisovali do systémy. Zručnosti rýchleho písania boli u používateľov rozdielne a tvorili hlavný dôvod veľkých rozdielov nameraných časov vypracovania úlohy. Pre zvýšenie použiteľnosti je nutné doplniť prostredie o lepšie a presnejšie vysvetlivky. Najhoršie výsledky nameraných časov ukazovala úloha číslo 3. Hlavným problémom bolo jej textové zadanie, ktoré nebolo vhodné. Lepšou variantnou by bolo využiť kombináciu textu a obrázkov aplikácie, poprípade umožnenie používateľovi naštudovania si pomocníka nachádzajúcej sa v aplikácií.

7 Čo sme nestihli

Napriek snahe splniť všetky stanovené ciele a implementovať systém so všetkými navrhnutými funkcionalitami, sa nám nie všetko podarilo. V ďalšej časti sú opísané, ciele a úlohy, ktoré sa nepodarilo splniť spolu s príčinou.. Tieto sú taktiež bodmi vhodnými pre rozpracovanie v pokračujúcom vývoji.

Vyvíjať prístupom TDD

Napriek snahe sa nám nepodarilo zahrnúť do vývoja pravidelné testovanie. Dôvodom bol slabé skúsenosti s použitými technológiami, kde vývojari často strávili väčšinu času spoznávaním RoR. Taktiež vyvinutá funkcionalita prebiehala viacerými úpravami, čo by viedlo k redundantnej práci. Absenciu TDD prístupu sme kompenzovali zahrnutím testovacej fázy, v ktorej sme manuálnym testovaním odhalili väčšinu chýb, ktoré sme následne opravili.

Vytvoriť platobnú bránu

V súčasnosti systém ne poskytuje možnosť zakúpenia virtuálnej meny prostredníctvom platobnej brány. Problematika tejto oblasti je príliš rozsiahla a nesie so sebou aj značnú mieru zodpovednosti. Počas vývoja sme si museli priority upraviť tak aby sme podchytili hlavnú funkcionalitu. Absencia platobnej brány a aj riešenie problému “sliepka a vajce” bolo iníciaľne poskytnutie našej virtuálnej meny pre prvých zadávateľov a vytvorenie našich vlastných úloh s primeraným ohodnotením pre zárobok nových používateľov.

Dynamické filtre pre výsledky

Pôvodný úmysel implementovať možnosť filtrovania výsledkov aj na základe používateľom definovaných filtrov (pozor: nepomýliť s cieľným zadávaním - filtre respondentov). Túto funkcionalitu sme sa snažili implementovať avšak kvôli (z pohľadu filtrovania) nevhodnej reprezentácii riešení úloh v dátovom modeli je neozhné implementovať dynamické filtre efektívnou cestou. V aktuálnej verzii sme poskytli len základnú sadu statických filtrov.

Verejný API

Možnosť prístupu do aplikácie aj prostredníctvom verejného api sme zahrnuli do riešenia. Avšak komplexnosť problému sme podcenili. Základná funkcionalita, ktorá však nie je kompletná, už existuje. V zásade sme schopní umožniť pridávanie bodov cez API avšak toto nie je dostatočne široká funkcionalita a nebola plne prepojená z implementovanými typmi úloh.

8 Čo sme sa naučili

Túto časť síce vypracoval len jeden človek avšak so súhlasom zúčastnených tímových kolegov. Všetci tento predmet považujeme za prínosom do našich pokračujúcich životov, či už študijných alebo pracovných.

Pri riešení problému sme sa stretávali s množstvom problémov. Počnúc problémom tímovej spolupráce a kooperácie. Nie všetci aktéri už mali tu česť pracovať v nejakom tíme. Všetkým, aj tím skúsenejším, prídu skúsenosti z tímovej práce na projekte, ktorý bol navrhovaný od “piky”, určite vhod. Naučili sme sa *organizácii práce*, s ohľadom na rozvrhnutie práce medzi viacerých ľudí.

Fráza „100 hláv 100 rozumov“ poukazuje na jedinečnosť každej osobnosti. Tento fakt bol počas našej spolupráce značne výrazný. Prejavoval sa často viac ako bolo dobré. Nie vždy sa nám podarilo správne odhadnúť schopnosti jednotlivca, čím tým utrpela nie len kvalita produktu ale aj morálka v tíme. Toto sme však skúsenosťou a časom dokázali riešiť preskupovaním síl medzi úlohami a vzájomnou pomocou. Tieto a nielen tieto skúsenosti sa dajú zastrešiť pod jednu - „s“ chopnosť tímovej práce. Aj v rámci tímovej spolupráce sa nám darilo predchádzať možným problémom vďaka metodike vývoja metodikou SCRUM, ktorá vniesla do tímu väčšiu slobodu jednotlivca a primeranú mieru organizovanosti. Hlavnú výhodu oproti klasickým prístupom hierarchického štruktúrovania tímu vidíme v efektívnosti práce malého počtu ľudí, prípadne viacerých podskupín na projektoch až stredne veľkých rozsahov.

Okrem cenných skúseností z tímovej práce sme sa naučili vnímať projekt ako celok. Snažili sme sa, aj pomocou procesu Code Review, udržiavať prehľad o celom projekte a tým pomôcť *zachovaniu tzv. Big Picture*. Tomuto napomáhal aj podproces retrospektívy, ktorý je súčasťou metodiky SCRUM.

Vďaka prerozdeleniu manažerských rolí jednotlivým aktérom a vypracovaním vlastných metodík sme sa naučili, že pevne stanovené pravidlá napomáhajú kvalite vývoja a vyvíjaného produktu. Tohoto faktu sme si vedomí, aj keď sa nám kvôli časovým obmedzeniam a absencii skúseností nepodarilo uplatniť metodiky vždy na 100%. Hovorí nám, že aj *pravidlá sú nutné*.

V skratke:

- tímová práca je základ úspechu
- význam “big picture” nepodceňovať
- aj pravidlá sú nutné
- plánovať, plánovať, plánovať
- aj povedané je nutné zdokumentovať (zápisnice - “ja som nič také nepovedal”)
- Nepredpokladaj! Pýtaj sa!
- Rýchla, častá spätná väzba = menej roboty = nižšia cena = vyššia kvalita
- každý ma svoje prednosti, len ich treba nájsť

PRÍLOHY

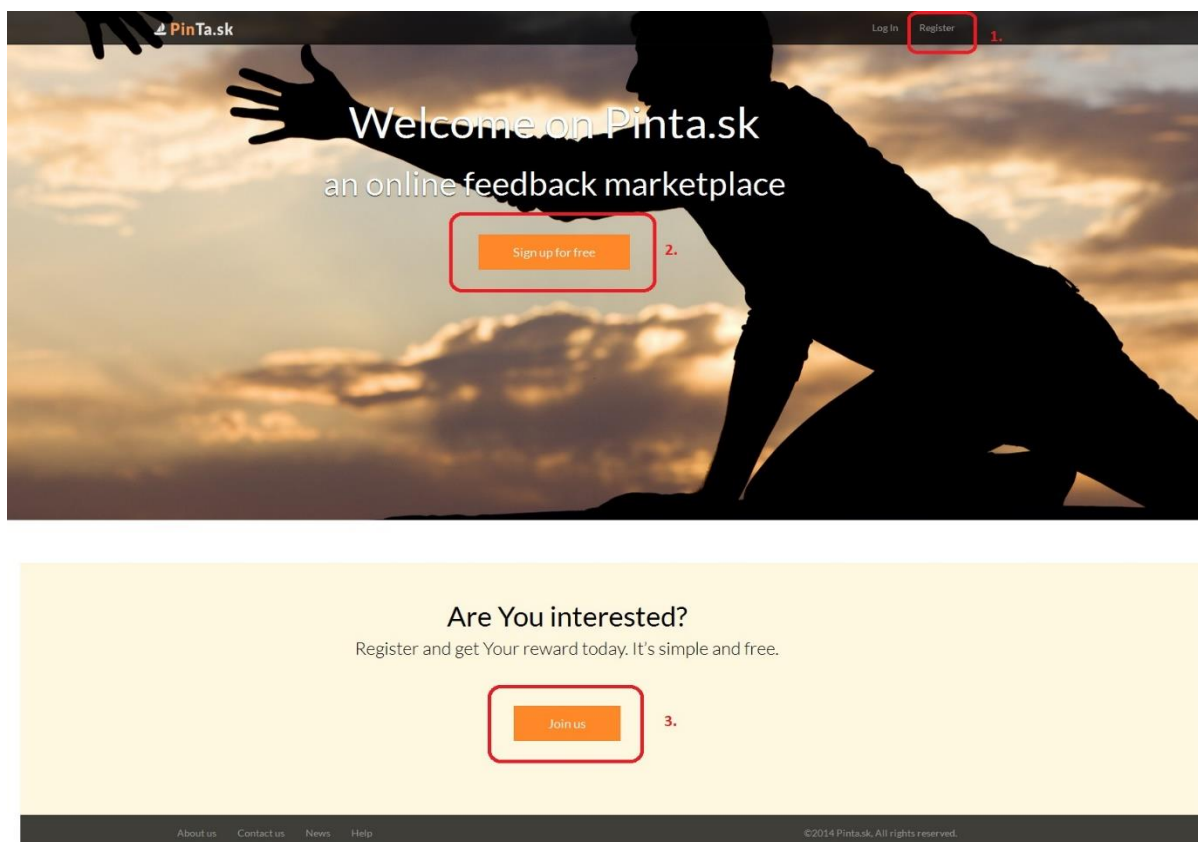
Príloha A

Používateľská príručka Pinta.sk

V tejto časti je popísaný spôsob použitia aplikácie používateľom. Aplikácia verejne dostupná na webovej stránke <http://www.pinta.sk>. Lubovoľný používateľ môže webovú stránku navštíviť a aplikáciu využiť.

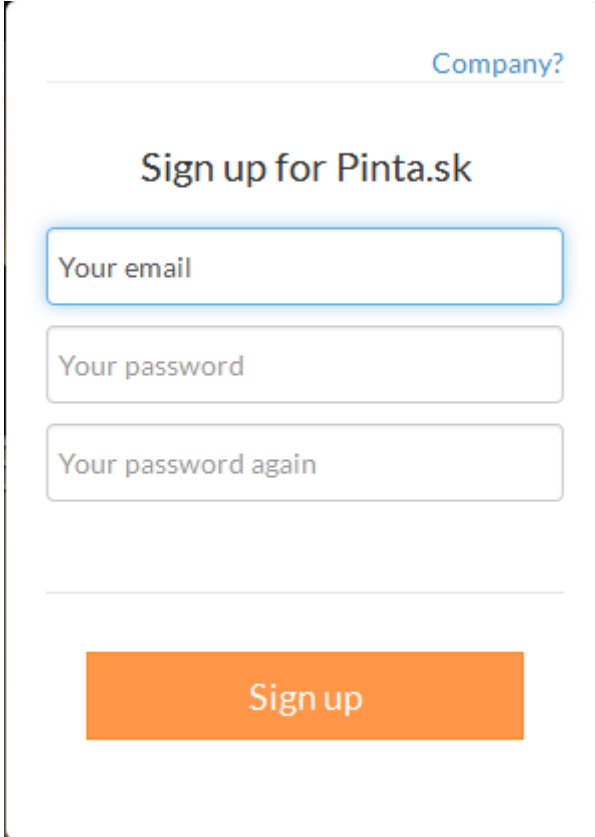
Registrácia používateľa a prihlásenie

Po navštívení <http://www.pinta.sk> sa používateľovi zobrazí úvodná stránka, na ktorej sú popísané základné informácie o aplikácii. Pre vstúpenie do aplikácie je potrebné byť registrovaný. Preto prvé čo musí nový používateľ po navštívení našej stránky urobiť je zaregistrovať sa. Registrácia ako aj všetka funkcionality je zadarmo. Celá aplikácia je školský projekt a jej cieľom je poskytovať svoje služby verejnosti. Pre registráciu je potrebné stlačiť na jednom z troch registračných tlačidiel nachádzajúcich sa na stránke ako môžeme vidieť na Obrázku A.1.



Obr. A.1: Obrázok zobrazujúci tlačidlá pre registráciu používateľa

Prvé dve sa nachádzajú v hornej časti stránky a tretie tlačidlo sa nachádza v dolnej časti stránky. Tak je to navrhnuté kvôli zvýšeniu komfortu používateľa aby sa ľahko mohol registrovať z ktorejkoľvek časti stránky bez toho, aby sa musel vracat' k jednému tlačidlu. Po stlačení tlačidla sa zobrazí registračný formulár, viac Obrázok A.2 nižšie.



The image shows a registration form for Pinta.sk. At the top right, there is a blue link labeled "Company?". Below this, the title "Sign up for Pinta.sk" is centered. The form consists of three input fields: "Your email", "Your password", and "Your password again". At the bottom, there is a large orange button labeled "Sign up".

Obr. A.2: Obrázok registračného formuláru.

Vo formulári sa nachádzajú tri okná. V prvom používateľ zadá svoj existujúci email, na ktorý mu bude zaslaný potvrdzujúci email o registrácii. Ďalšie dve okná slúžia na zadanie hesla a jeho kópie pre potvrdenie. Zadaný email bude používateľovi po úspešnej registrácii slúžiť ako prihlasovacie meno a heslo ako heslo k prihláseniu do aplikácie. Okno prihlásenia následne vyzerá takto, viac Obrázok A.3 nižšie.

Welcome back to Pinta.sk

Your email

Your password

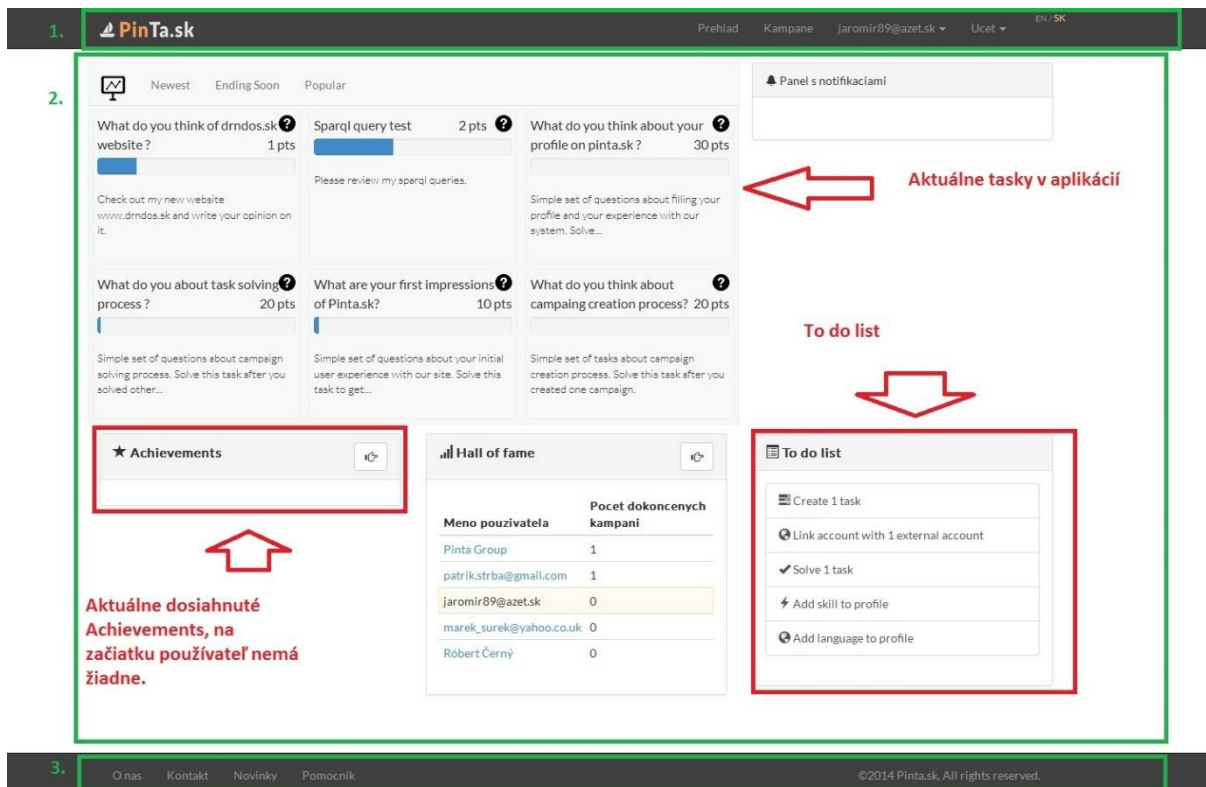
Login

Obr. A.3:Obrázok prihlasovacieho okna.

Po ukončení práce sa používateľ môže odhlásiť pomocou Logout, lebo ho aplikácia odhlási sama po dlhšej nečinnosti.

Popis hlavnej stránky

Po úspešnom prihlásení používateľ bude automaticky presmerovaný na stránku s menom *Dashboard*, ktorá slúži ako hlavná stránka v aplikácii. Viac Obrázik A.4 nižšie.



Obr A.4 : Dashboard hlavná stránka aplikácie. Zelenou sú vyznačené hlavné časti aplikácie, červeným jednotlivé informačné sekcie.

Celá stránka sa dá rozdeliť na tri časti a to horné menu, stredný panel a dolné menu. V hornom menu sa nachádzajú jednotlivé sekcie stránky *Prehľad*, *Kampane*, *Účet*, menu jazykov. Po prepnutí do angličtiny *DashBoard*, *Campaigns*, *Wallet*. Používateľ si môže vybrať z dvoch dostupných jazykov a to Angličtina a Slovenčina. Predvoleným jazykom je Angličtina. Ešte sa tam nachádza rozbaľovací panel s funkcionalitou účtu a na ľavej strane logo pomocou ktorého sa používateľ vráti na *Dashbord*.

V Strednej časti sa nachádza viacero informačných častí pre používateľa. Cieľom stránky je podať používateľovi všetky základné informácie o systéme a stave používateľa k systému. Stránka zobrazuje niektoré kampane, ktoré používateľ môže riešiť, rovnako zobrazuje aj používateľom získané *Achievements*, ktoré získal v systéme splnením určitých kritérií. Na stránke v dolnej časti používateľ môže nájsť zoznam používateľov a svoju pozíciu v zozname. V právo dole sa nachádza aj *To do list* v ktorom používateľ môže nájsť rady, k ďalším akciám na stránke.

V Tretej časti stránky uloženej v dolnej časti sa nachádza informačné menu s *O nás*, *Kontakty*, *Novinky* a *Pomocník* v angličtine *About us*, *Contact us*, *News*, *Help*. Dole nižšie môžeme vidieť Obrázok A.5 nápovedy a formulár pre kontaktovanie z *Contact us* stránky na *Obrázku A.6*.

PinTa.sk an online **feedback** marketplace

ČO JE Pinta.sk ?

Pinta.sk je moderné webové rozhranie na sprostredkovanie mikro-práce. Nazov je odvodený z anglického "pin task", príjmi úlohu. Pinta.sk sa od ostatných konkurenčných aplikácií líši moderným a prepracovaným používateľským rozhraním, podporou veľkého množstva typov úloh a zábavným a pútavým spracovaním, ktoré Vás núti "pracovať" viac a viac. Pre pracantov máme pripravené :

- prepracované a príjemné používateľské rozhranie,
- kreditný systém a možnosti výmenného obchodu,
- systém evolúcie (vývoja používateľa),
- karma systém (obľúbenosť úloh, zadávateľov aj pracantov),
- sociálnu integráciu a možnosť súperiť s priateľmi.

Zadávatelia sa môžu tešiť na :

- množstvo nažhavených pracantov,
- rozsiahle možnosti definovania úloh s vysokou mierou automatizácie,
- možnosti detailne špecifikovať cieľovú skupinu na základe :
 - pohlavia,
 - veku,
 - krajiny,
 - vzdelania,
 - zručností,
 - karmy,
 - a iných atribútov.
- A v neposlednom rade na kvalitnú podporu pripravenú pomôcť v akejkoľvek situácii.

Obr. A.5: Stránka About Us, kde sú informácie o tímovom projekte.

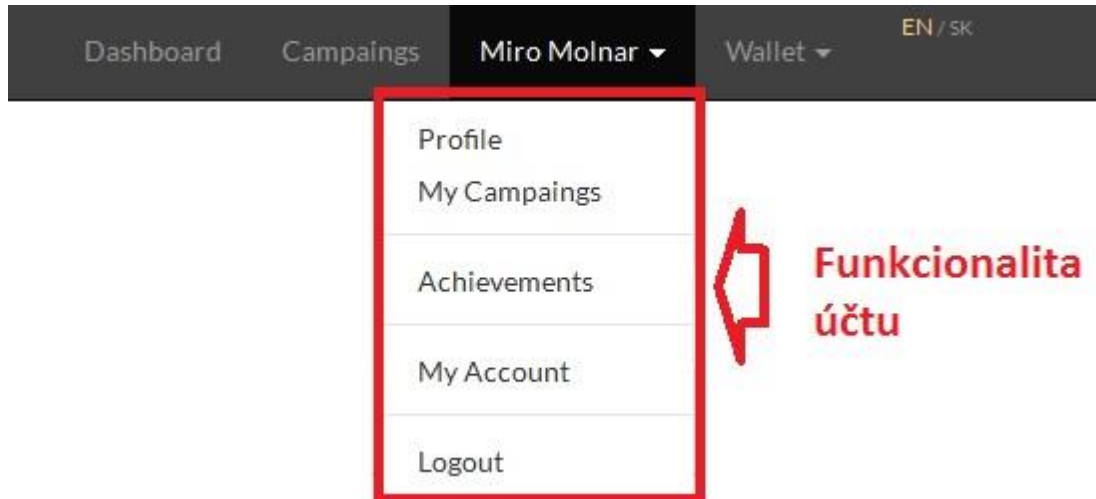
Send A message to Us

Your message..

Send message

Obr. A.6: Stránka *Contact us*, kde nás používateľ môže v prípade problému kontaktovať prostredníctvom emailu.

Funkcionalita účtu sa skladá z Profile, My Campaings, Achievements, MyAccount a Logout viac Obrázok A.7 nižšie. Používateľ v aplikácii je okrem budged ohodnocovaný aj pomocou achievements, ktoré získava za splnenie určitých podúloh v systéme, viac Obrázok A.8. Zobrazenie stránky My Acount môžeme vidieť na Obrázku A.9, kde si používateľ môže zmeniť svoje prihlasovacie heslo.



Obr. A.7: Obrázok funkcionality účtu.

Your Achievements



Obr. A.8: Stránka *Achievements* zobrazuje používateľove achievementst.

Your account details

Email

jaromir89@azet.sk

Password

Password confirmation

Current password

Cancel

Update my account

Obr. A.9: Stránka *My Account*, kde si používateľ môže zmeniť svoje heslo k účtu.

Profil používateľa

Prvou vecou, ktorú používateľovi môže pomôcť k rozšíreniu možností je vyplnenie profilu. Veľa kampaní je určených len používateľom s určitými vlastnosťami zadávanými v profile. Preto dôkladné vyplnenie profilu môže používateľovi sprístupniť nové kampane. Profil sa skladá z viacerých častí ako sú *Basic info*, *Positions*, *Education*, *Referral Link*, *Skills*, *Achievements*, *Languages*. Viac Obrázok A.10 nižšie.

The screenshot shows a user profile page with the following sections:

- Basic info**: First Name: Miro, Last Name: Molnar, Gender: Male, Birth Date: January 01, 1900, Age: 114, About Me: (empty)
- Referral Link**: Invite friends to Pinta.sk: <http://www.pinta.sk/referral=21> Copy link
- Languages**: (empty)
- Positions**: (empty)
- Education**: (empty)
- Skills**: Develop in Java programator language (with close button)
- Achievements**: (empty)

At the bottom, there are three buttons: Update profile with LinkedIn, Link with Facebook, and Link with Google+.

Obr. A.10: Obrázok zobrazujúci stránku profilu používateľa.

Používateľ si môže jednotlivé časti vyplniť a modifikovať. Ak má účet na LinkedIn môže využiť tlačidlo v ľavom dolnom rohu pre načítanie profilu zo sociálnej siete.

Pri stlačení editácie *Basic info* sa otvorí používateľovi modálne okno *Basic info*, kde môže upraviť informácie o ňom. Rovnaký štýl zobrazenia editácie je aj pre ostatné časti profilu, viac Obrázky A.11 – A.15 nižšie.

Basic info ×

Firstname
Miro

Lastname
Molnar

User description

Birth date
1 1 1924

Gender
Male

Update Profile

Obr. A.11: Modálne okno Basic info.

Positions ×

Title

Summary

Start date

End date

Is current

Company

[Create Position](#)

Obr. A.12: Modálne okno Position.

Add education



School name

Field of study

Start date

End date

Degree

Create Education

Obr. A.13: Modálne okno Education.

Add language

Name

Afrikanns

Proficiency

Elementary proficiency

Add to profile

Obr. A.14: Modálne okno Language

Add skill

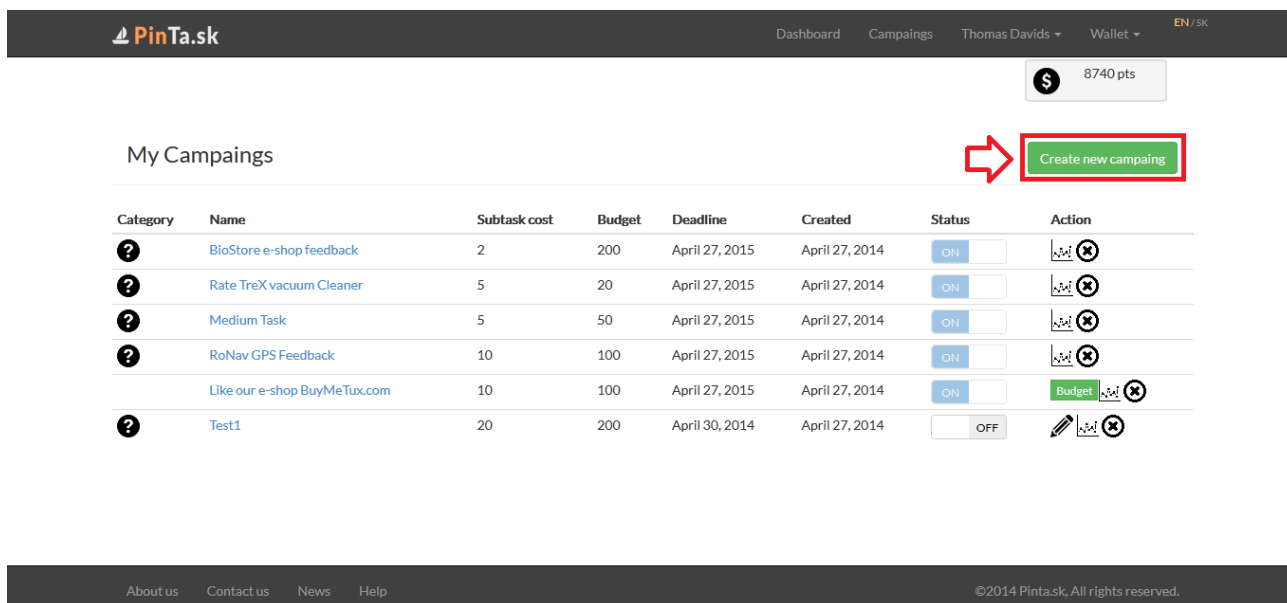
Name

Create Skill

Obr. A.15: Modálne okno Skill

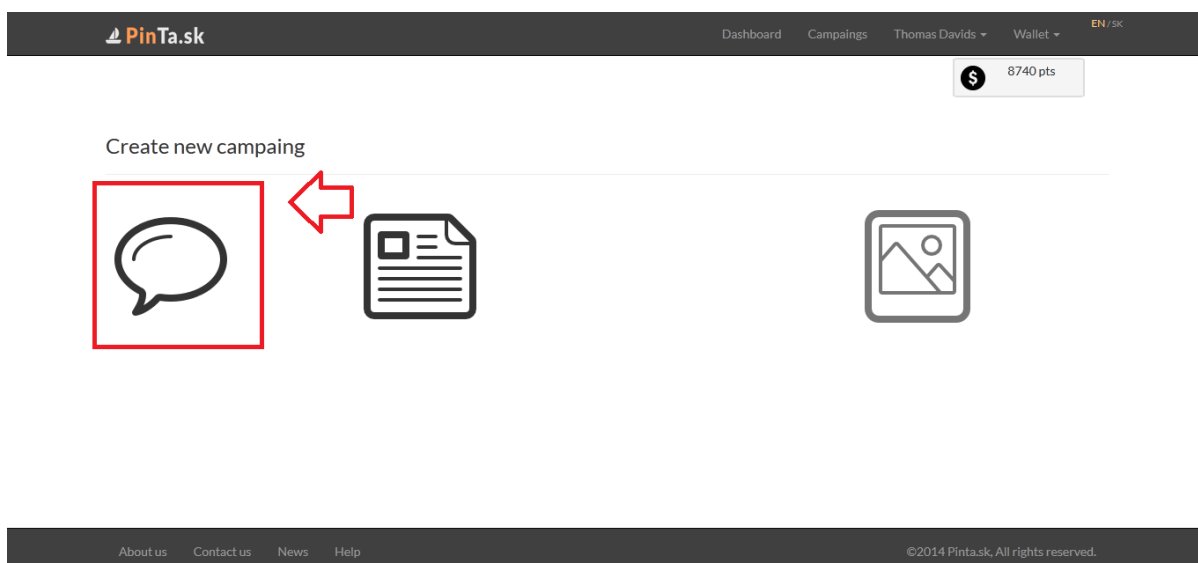
Vytvorenie kampane

Po stlačení *My Campaigns* vo funkcionalite účtu si používateľ môže zobrazit' svoje vlastné kampane, prezerat' aktuálny stav a vykonávat' nad nimi rôzne akcie, ako je vymazať, editovať, zobrazit' výsledky. Rovnako si môže vytvorit' novú kampaň, viac Obrázok A.16 nižšie.



Obr. A.16: Stránka MyCampaings.

Po stlačení na tlačidlo *Create new campaigning* sa zobrazí používateľovi stránka s typmi kampaní. V ponuke sú *Survey*, *Notice*, *Facebook likes*, *Picture tagging*. My si blyžšie popíšeme vytvorenie *Survey* kampane. Musíme stlačiť tlačidlo *Survey* viac Obrázok A.17 nižšie.



Obr. A.17: Stránka s typmi kampani Vybratie Survey kampane.

Po stlačení tlačidla sa nám zobrazí formulár pre vyplnenie bližších informácií o kampani ako je meno, popis, počet budgetov za kampaň, počet riešiteľov, čas respirácie kampane, viac Obrázok A.18 nižšie.

PinTa.sk Dashboard Campaigns Thomas Davids Wallet EN/SK

New Survey

8740 pts

Name

Write description here ...

Budget

Solution amount

Subtask cost

Deadline

Next Cancel

About us Contact us News Help ©2014 Pinta.sk, All rights reserved.

Obr A.18: Stránka formuláru Survey.

Po stlačení *Next* prejdeme na stránku, kde sa vyskladá *Survey* kampaň z jednotlivých jej častí. Vyskladávanie prebieha tak, že si vyberieme jednu časť a pretiahne ju do stredného okna pomocou *DRAG AND DROP* princípu. Viac Obrázok A.19 a A.20 nižšie.

PinTa.sk Dashboard Campaigns Thomas Davids Wallet EN/SK

8740 pts

Content

Single Choice

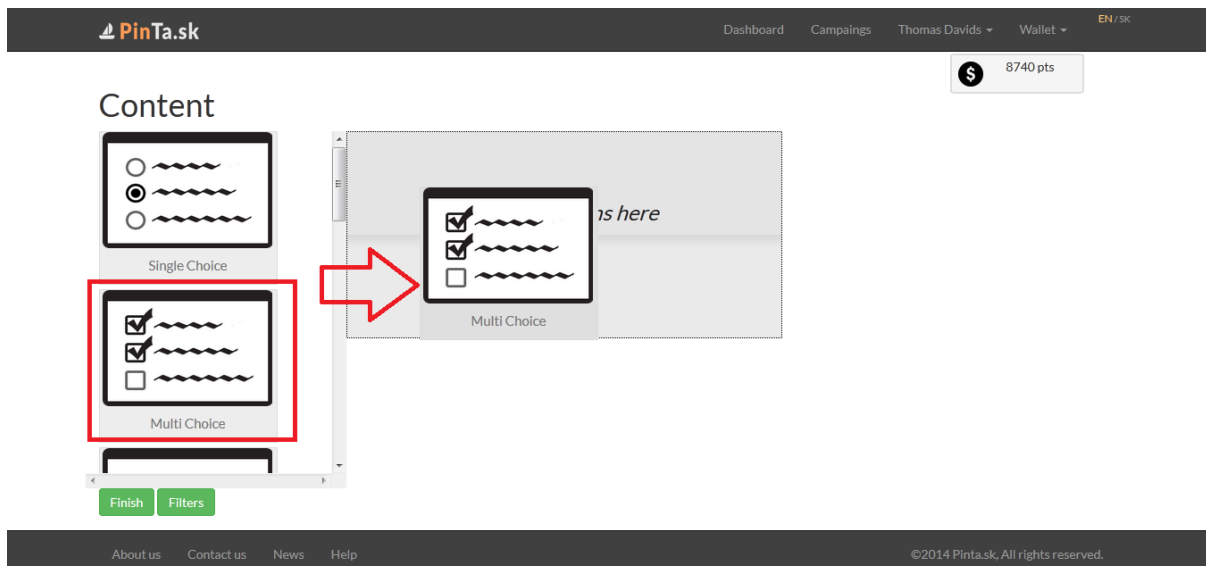
Multi Choice

Drag questions here

Finish Filters

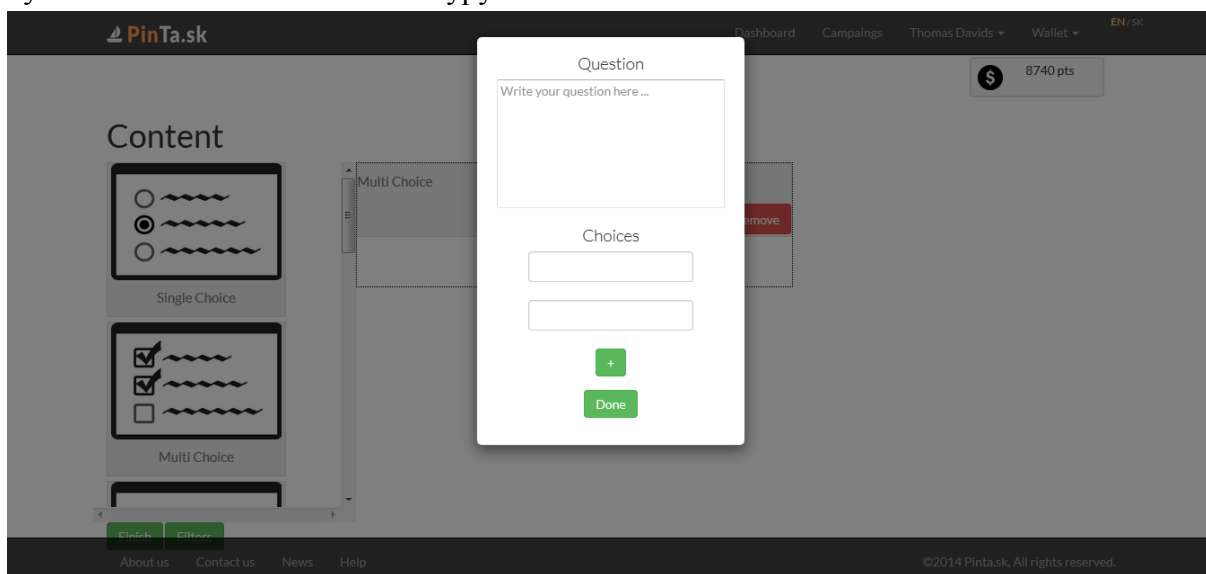
About us Contact us News Help ©2014 Pinta.sk, All rights reserved.

Obr. 19: Hlavné okno s časťami *Survey* na ľavej strane a stredným panelom kam sa pretahujú jednotlivé časti.



Obr. A.20: Okno vytvárania Survey po pretiahnutí časti *Multi Choice* do stredného panelu.

Po pretiahnutí časti je potrebné doplniť potrebné informácie ako sú otázka, a možnosti výberu závisiace od konkrétneho typu časti Viac Obrázok A.21 nižšie.



Obr. A.21: Okno formuláru *Multi Choise*.

Po vyskladaní Survey ak používateľ stlačí tlačidlo *Filters* zobrazí sa okno s filtrami, ktoré pracuje na podobnom princípe ako vytváranie *Survey*, Na ľavej strane sa nachádzajú jednotlivé filtre, ktoré treba presunúť *DRAG AND DROP* do stredného panelu stránky a následne nastaviť ich hodnoty. Viac Obrázok A.22 nižšie.

Filters



Obr. A.22: Obrázok stránky s filtrami.

Po Stlačení tlačidla *Finish* sa používateľovi zobrazí detail vytvorenej kampane, viac Obrázok A.23 nižšie.

Task was successfully created. Task is not activate yet.

Test3

Basic info

Name: Test3
Description: description
Task Category: Survey
Budget: 1
Deadline: May 23, 2014
User: Miro Molnar

Actions

[Edit](#) [Activate](#)

Filters

Source	Condition	Value	Action
--------	-----------	-------	--------

Statistics

Budget left: 1
Starting budget: 1
Subtask cost: 1
Number of subtasks: 1
Number of solutions: 0
Available feedbacks ... 1
Number of valid sol... 0
Task Progress:

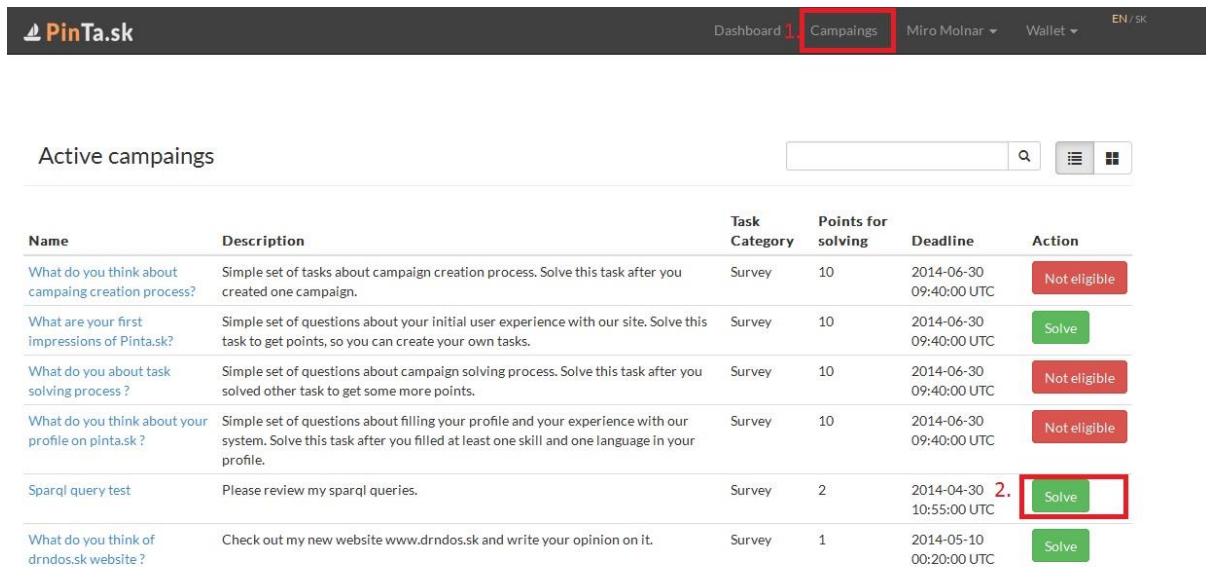
Results

Obr. A.23:Detail novovytvoreného *Survey*.

Rovnako si novovytvorený kampaň môže pozrieť v *My Campaigns*. V prípade problému manipulácie s aplikáciou používateľ môže použiť stránky *Help*, na ktorej je presne popísaný postup vytvorenia kampane.

Vypracovanie kampane

Pre vypracovanie kampane používateľ musí navštíviť stránky Campaigns, na ktorej si zvolí príslušnú kampaň pomocou tlačidla *Solve*. Viac Obrázok A.24 nižšie.



Name	Description	Task Category	Points for solving	Deadline	Action
What do you think about campaign creation process?	Simple set of tasks about campaign creation process. Solve this task after you created one campaign.	Survey	10	2014-06-30 09:40:00 UTC	Not eligible
What are your first impressions of Pinta.sk?	Simple set of questions about your initial user experience with our site. Solve this task to get points, so you can create your own tasks.	Survey	10	2014-06-30 09:40:00 UTC	Solve
What do you about task solving process ?	Simple set of questions about campaign solving process. Solve this task after you solved other task to get some more points.	Survey	10	2014-06-30 09:40:00 UTC	Not eligible
What do you think about your profile on pinta.sk ?	Simple set of questions about filling your profile and your experience with our system. Solve this task after you filled at least one skill and one language in your profile.	Survey	10	2014-06-30 09:40:00 UTC	Not eligible
Sparql query test	Please review my sparql queries.	Survey	2	2014-04-30 10:55:00 UTC	Solve
What do you think of drndos.sk website ?	Check out my new website www.drndos.sk and write your opinion on it.	Survey	1	2014-05-10 00:20:00 UTC	Solve

Obr. A.24: Obrázok z aktuálnymi kampaňami v aplikácii.

Následne sa zobrazujú formuláre, ktoré používateľ vyplňuje podľa pokynov. Viac Obrázok A.25 nižšie.

What are your first impressions of Pinta.sk?

Select each process complexity

registration	<input type="text" value="easy"/>
navigation	<input type="text" value="easy"/>
orientation	<input type="text" value="easy"/>

Next step

Step: 1 / 1

Campaing deadline: June 30, 2014

Obr. A.25: Príklad okna kampane.

Po vyplnení používateľ musí potvrdiť údaje tlačidlom *Next step*.

Aplikácia obsahuje aj v menu *Wallet* po stlačení *My Transations* zoznam transakcií účtu, kde vidí používateľ prehľadne komu koľko points dal za akú činnosť, viac Obrázok A.26 nižšie.

From	To	Points	Reason	Status	Result	Description	Time
------	----	--------	--------	--------	--------	-------------	------

Obr. A.26: Stránka *My Transations*, s transakciami používateľa.

Príloha B

PINTA.SK - Digital Sweatshop

Bc. Filip Bednárík, Bc. Róbert Černý, Bc. Marek Lenčేశ, Bc. Miroslav Molnár, Bc.

Patrik Štrba, Bc. Martin Toma, Bc. Miroslav Vojtuš*

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Ilkovičova 2, 842 16 Bratislava, Slovakia

team11.fiit@gmail.com

Extended abstract

Even today, with all this existing technology, there is still a problem with obtaining fast, high quality feedback from real people. We all know that constructive feedback is a crucial element in many businesses nowadays. Companies hire marketers who try their best to sell their product. Marketers struggle finding people providing valid feedback. Usual approaches like phone marketing, interviews and surveys lack motivation for people to answer. These ways are not entertaining nor do they make any profit for feedback provider.

We extend this problem by looking on term feedback as not only answer to a question or opinion in that matter but also evaluation, some action or micro procedure.

Our main goals we aim to achieve :

- to build online environment for exchanging quality feedback
- to help marketers aim their campaign for specific group of people (age, location, gender, qualification, interests)
- to motivate feedback providers to get involved by rewarding them and by making the process of feedback leaving simple and entertaining
- to provide evaluated feedback results back to marketers

To achieve these goals we propose web-based application for online feedback exchange called “Pinta.sk”. This application should provide a powerful, yet easy to use tools which will help us to achieve these goals. In order to be able to achieve these goals, we must create the ability to use and combine multiple filters to exactly specify desirable group, implement some of the gamification principles, which are very popular nowadays and also very importantly, set up an effective and extendable validation mechanism which will greatly improve feedback processing process.

Feedback requester process.

1. Focus

We provide an ability to set up an unlimited number of filters, which are applied to general policy which determinates who can see that particular campaign. These filters will be also available in result visualizations so there is a possibility to make further analysis even if we have firstly focused on a broad area. And there should also be ability to change and extend these filters during active campaigns.

2. Validate

We will focus to provide a basic validation capabilities for common campaign types, but we also plan to provide a way to specify some custom validation implementations, so everybody can extend our basic stack and satisfy his/her needs. This mechanism will be probably available through our own API which we plan to design and implement in next weeks.

3. Get results

Probably the most important and also most valuable part of this process is to use the proper way to visualize all the data we gathered. We want to make this visualization as valuable as we can, because we want to make possible executing further analyses and gain added value from the data itself. At the present there is only basic chart-like visualization of quiz campaign. We identificate this as our top priority feature.

1. Set up

To be able to provide feedback we need some details about the person behind computer. We will not push every community member to fill every detail, but we rather use some motivational techniques to gain these data. We will also try to have some special campaigns which won't have many constraints, to give also new members something they can actually do.

We realize that filling a full person profile is a time consuming process, so we already provide a way to connect user profile on social networks like facebook, google and linkedIn to greatly simplify this process. We also motivate members to make this connection because we realize that they actually perform some very valuable promotion for us.

2. Contribute

Members can see every campaign which is in alignment with their profile details. Besides the basics like gender, age, country, education we want also provide some selection based on member karma. Karma will be our special attribute. It's a way to deal with member which will try to make frauds and use the system in their favor. A specific way of how karma will be counted is also something we have to analyze and discuss further in next weeks.

3. Get reward

For the commercial version of this project we need to deal with rewarding. We already implemented basic credit system. In future we want to implement advanced possibilities, like credit bank, which will allow members to even change these credits for real money.

In education version we believe that members will contribute from their own will to help their colleagues, eventually for the ability to consequently set up their own campaigns.

V čom sme lepší ako ostatní - nieje to vysvetlene nízšie?

Our application is simple but contains advanced features.

Filtering ability

Aim Your feedback requests easily.

Custom campaigns

Templates are not enough? No problem.

Automation

Save Your time and avoid silly errors.

Live results

See what's going on, anytime.

Screenshoty alebo aj obrázky ?

The screenshot shows the PinTa.sk website interface. At the top, there is a navigation bar with the logo 'PinTa.sk' and links for 'Home', 'Profile', 'Badges', 'All tasks', and 'Log out'. Below the navigation bar, the main content area is titled 'Available tasks' and indicates 'Number of available tasks: 13'. A search bar is located to the right of this text. The tasks are listed in a table with the following columns: Name, Description, Category, Points for solving, Deadline, and Action. The table contains three rows of tasks. Below the table, there is a pagination control showing 'Previous', '1', '2' (highlighted), and 'Next'.

Name	Description	Category	Points for solving	Deadline	Action
Amazing people facebook page	Like our page on facebook	Facebook Likes	1	2014-02-18 10:59:00 UTC	Solve
Tag pictures (animals)	Find all animals on the picture	Picture Tagging	1	2014-02-18 11:39:00 UTC	Solve
Nutela survey	Answer questions about nutela	Survey	2	2014-02-18 11:41:00 UTC	Solve

© Pinta Group 2014

nieviem kolko a ktore tam chces.. nieco vyber

Something about AMT

Same kind of problem we trying to solve is aimed by an amazon project called Amazon Mechanical Turk. We identifies several weaknesses this we believe this system is suffering from.

- Not very usable UI/UX implemented for year 2014,
- lack of motivation and involvement in community,
- lack of social interaction and other competitive elements,

- not very good result presentation,
- and so on.

We also tries to set our features priorities based on these weaknesses.

Technology architecture

Programming platform : Ruby on Rails, JavaScript, CSS, SASS, HTML5

Version control system : Git

Issue tracking : Jira

Continuous integration : Jenkins

Virtual machine OS : Debian

Čo máme čo chceme

At the present we already have implemented :

- three types of campaigns,
- ability to connect user profile with facebook, google and linkedin social networks,
- result visualization for quiz campaign,
- basic motivational features like achievement system,
- presentational homepage Pinta.sk and basic responsive design.

What we plan to do :

- UI/UX usability evaluation and improvement,
- better and richer result presentation system,
- more gamification features and karma system,
- special education version of system, with specific campaigns,
- better and extendable validation process system,
- contacting eventual customers and PeWe members for their feedback.

Overenie na 1-2 vety

References

[1] Amazon Mechanical Turk. [Online] [Cited: February 18, 2014.]
[https://www.mturk.com/mturk/welcome.](https://www.mturk.com/mturk/welcome)

* Master degree study programme in field: Information Systems

Supervisor: Ing. Dušan Zeleník, Institute of Informatics and Software Engineering,
 Faculty of Informatics and Information Technologies STU in Bratislava

Definícia cieľov a očakávaných dopadov/prínosov

Primárnym cieľom projektu je vytvoriť webovú aplikáciu, kde sa budú stretávať obyčajní ľudia a budú poskytovať svoje názory. Taktiež budú môcť vykonávať jednoduché úlohy, ktorých výsledky budú užitočné pre zadávateľov úloh. Zadávatelia úloh v aplikácií budú hlavne študenti a marketingové spoločnosti. Takýmto spôsobom budú môcť získavať spätnú väzbu, ktorú by inou cestou nezískali alebo len veľmi obtiažne.

V aplikácií je navrhnutý systém odmeňovania, ktorý má motivovať ľudí k práci. Používatelia aplikácie budú rozdelení na dve skupiny. Jedna skupina bude hrať rolu zadávateľov úloh a druhá skupina bude na týchto úlohách pracovať. Používateľ v aplikácií môže zastávať naraz obe roly. Týmto sa môže zapájať na úlohách a zbierať odmeny, ktoré bude môcť neskôr premeniť na spätnú väzbu od používateľov v aplikácií.

Medzi základné úlohy patrí dotazník, kde musí používateľ vyplniť všetky otázky. Tieto odpovede sa môžu vyhodnocovať v aplikácií a po schválení používateľ dostane svoju odmenu, ktorá bola za úspešné dokončenie úlohy sľúbená.

Ľudia poskytujúci spätnú väzbu nemusia byť vždy kvalifikovaní odborníci. Často je teda požadovaná spätná väzba od obyčajných ľudí nezávisle od ich vzdelania, či spoločenskej prestíže.

Niekedy je však potrebné poznať názor užšej skupiny kvalifikovaných odborníkov v danej oblasti. Preto je potrebné úlohy zacieliť len na požadovaných používateľov. Aj to je jedným z dôležitých cieľov nášho produktu. Poskytnúť relevantnú spätnú väzbu, pomocou pokročilých funkcií a nastavení aplikácie, pre čo najkonkrétnejšiu špecifikáciu požiadaviek hociktorej skupiny.

Konkrétne ciele:

- Efektívnejšie získavanie spätnej väzby v problémových oblastiach
- Dosiachnutie spokojnosti u oboch skupín používateľov
- Cielené získavanie spätnej väzby
- Jednoduché rozhranie, ktoré môže používať aj osoba, ktorá nie je vzdelaná v informačných technológiach

Opis problémovej oblasti

Ponúkaný produkt sa zameriava na využitie internetu ako alternatívneho zdroja pre získanie spätnej väzby neinvazívnym spôsobom ponúkajúcim početné výhody pre zúčastnené skupiny.

Marketingové spoločnosti v súčasnosti využívajú rôzne prostriedky pre získanie spätnej väzby, ako napríklad papierové alebo online dotazníky, osobné rozhovory, telefonické kampane, postredníctvom internetu najčastejšie formou oneskorenej požiadavky o spätnú väzbu prostredníctvom emailu alebo formou nevyžiadanej pošty.

Nevýhodou týchto zdrojov je náhodný výber kandidátov, ktorý často vedie k násilnému až nepríjemnému spôsobu získavaniu spätnej väzby. Verejnosť takúto formu nevyžiadanej aktivity v súčasnosti ignoruje až odsudzuje. To má vo výsledku negatívny dopad na samotnú spoločnosť/produkt, pre ktorý je spätná väzba požadovaná.

Druhá cieľová skupina akademických pracovníkov a študentov má taktiež vlastné spôsoby získavania spätnej väzby. Študenti svoje práce overujú osobným dotazovaním kolegov o pomoc pri riešení, zvyčajne s príslubom spätnej pomoci. Akademickí pracovníci hlavne profesori, sa snažia získať spätnú väzbu od kolegov a študentov prípadne inej cieľovej skupiny, podobnou avšak nie tak násilnou formou ako v prípade marketingových spoločností. V oboch prípadoch sa často nenájde dostatok ochotných respondentov, a tí ktorí ochotu prejavia nemajú vždy dostatočnú motiváciu pre podanie kvalitnej spätnej väzby. Problémom však nie je len získať ľudí, ktorí by túto spätnú väzbu poskytli ale aj zabezpečiť prostredie, do ktorého by túto spätnú väzbu vložili a jednoducho vyhodnotili výsledky.

Výhodou prostriedkov využívaných oboma skupinami je hlavne priamy kontakt s opýtanými. Taktiež nie sú obmedzení pokrytím internetovej siete, a využitím alternatívnych kanálov sú schopní získať požadovanú službu.

Prehľad riešenia

Cieľom Produktu Pinta.sk je riešiť tieto problémy (opísane vyššie). Prioritným kanálom pre získanie spätnej väzby je internet, prostredníctvom webového rozhrania aplikácie.

System je založený primárne na preddefinovaných úlohach. Úloha je jednotka, ktorá má špecifickú formu vstupu a očakávaného výstupu. Každá forma úlohy môže mať naviazané tzv.validátory, ktoré môžu byť dodatočne špecifikované. Tie zabezpečujú evaluáciu odpovedí/výkonu pracantov.

V systéme vystupujú dve role - pracant a zadávateľ. Každý používateľ môže vystupovať v oboch roliach avšak v jednej úlohe len v jednej role. Hodnotenie rolí je oddelené. Čiže jeden používateľ môže mať napríklad v každej roli inú reputáciu.

Hlavným prínosom produktu je nenásilnosť získavania spätnej väzby. Systém ma za cieľ evokovať v potencionálnych respondentoch potrebu sa zúčasniť jednotlivých aktivít. Ako motivačné prostriedky využíva:

- **financie** - formou kreditu, ktorý je obojsmerne transformovateľný na peňažné jednotky. Kredit je výmennou hodnotou “obchodu” medzi zadávateľom a pracantom. Každá forma úlohy ktorú zadáva zadávateľ, je hodnotená určeným množstvom kreditu za jej splnenie.

Zadávateľ musí do systému vložiť kredit, ktorý využíva na tvorbu úloh. Tie môžu byť distribuované v konkrétnej cieľovej skupine zadefinovaním distribučných parametrov. Napríklad: *1 kredit za každú označovanú fotografiu. Parametre cieľovej skupiny: vek viac ako 20 rokov, študent, olasť zamerania fotografia.*

Na druhej strane pracant akceptovaním a splnením takejto úlohy dostane zarezervovanú získať položku. Tá mu bude pripočítaná ihneď no využiť ju bude môcť až po validácii jeho vstupu.

- **odmeňovanie** - na základe definovaných podmienok môže pracant (alternatívne aj zadávateľ) získať motivačné odmeny. Napríklad: *Kvalitným dokončením 50 úloh v oblasti značkovania fotografií získa 100 kreditov.*
- **reputácia** - vzťahuje sa na obe zúčastnené strany.

Pracant - každá splnená úloha prechádza vopred definovanou formou validácie. Výstupom validácie je boolovska hodnota. Na základe pomeru validných a nevalidných odpovedí je reputácia upravená nahor alebo nadol. Do výpočtu reputácie môžu byť zahrnuté aj ine parametre ako napríklad odbornosť výstupu, kvantita splnených úloh a iné.

Zadávateľ - po splení sady úloh môže byť každý zadávateľ pracantom ohodnotený v rôznych kategóriach. Napríklad: kvalita opisu úlohy, kvalita vstupov, kvalita dokumentácie a iné. Tieto parametre môžu byť špecifické pre jednotlivé typy úloh a formu validácie. Napríklad manuálna validácia s nadmerným počtom zamietnutých odpovedí môže byť tiež zahrnuté do výpočtu (automaticky/zásahom administrátora).

- **osobnostný rast** - motivačný prvok pre pracantov. Zameriava s na postupné sprístupňovanie nových/kvalitnejších typov úloh. Výstupom je “úroveň” jednotlivca, ktorá môže byť súčasťou parametrov zadávania úloh.

!!! Opis riešenia pre akademických pracovníkov - niečo na tému webová služba/ iná forma rozhrania. Sám neviem čo teda chceme implementovať. !!!

Pre skupinu pedagogických zamestnancov univerzít a študentov má získanie spätnej väzby prostredníctvom našej aplikácie tiež veľký význam. Študenti často potrebujú vo svojich diplomových prácach validovať svoje výsledky alebo zistiť a vyhodnotiť rôzne štatistiky týkajúce sa používateľnosti produktov alebo služieb rôznych firiem pôsobiacich v danej oblasti výskumu, názorov pri využívaní akademických nástrojov a prostriedkov výučby, preferencií techník výučby, atď. Aplikácia im umožní jednoduché získanie spätnej väzby a to aj za inú než finančnú odmenu, no ponuka odmeny musí byť dostatočne lákavá pre získanie záujmu poskytovateľov spätnej väzby.

Zhrnutie súčasného stavu riešenia projektu

1. Definované základné typy úloh
 - a. značkovanie obrázkov - cieľom je identifikácia objektov na obrázku, ktorú nie je možné vykonať automatizovane, a pritom je nevyhnutná pre spracovanie získaných údajov z obrázkov pre ďalšie účely,
 - b. jednoduchá otázka/odpoveď(dotazník) - jeden z najpoužívanejších typov úloh, cieľom je získať spätnú väzbu pomocou odpovedí na jednotlivé otázky používateľsky zvoleným spôsobom, hlavne však zodpovedaním otázky áno/nie, alebo písomnou opoveďou. Výsledkom z viacerých dotazníkov vznikne štatistika odpovedí, ktoré sa ďalej využijú na marketingové účely,
 - c. like na facebook/youtube/google plus
2. Zobrazenie výsledkov formou pie-chart - po poskytnutí spätnej väzby je zobrazená štatistika výsledkov vo forme pie-chart grafu, pre poskytovateľa zo zaujímavosti, marketingovými spoločnosťami využiteľné pre finančné ciele,
3. Systém osobného rastu a hodnotenia jednotlivca - každý zaregistrovaný používateľ dostáva odmeny za svoju pozitívnu aktivitu v systéme, je ohodnotený kreditmi za úspešné poskytovanie spätnej väzby, ďalšou odmenou je "virtuálna" odmena pomocou vyznamenaní (badges), ktoré umožňujú postup v hierarchii všetkých používateľov smerom nahor,
4. Autentifikácia, autorizácia, profil používateľa
5. Integrácia so sociálnymi skupinami
6. Moderný dizajn s využitím knižnice Bootstrap

Plán ďalšieho postupu

1. Implementovanie ďalších typov úloh

2. Export výsledkov pre zadávateľa - csv, zip - podľa typu úlohy
3. Definovanie webovej služby pre akademických pracovníkov
4. Integrovanie platobných modulov a nasadenie kreditného systému
5. Nasadenie betaverzie - evaluácia nápadu - získanie spätnej väzby

Ďalší vývoj závisí od reakcií potenciálnych návštevníkov

PINTA.SK - Online trh so spätnou väzbou

(TÍM 11)

Bc. Filip Bednárík, Bc. Róbert Černý, Bc. Marek Lenčoš, Bc. Miroslav Molnár, Bc. Patrik Štrba, Bc. Martin Toma, Bc. Miroslav Vojtuš

Vedúci tímu: Ing. Dušan Zeleník, PhD.

Slovenská technická univerzita, Bratislava
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava, Slovensko
team11.fiit@gmail.com

Abstrakt:

Ešte stále, aj napriek všetkým tým existujúcim technológiám je zložité získať rýchlu a zároveň kvalitnú spätnú väzbu od reálnych ľudí. Všetci vieme, že konštruktívna spätná väzba je kritický element v mnohých biznis oblastiach v súčasnosti. Spoločnosti najímajú agentúry, ktoré poskytujú ľudí a prostriedky nato, aby potom nejakým spôsobom získali spätnú väzbu od požadovanej cieľovej skupiny. Tu často nastáva problém. Nie všetky cieľové skupiny sa dajú jednoducho pokryť. Agentúry, ktoré majú v databázach veľké množstvo ľudí sú často príliš nákladné pre malé a stredné podniky. Okrem toho, tieto agentúry často využívajú už dnes zastaralé metódy ako marketing po telefóne, rôzne formy interview a dotazníkov ktorým chýba určitá forma hravosti a motivácie pre respondenta.

My sa snažíme rozšíriť tento problém spôsobom, že výraz spätná väzba pre nás neznamená len jednoduchú odpoveď na otázku ale tiež nejaká akcia ale malá úloha, ktorú môžeme splniť a priniesť tak hodnotu jej zadávateľovi.

Medzi naše hlavné ciele ktoré sa snažíme dosiahnuť :

- vybudovať online priestor, ktorý bude plniť úlohu trhu so spätnou väzbou,
- ponúknuť jednoduchú možnosť cielenia kampaní podľa rôznych atribútov (pohlavie, vek, miesto bydliska, vzdelanie, vedomosti, záujmy...)
- motivovať poskytovateľov spätnej väzby tým, že za svoju participáciu získajú odmenu a že celý proces poskytnutia tejto spätnej väzby bude jednoduchý a intuitívny,

- poskytnúť zadávateľom úloh možnosť vizualizácie výsledkov.

Ako nástroj na dosiahnutie týchto cieľov sme navrhli webovú aplikáciu slúžiacu ako trh so spätnou väzbou s názvom “Pinta.sk”. Táto aplikácia má poskytnúť bohatý ale zároveň používateľský prívetivý systém. Na dosiahnutie ďalších cieľov je potrebné vytvoriť filtračný systém, implementovať viaceré, dnes veľmi populárne “gamification” princípy, ale aj vytvoriť transakčný systém.

1. Prehľad riešenia

Pri vývoji aplikácie sme vychádzali so slovného opisu riešenia, ktoré poskytol náš vedúci a ktoré bolo dostupné na začiatku zimného semestra všetkým tímom. Prípadné modifikácie tohto pôvodného návrhu sme schvaľovali všetci, vrátane vedúceho tímu.

V aplikácii je navrhnutý systém odmeňovania, ktorý má motivovať ľudí k práci. Používatelia aplikácie budú rozdelení na dve skupiny. Jedna skupina bude hrať rolu zadávateľov úloh a druhá skupina bude na týchto úlohách pracovať. Používateľ v aplikácii môže zastávať naraz obe roly. Týmto sa môže zapájať na úlohách a zbierať odmeny, ktoré bude môcť neskôr premeniť na spätnú väzbu od používateľov v aplikácii.

Medzi základné úlohy patrí dotazník, kde musí používateľ vyplniť všetky otázky. Tieto odpovede sa môžu vyhodnocovať v aplikácii a po schválení používateľ dostane svoju odmenu, ktorá bola za úspešné dokončenie úlohy sľúbená.

Systém je založený primárne na preddefinovaných úlohách. Úloha je jednotka, ktorá má špecifickú formu vstupu a očakávaného výstupu. V systéme vystupujú dve role - pracant a zadávateľ. Každý používateľ môže vystupovať v oboch roliach avšak v jednej úlohe len v jednej role. Hodnotenie rolí je oddelené. Čiže jeden používateľ môže mať napríklad v každej roli inú reputáciu.

Hlavným prínosom produktu je nenásilnosť získavania spätnej väzby. Systém má za cieľ evokovať v potencionálnych respondentoch potrebu sa zúčastniť jednotlivých aktivít. Ako motivačné prostriedky využíva:

- **financie** - formou kreditu, ktorý bude v budúcnosti obojsmerne transformovateľný na peňažné jednotky. Kredit je výmennou hodnotou “obchodu” medzi zadávateľom a pracantom. Každá forma úlohy ktorú zadáva zadávateľ, je hodnotená určeným množstvom kreditu za jej splnenie.

Zadávatel' musí do systému vložiť kredit, ktorý využíva na tvorbu úloh. Tie môžu byť distribuované v konkrétnej cieľovej skupine zadefinovaním distribučných parametrov. Napríklad: *1 kredit za každú označovanú fotografiu. Parametre cieľovej skupiny: vek viac ako 20 rokov, študent, oblasť zamerania fotografia.*

Na druhej strane pracant akceptovaním a splnením takejto úlohy dostane rezervovanú získanú položku. Tá mu bude pripočítaná ihneď no využiť ju bude môcť až po validácii jeho vstupu.

- **špeciálne odmeňovanie** - na základe definovaných podmienok môže pracant (alternatívne aj zadávateľ) získať motivačné odmeny. Napríklad: *Kvalitným dokončením 50 úloh v oblasti značkovania fotografií získa 100 kreditov.*
- **reputácia (karma)** - vzťahuje sa na obe zúčastnené strany.

Pracant - každá splnená úloha prechádza vopred definovanou formou validácie. Výstupom validácie je hodnota 0 alebo 1. Na základe pomeru validných a nevalidných odpovedí je reputácia upravená nahor alebo nadol. Do výpočtu reputácie môžu byť zahrnuté aj iné parametre ako napríklad odbornosť výstupu, kvantita splnených úloh a iné.

Zadávatel' - po splnení sady úloh môže byť každý zadávateľ pracantom ohodnotený v rôznych kategóriách. Napríklad: kvalita opisu úlohy, kvalita vstupov, kvalita dokumentácie a iné. Tieto parametre môžu byť špecifické pre jednotlivé typy úloh a formu validácie. Napríklad manuálna validácia s nadmerným počtom zamietnutých odpovedí môže byť tiež zahrnutá do výpočtu (automaticky/zásahom administrátora).

- **osobnostný rast (odznaky)** - motivačný prvok pre pracantov. Zameriava sa na postupné sprístupňovanie nových, prestížnejších typov úloh. Výstupom je "úroveň" jednotlivca, ktorá môže byť súčasťou parametrov zadávania úloh.

Pre skupinu pedagogických zamestnancov univerzít a študentov má získanie spätnej väzby prostredníctvom našej aplikácie tiež veľký význam. Študenti často potrebujú vo svojich diplomových prácach validovať svoje výsledky alebo zistiť a vyhodnotiť rôzne štatistiky týkajúce sa používanosti produktov alebo služieb rôznych firiem pôsobiacich v danej oblasti výskumu, názorov pri využívaní akademických nástrojov a prostriedkov výučby, preferencií techník výučby, atď. Aplikácia im umožňuje jednoduché získanie spätnej väzby a to aj za inú než finančnú odmenu, no ponuka odmeny musí byť dostatočne lákavá pre získanie záujmu poskytovateľov spätnej väzby.

Táto cieľová skupina akademických pracovníkov a študentov má momentálne vlastné spôsoby získavania spätnej väzby. Študenti svoje práce overujú osobným dotazovaním kolegov o pomoc pri riešení, zvyčajne s príslubom spätnej pomoci. Akademickí pracovníci hlavne profesori, sa snažia získať spätnú väzbu od kolegov a študentov prípadne inej cieľovej skupiny, podobnou avšak nie tak násilnou formou ako v prípade marketingových spoločností. V oboch prípadoch sa často nenájde dostatok ochotných respondentov, a tí ktorí ochotu prejavia nemajú vždy dostatočnú motiváciu pre podanie kvalitnej spätnej väzby. Problémom však nie je len získať ľudí, ktorí by túto spätnú väzbu poskytli ale aj zabezpečiť prostredie, do ktorého by túto spätnú väzbu vložili a jednoducho vyhodnotili výsledky.

2. Realizácia

Zo všetkých navrhnutých funkcionalít opísaných v predchádzajúcej kapitole sme sa snažili postupne implementovať tie, ktoré prinesú pre potencionálnych používateľov najväčšiu pridanú hodnotu. Následnou analýzou navrhnutých funkcionalít a konzultáciou nápadu s viacerými potencionálnymi používateľmi sme sa rozhodli niektoré z nich zatiaľ neimplementovať a zamerali sme sa na iné oblasti.

2.1. Implementované typy úloh:

System v súčasnosti podporuje dva typy kampaní. Typ oznámenie (angl. notice) je určený na definovanie rôznych akcií či podujatí, kde sa uskutoční podstata spätnej väzby, prípadne pôjde o nejaký experiment. Tento typ úlohy sa môžeme využiť napríklad v prípade, že študent potrebuje zohnať ľudí na experiment. Pokiaľ sa jedná o experiment so špeciálnym hardvérom a podobne, je potrebné sa zúčastniť osobne. Okrem toho rôzne podujatia sú organizované na získanie popularity určitého produktu alebo služby.

Druhý typ, dotazník (angl. survey) podporuje tvorbu formulára, ktorý môže obsahovať rôzne typy otázok a odpovedí (áno/nie, viacero možností, likertové škály...) a ktorý je následne po jednotlivých otázkach zobrazený respondentom. Tento typ úlohy môže pokryť všetky kampane ktorých cieľom je získať štruktúrovanú spätnú väzbu. Či už sa jedná o hodnotenie, vlastný názor, výber z ponúkaných možností a podobne.

Okrem týchto dvoch vyladených kampaní je v systéme taktiež k dispozícii kampaň typu značkovanie obrázkov a vykonania facebook akcie "páči sa mi to" na zadaných URL adresách.

2.2. Filtre a profil používateľa

Po registrácii na portáli Pinta.sk si môže respondent spojiť svoj účet so sociálnymi sieťami ako LinkedIn, Facebook alebo Google a tým si automaticky vyplní veľkú časť profilu a zároveň je takto schopný ihneď nájsť dostupné kampane. Na druhej strane tým dávame možnosť zadávateľom kampaní špecifikovať cieľovú skupinu na pokročilých atribútoch ako vzdelanie, pracovná oblasť, znalosti a podobne.

Každá vytváraná kampaň teda môže obsahovať neobmedzené množstvo filtrov, ktoré udávajú presné obmedzenie na to, kto môže a kto nemôže participovať na riešení danej kampane. Momentálne podporujeme filtrovania na základe veku, krajiny, znalosti,

získané odznaky (viď kapitolu 2.4). Pridanie nových filtrov je z princípu jednoduché avšak pridáva zložitosť do zdrojového kódu.

2.3. Zobrazenie výsledkov:

Samozrejmosťou je aj možnosť zobrazenia aktuálnych výsledkov kampaní v prehľadných grafoch priamo cez naše webové rozhranie. Žiadateľ (firma, študent) teda nemusí ďalej spracovávať výsledky manuálne. Podporujeme zobrazenie výsledkov formou grafov (pie-chart, column-char, bar-chart). Taktiež aj výsledky je možné ďalej filtrovať. Toto dvoj-úrovňové filtrovanie nám poskytne možnosti ako už špecificky zameranú kampaň ďalej analyzovať. Okrem toho ponúkame možnosť exportovania výsledkov vo forme CSV, a tým dávame zadávateľovi možnosť využiť svoje vlastné metódy reprezentácie výsledkov, pokiaľ má o to záujem.

2.4. Gamification princípy:

V súčasnosti je veľmi populárny pojem “gamification”. V podstate ide o využitie prvkov, ktoré sa primárne vyskytovali iba v hrách a ich aplikácia v bežných aplikáciách. Pinta.sk implementuje niekoľko takýchto prvkov ako napríklad (vývoj používateľa, karma, rebríčky) a tým prináša do systému viac zábavy a motivácie.

System osobného rastu a hodnotenia jednotlivca - každý zaregistrovaný používateľ dostáva odmeny za svoju pozitívnu aktivitu v systéme, je ohodnotený kreditmi za úspešné poskytovanie spätnej väzby, ďalšou odmenou je “virtuálna” odmena pomocou vyznamenaní (odznaky), ktoré umožňujú postup v hierarchii všetkých používateľov smerom nahor, a môžu sa vyskytovať ako filtračný atribút pre niektoré kampane.

2.5. Používateľské rozhranie

Jedným z našich hlavných cieľov bolo vytvoriť jednoduché a použiteľné používateľské rozhranie. Používateľské rozhranie je prispôbené na mobilné aj desktopové zariadenia. Primárne sme sa však zamerali na desktop. Využili sme upravený bootstrap framework. Celkový design aplikácie však nie je dokončený, nakoľko by si to žiadalo skúsenejšieho grafika.

2.6. Pinta.sk API

Ku koncu letného semestra sme začali pracovať aj na rozhraní API pre zadávateľov úloh. API je realizované pomocou ako REST rozhranie a v súčasnosti poskytuje možnosť

získanie údajov o úlohách. Vytvorenie úlohy, získanie výsledkov a iné akcie zatiaľ nie sú implementované.

2.7 Overenie

Funkčnosť, zmyslupnosť a využiteľnosť nami implementovaných funkcionalít sme overili hlavne pri stretnutiach s potencionálnymi používateľmi. Čo sa týka aplikačnej funkčnosti, v systéme máme niekoľko testovacích scenárov vo frameworku cucumber.

3. Zhrnutie

Projekt v aktuálnom stave ponúka viacero možností na ďalší rozvoj. Projekt nebol doteraz smerovaný na konkrétny cieľ a preto sme ho vyvíjali univerzálne, preto je možné projekt upraviť pre konkrétnejšie potreby cieľových zákazníkov. V rámci projektu sme mali stretnutie s osobami v oblasti marketingu a startupov. Každý z týchto osôb dostal informácie o našom projekte a ponúkol nám smerovanie, prípadne čo by sme mali zlepšiť.

Najužitočnejšiu spätnú väzbu sme získali od Kataríny Chomovej, ktorá je výskumná pracovníčka na Ekonomickej Univerzite v Bratislave. Tá nám poradila zamerať sa na cieľovú oblasť študentov. Vzhľadom na to, že náš projekt umožňuje získanie odmeny za vykonanú prácu, tak je dobrým nástrojom pre študentov, ktorí majú veľa času a väčšinou málo financií a takto by si mohli pomôcť. Ako druhú stranu nám poradila sa zacieliť na malé a stredné firmy. Náš projekt poskytuje výmenu spätnej väzby čo už poskytujú aj iné produkty, avšak tieto produkty sú väčšinou neprístupné pre malé a stredné firmy. Preto by sme sa chceli ako zadávateľov úloh zamerať na túto cieľovú skupinu. Toto riešenie má veľa výhod a jedna z najdôležitejších, že by sa na tomto projekte dalo zarábať, teda ak by sa zvolil vhodný biznis model.

Ďalšie možné zameranie je pomoc pri získavaní spätnej väzby pri bakalárskych, diplomových či dizertačných prác. Takto študenti v rámci svojich prác často potrebujú získavať spätnú väzbu od relevantného publika. Náš nástroj by bol pre takúto činnosť výborný vzhľadom na to, že u nás v systéme sa dajú výsledky filtrovať a získavať len relevantné dáta. Táto cesta má svoje výhody, ale aj nevýhody. Vzhľadom na to, že by sa jednalo prevažne školský projekt, tak by sa za tým nedal postaviť vhodný biznis model ako pri prvom riešení.

PinTa.sk

online feedback marketplace

1. Focus

Targeting who You are interested in was never easier. With **PinTa.sk** You can set up an unlimited amount of filters and therefore get the right feedback, from the right people.

2. Validate

Need to validate results before publishing them? Set up Your custom validation process once, and it will be run before every feedback response. No more errors and wasted time.

3. Get results

Choose from many types of charts and views and watch them changing in real-time. Export Your results and present them where You need it, whenever You need it.

1. Set up Your profile

Register, fill Your profile and You will see available campaigns which best suits You. Use services like LinkedIn, Facebook or Google to save Your time and fill some profile details automatically.

2. Share feedback

Choose a campaign and start contributing. You can pause at any time and come back later or You can even skip the rest of a campaign and obtain only that amount of reward You deserve.

3. Get Your reward

As soon as Your contribution will pass validation processes, You will get Your reward. You can then use it to create Your own campaign, or easily exchange for products or real money.

Supervisor: Ing. Dušan Zeleník



www: www.pinta.sk
email: team1.fitt@gmail.com



Bc. Filip Bednářík, Bc. Róbert Černý, Bc. Marek Lencěš, Bc. Miroslav Molnár, Bc. Patrik Štrba, Bc. Martin Toma, Bc. Miroslav Vojtuš

C: Preberací protokol

Preberací protokol

Tímový projekt 2013/2014

Tím 11 - Pinta.sk

Predmet odovzdávania:

- Dokumentácia k riadeniu projektu - letný semester
- Dokumentácia k inžinierskemu dielu - letný semester

Pedagogický vedúci - Ing. Dušan Zeleník

Podpisom potvrdzuje prevzatie vyššie uvedených častí projektu a dokumentácie

dátum _____

podpis _____

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava 4

Pinta.sk
Digital Sweatshop
Dokumentácia k riadeniu

Študijný program: Informačné systémy, Softvérové inžinierstvo

Ročník: 1.

Predmet: Tímový projekt I.

Vedúci projektu: Ing. Dušan Zeleník

Členovia tímu: Bc. Filip Bednárík

Bc. Róbert Černý

Bc. Marek Lenčes

Bc. Miroslav Molnár

Bc. Patrik Štrba

Bc. Martin Toma

Bc. Miroslav Vojtuš

Ak. rok: 2013/2014

Obsah

1	Úvod	5
1.1	Účel dokumentu	5
1.2	O projekte	5
1.3	Štruktúra dokumentu	5
2	Ponuka	6
2.1	Členovia tímu	6
2.1.1	Bc. Filip Bednárík	6
2.1.2	Bc. Róbert Černý	6
2.1.3	Bc. Marek Lenčoš	6
2.1.4	Bc. Miroslav Molnár	6
2.1.5	Bc. Patrik Štrba	6
2.1.6	Bc. Martin Toma	7
2.1.7	Bc. Miroslav Vojtuš	7
2.2	Motivácia	7
2.3	Koncept riešenia	7
3	Záznamy zo stretnutí	8
3.1	Šablóna zápisnice	8
3.2	Zápis zo stretnutia č. 1	9
3.3	Zápis zo stretnutia č. 2	11
3.4	Zápis zo stretnutia č. 3	13
3.5	Zápis zo stretnutia č. 4	15
3.6	Zápis zo stretnutia č. 5	17
3.7	Zápis zo stretnutia č. 6	19
3.8	Zápis zo stretnutia č. 7	21
3.9	Zápis zo stretnutia č. 8	23
3.10	Zápis zo stretnutia č. 9	25
3.11	Zápis zo stretnutia č. 10	27
3.12	Zápis zo stretnutia č. 11	29
3.13	Zápisnica zo stretnutia č. 12	31
3.14	Zápisnica zo stretnutia č. 13	33
3.15	Zápisnica zo stretnutia č. 14	35
3.16	Zápisnica zo stretnutia č. 15	37
3.17	Zápisnica zo stretnutia č. 16	40
3.18	Zápisnica zo stretnutia č. 17	43
3.19	Zápisnica zo stretnutia č. 18	45
3.20	Zápisnica zo stretnutia č. 19	48
4	Úlohy členov tímu	50
4.1	Manažérske roly v tíme	50
4.2	Krátkodobé úlohy	50
4.3	Dlhodobé úlohy	51
4.4	Podiel členov na tvorbe dokumentácie k inžinierskemu dielu	52
4.4.1	Zimný semester	52
4.4.2	Letný semester	52
4.5	Podiel členov pri tvorbe diela	52
4.5.1	Zimný semester	52

4.5.2	Letný semester	52
5	Manažment rozvrhu a plánovania	53
5.1	Procesy	53
5.1.1	Tvorba a udržiavanie produktového backlogu	53
5.1.2	Tímové stretnutia a šprinty	53
5.1.3	Komunikácia počas šprintu	55
6	Manažment rizík	57
6.1	Zmeny v letnom semestri	62
7	Manažment monitorovania projektu	65
7.1	Navrhovaný prístup	65
7.2	Aplikácia prístupu	66
7.2.1	Dokumentovanie práce	66
7.2.2	Vyhodnotenie stavu projektu	66
7.3	Zmeny v letnom semestri	67
8	Manažment komunikácie a ľudských zdrojov	70
8.1	Metodika – Zaznamenávanie práce a sledovanie postupu	71
8.2	Zmeny v letnom semestri	73
9	Manažment dokumentácie	75
9.1	Roly účastníkov	75
9.2	Projektová dokumentácia	75
9.3	Zápisnice	76
9.4	Dokumentácia zdrojového kódu	77
9.5	Procesy	77
9.5.1	Dokumentovanie dokončenej funkcionality	77
9.5.2	Revízia dokumentácie	77
9.5.3	Dokumentovanie stretnutí	78
9.6	Zmeny v letnom semestri	78
10	Manažment kvality	79
10.1	Konvencie v jazyku Ruby a programovom rámci RoR	79
10.1.1	Všeobecné	79
10.1.2	Písanie názvov	79
10.1.3	Ruby konvencie	80
10.1.4	Migrácie	81
10.1.5	Model	81
10.1.6	Ovládač (ang. Controller)	81
10.1.7	Pohľad (ang. View)	82
10.2	Testovanie	82
10.2.1	Procesy	82
10.2.2	Konvencie pre testovanie	84
10.3	Spätná kontrola kvality	86
10.4	Zmeny v letnom semestri	86

11 Manažment podpory vývoja	88
11.1 Použitie vetiev (branch) v tímovom repozitári	88
11.2 Potvrdenie funkcionality do tímového repozitára	89
11.3 Vytvorenie novej verzie	89
11.4 Vytvorenie novej funkcionality	89
11.5 Oprava chyby na vývojovej verzii aplikácie	90
11.6 Oprava chyby na produkčnej verzii aplikácie	90
11.7 Spolupráca na spoločnej funkcionalite	90
11.8 Zmeny v letnom semestri	91
12 Plány	92
12.1 Zimný semester	92
12.2 Letný semester	92
12.2.1 Krátkodobý plán	92
12.2.2 Dlhodobý plán	93

1 Úvod

1.1 Účel dokumentu

Tento dokument je jednou z dvoch častí dokumentácie k vyvíjanému projektu, a obsahuje podrobnú dokumentáciu procesov riadenia pri vývoji projektu Pinta.sk vyvíjaného v rámci predmetu Tímový projekt na Fakulte informatiky a informačných technológií STU v Bratislave, so všeobecným názvom projektu Digital Sweatshop. Druhá časť dokumentácie s názvom Dokumentácia k inžinierskemu dielu, je dokumentáciou k vytváranému produktu.

1.2 O projekte

Projekt je akademickým dielom, a na jeho tvorbe sa podieľajú študenti spomenutej fakulty, konkrétne sedem členov tímu č. 11, ktorí svoj tím nazvali PintaGroup. Projekt Pinta.sk je originálnym a inovatívnym spôsobom ako si zarobiť peniaze za primeranú prácu aj z pohodlia domova, a zároveň ponúka možnosť rôznym firmám riešiť problémy a úlohy, pri riešení ktorých sa nezaobídu bez ľudskej pomoci. Ide o systém, presnejšie aplikáciu s webovým rozhraním pre zadávanie rôznych úloh za finančnú alebo inú odmenu. Vypracovanie týchto úloh za odmenu je podmienené zručnosťami a schopnosťami riešiteľa, a tiež jeho aktuálnou “pozíciou”, respektíve levelom na ktorom sa nachádza v rámci systému, podľa ktorého výšky sa mu sprístupňujú stále iné úlohy za vyššie odmeny.

Vytváranie projektu v rámci tímu je riadené Scrum metodikou, ide o agilný vývoj, teda o prinášanie novej funkcionality po každom šprinte, na ktoré je celkový vývoj aplikácie rozdelený. Tím sa počas šprintov pravidelne stretáva a konzultuje výsledky práce, svoj postup, a prípadné nedostatky v tíme.

1.3 Štruktúra dokumentu

Prvá kapitola dokumentu obsahuje ponuku vypracovaní tímom pre získanie projektu spomedzi ponúkaných projektov.

V druhej kapitole sa nachádzajú všetky doterajšie zápisnice zo stretnutí tímu podľa jednotlivých šprintov.

Tretia kapitola obsahuje zoznam krátkodobých a dlhodobých úloh členov v rámci tímu.

Ďalsie kapitoly obsahujú návody, postupy, a metodiky, ako používať rôzne nástroje, alebo ako postupovať pri vývoji produktu metodikou Scrum.

Predposledná kapitola obsahuje podrobný plán pre vypracovanie projektu na obidva semestre.

2 Ponuka

2.1 Členovia tímu

2.1.1 Bc. Filip Bednárík

Volám sa Filip, v tíme zastávam pozíciu hlavného softvérového architekta a popri tom sa venujem manažérstvu rizík. Okrem svojich programátorských povinností zastávam aj úlohu inšpektora pri prehliadkach zdrojového kódu. Vyštudoval som bakalára na FIITke a mám dlhoročné skúsenosti s prácou vo firmách na tvorbe rozličných webov. Vo voľnom čase sa venujem kamarátom, gitare a sem tam si niečo nakódim. Motto: *„Keď sa všetci chytíme vesla, tak hádam niekam doveslujeme“*.

2.1.2 Bc. Róbert Černý

Volám sa Robo, mám 22 rokov a v tíme zastávam úlohu manažéra plánovania a organizácie. Dozerám na správnosť priebehu stretnutí, usmerňujem komunikáciu a dozerám na dodržiavanie dohodnutých termínov pre plnenie potrebných úloh. Náš tím je agilný a moja metodika SCRUM-u je prispôbená našim podmienkam. Som taktiež absolventom bakalárskeho štúdia na FIIT STU v obore informatika. Mám praktické skúsenosti s vývojom webových aplikácií a som otvorený novým možnostiam a prístupom. Voľný čas trávim cestovaním, športovaním, počúvaním hudby a hľadaním zmyslu života. Motto: *„Shut up and squat“*.

2.1.3 Bc. Marek Lenčes

Moje meno je Marek a v tíme PintaGroup zohrávam úlohu manažéra dokumentácie. Kontrolujem jej stav, zabezpečujem jej konzistentnosť, bezchybnosť a som zodpovedný za jej finálne prezentovanie. Štúdium som absolvoval na FIIT STU a počas štúdia som sa priučil mnohým veciam. Medzi moje záľuby patrí navrhovanie 3D dizajnov interiérov, ale aj exteriérov budov, tiež modelovanie 3D grafiky vo všeobecnosti, oprava notebookov a kšeftovanie s elektronikou, príležitostne aj indoorové športy, napríklad biliard, bowling, či stolný tenis. V živote sa riadim heslom: *„Keď druhému jamu kopeš, sprav ju aspoň poriadne hlbokú.“*

2.1.4 Bc. Miroslav Molnár

Moje meno je Miroslav, v tíme zastávam úlohu monitorovania projektu. Baví ma kreatívna činnosť a často sa podieľam na hľadaní nových funkcionalít v systéme. Som absolvent FIT VUT a vo svojom voľnom čase sa venujem športu a záhradkárstvu. Motto: *„Život je krutý, život je pes a pes je najlepší priateľ človeka“*.

2.1.5 Bc. Patrik Štrba

Volám sa Patrik a mám 22 rokov. V tíme sa starám o spoločný tímový repozitár. Taktiež sa starám o proces nasadenia projektu na produkčný server. Som absolventom bakalárskeho štúdia na FIIT STU. Aktívne sa venujem start-up projektom a zaujímam sa o počítačovú bezpečnosť. Mám dlhoročné skúsenosti s vývojom webových aplikácií. Vo voľnom čase rád počúvam hudbu, učím nové zaujímavé veci alebo si rád pozriem dobrý film.

2.1.6 Bc. Martin Toma

Volám sa Martin mám 22 rokov a v tíme sa starám o to, aby sme spolu komunikovali čo možno najefektívnejšie. Zabídam však skoro do každej oblasti nakoľko plním aj funkciu vedúceho projektu a chcem mať vo všetkom prehľad. Som absolventom bakalárskeho štúdia na FIIT STU, rád sa venujem najmä soft-skills oblasti a zaujímam sa o start-up scénu a agilné metódy. V súkromí si rád zájdem zašportovať, mám rád dobrú hudbu, chutné jedlo a pekné ženy. Mojm motto je: „*Everything is possible with the right attitude!*“

2.1.7 Bc. Miroslav Vojtuš

Volám sa Miroslav, mám 23 rokov a v tíme sa starám o kvalitu produktu. Okrem tejto manažérskej role som pracant-vývojár alebo inak aj kóder. Vo voľnom čase sa rád venujem športom, hlavne streľbe z luku, alebo šermu. Rád pozerám fantastické filmy s nádychom histórie. Okrem toho ešte rád splavujem divoké rieky, a objavujem dobrodružstvá pri cestovaní svetom. Moje motto je: „*Just do it !!!*“

2.2 Motivácia

Základnou náplňou projektu je prostredníctvom webovej aplikácie poskytnúť istý druh rozhrania medzi dvoma skupinami ľudí, z ktorých jedna skupina potrebuje získať určitú formu spätnej väzby a druhá skupina ľudí si chce privyrobiť a v neposlednom rade aj trochu pomôcť. Využitie tohto rozhrania je tak veľké, že môžeme pokryť až veľmi širokú oblasť. Od akademickej pôdy pri rôznych formách úloh, od dotazníkov cez jednoduché úlohy až po zložitejšie overovanie hypotéz vrámci bakalárskych, diplomových či dizertačných prác. Ďalšou veľkou oblasťou je marketing. Nahradenie zastaraných marketingových techník a aktivít (telefónny marketing) za priame oslovenie cieľovej skupiny prostredníctvom nášho rozhrania. Možnosti využitia tohto rozhrania sú teda naozaj skoro neobmedzené.

2.3 Koncept riešenia

Základným cieľom projektu je vytvoriť aktívne používanú webovú či mobilnú aplikáciu, ktorej hlavným cieľom je sprostredkovať mikroprácu. Chceme dosiahnuť spokojnosť na oboch stranách (zadávateľ aj pracant). Zadávateľovi chceme ponúknuť možnosť získať kvalitnú spätnú väzbu tým že efektívne nájdeme správnych kandidátov na danú úlohu. Pracantovi chceme dať možnosť si privyrobiť a motivovať ho systémom reputácie a odmeňovania. Projekt má potenciál poskytnúť rozhranie využiteľné v rôznych oblastiach. Hlavným cieľom tohto projektu je však pokrytie akademickej a marketingovej oblasti. V budúcnosti sa však určite zameriame aj na iné oblasti.

3 Záznamy zo stretnutí

3.1 Šablóna zápisnice

Zápisnica č. 4. / (číslo zápisnice)

Téma Plánovanie 3. šprintu	Dátum 21.10.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Marek Lenčes	

Priebeh stretnutia

(Obsahuje informácie o priebehu stretnutia, najprv vyhodnotenie šprintu, čo sa spravilo, čo sa nestihlo, ďalej rozoberané veci na stretnutí, a nakoniec naplánovanie úloh do ďalšieho šprintu)

- Vyhodnotenie druhého šprintu
 - vyhodnotenie splnenia zadaných úloh
- Diskusia
 - čo robíme dobre, čo robíme zle, návrhy, pripomienky
- Naplánovanie ďalšieho šprintu
 - výber úloh z produktového backlogu a pridelenie členom tímu

Rozdelenie úloh

(Rozdelenie úloh z backlogu a pridelenie členom tímu bude usporiadane zobrazené v tabuľke, s krátkym popisom úlohy, menom člena, ktorému bola úloha pridelená, dátum kedy bola zadaná, kedy bude približne ukončená, a v akom stave sa momentálne nachádza. Príklad:)

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokladané ukončenie	Stav
3.1	Definovanie prvej user story a rozdelenie na podúlohy	Filip	14.10.2013	20.10.2013	Riešená
3.2	Analýza podporných nástrojov pre implementáciu v Ruby	Robo	14.10.2013	20.10.2013	Splnená

3.2 Zázpis zo stretnutia č. 1

Zázpisnica č. 1.

Téma	Dátum	Čas	Miesto
Úvodné stretnutie	30.9.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Molnár		

Priebeh stretnutia

- Podrobné rozobratie témy projektu
 - Oboznámenie sa s problémovým prostredím
 - Popis štruktúry aplikácie
 - Popis podobných aplikácií a ich nedostatky
 - Možnosti využitia aplikácie v našom okolí
 - Definovanie potencionálnych používateľov a klientov

- Diskusia o projekte
 - Výber podporných nástrojov pre realizáciu projektu
 - Identifikácia prvých základných features aplikácie
 - Dohoda na dĺžke trvania šprintov
 - Dohoda na čase a mieste stretnutí
 - Rozdelenie jednotlivých rolí v tíme
 - Prediskutovanie vybraných podporných nástrojov - Rails, Github, Jira
 - Výber nástroja pre code review
 - Diskusia o problémoch pri použití code review nástrojov

- Naplánovanie akcií do prvého šprintu
 - Analýza existujúcich riešení
 - Identifikácia čo najväčšieho množstva features aplikácie
 - Vytvorenie prezentačnej stránky tímu
 - Dohodnutie sa na použití podporných nástrojov a ich inštalácia
 - Vytvorenie šablóny zázpisnice
 - Vytvorenie Lean Canvas
 - Naštudovať základné konvencie jazyka Ruby a frameworku Rails
 - Dohodnúť sa na verzii týchto nástrojov, ktorá sa bude používať
 - Naštudovať konvencie používania verziovacieho systému GitHub
 - Konfigurácia podporných nástrojov pre naše potreby - Jira, Jenkins

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
0.1	Definovanie dôležitých vlastností produktu, vymenovanie features, doporučiť technológie použité v projekte.	Filip, Robo, Martin, Patrik, Miro V, Miro M, Marek	30.9.2013	13.10.2013	Riešená
0.2	Inštalácia a konfigurácia podporných nástrojov	Martin, Patrik	30.9.2013	13.10.2013	Riešená
0.3	Vytvorenie prezentačnej stránky	Miro M., Marek, Robo	30.9.2013	13.10.2013	Riešená
0.4	Vytvorenie šablóny zápisnice	Marek	30.9.2013	13.10.2013	Riešená
0.5	Analýza existujúcich riešení	Marek	30.9.2013	13.10.2013	Riešená
0.6	Vytvorenie Lean Canvas	Spoločná	30.9.2013	13.10.2013	Riešená
0.7	Vymyslenie názvu aplikácie a tímu	Spoločná	30.9.2013	13.10.2013	Riešená
0.8	Štúdium Ruby on Rails, Github	Spoločná	30.9.2013	13.10.2013	Riešená

3.3 Zápis zo stretnutia č. 2

Zápisnica č. 2.

Téma	Dátum	Čas	Miesto
Plánovanie 1. šprintu	14.10.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčes		

Priebeh stretnutia

- Vyhodnotenie úvodnej prípravy
 - Vyhodnotenie splnenia zadaných úloh členmi tímu
 - Diskusia o problémoch pri úlohách
 - Prehodnotenie trvania dĺžky šprintov - 1-týždňové
 - Zhodnotenie vhodnosti vytvorenej prezentačnej stránky

- Diskusia o projekte
 - Prediskutovanie vybraných podporných nástrojov - Rails, Github, Jira
 - Diskusia o návrhoch názvu aplikácie - zhodli sme sa na Taskerix
 - Prvotné návrhy štruktúry a dizajnu aplikácie
 - Návrhy pre zlepšenie a rozšírenie vytvárania úloh používateľom
 - Diskusia o potrebe definovania produktového backlogu aplikácie

- Naplánovanie prvého šprintu
 - Potreba definovať úlohy v rámci prvého user story
 - Návrh rozhrania aplikácie
 - Vytvoriť ďalšie features pri vytváraní úlohy
 - Vytvoriť formuláre pre registráciu a prihlásenie
 - Vyriešiť autorizáciu a autentifikáciu pri prihlasovaní
 - Vytvorenie uploadera na obrázky
 - Formulár na zadanie podrobností vytváranej úlohy, štruktúra úlohy, nápoveda k úlohe, definovanie kategórií úloh

Stav úloh z úvodného stretnutia

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
0.1	Definovanie dôležitých vlastností produktu, vymenovanie features, doporučiť technológie použité v projekte.	Filip, Robo, Martin, Patrik, Miro V, Miro M, Marek	30.9.2013	13.10.2013	Splnená
0.2	Inštalácia a konfigurácia podporných nástrojov	Martin, Patrik	30.9.2013	13.10.2013	Splnená
0.3	Vytvorenie prezentačnej stránky	Miro M., Marek, Robo	30.9.2013	13.10.2013	Splnená
0.4	Vytvorenie šablóny zápisnice	Marek	30.9.2013	13.10.2013	Splnená
0.5	Analýza existujúcich riešení	Marek	30.9.2013	13.10.2013	Splnená
0.6	Vytvorenie Lean Canvas	Spoločná	30.9.2013	13.10.2013	Splnená
0.7	Vymyslenie názvu aplikácie a tímu	Spoločná	30.9.2013	13.10.2013	Splnená
0.8	Štúdium Ruby on Rails, Github	Spoločná	30.9.2013	13.10.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
1.1	Definovanie úloh prvej User Story	Filip	14.10.2013	20.10.2013	Riešená
1.2	Definovanie úloh prvej User Story + návrh rozhrania aplikácie	Robo	14.10.2013	20.10.2013	Riešená
1.3	User model, autorizácia, autentifikácia používateľa	Martin, Patrik	14.10.2013	20.10.2013	Riešená
1.4	Uploader na obrázky a súbory	Miro V.	14.10.2013	20.10.2013	Riešená
1.5	Formulár na zadanie podrobností vytváratej úlohy, štruktúra úlohy, nápoveda k úlohe, definovanie kategórií úloh	Marek, Miro M.	14.10.2013	20.10.2013	Riešená
1.6	Vytvorenie rozhrania pre registráciu a prihlásenie	Martin, Patrik	14.10.2013	20.10.2013	Riešená
1.7	Vytvorenie rozhrania pre vytvorenie úlohy používateľom	Robo, Filip	14.10.2013	20.10.2013	Riešená

3.4 Zázpis zo stretnutia č. 3

Zázpisnica č. 3.

Téma	Dátum	Čas	Miesto
Plánovanie 2. šprintu	21.10.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Vojtuš		

Priebeh stretnutia

- Vyhodnotenie prvého šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie low fidelity návrhu dizajnu aplikácie
 - Ukážka funkcionality riešených features

- Diskusia o projekte
 - Identifikácia jednotlivých tém pre metodiky
 - Rozdelenie metodík medzi členov tímu
 - Metodika Code Review - Bc. Filip Bednárík
 - Metodika Scrum fungovania - Bc. Róbert Černý
 - Metodika Jira - Bc. Martin Toma
 - Metodika dokumentácie - Bc. Marek Lenčes
 - Metodika Git verziovacieho systému - Bc. Patrik Štrba
 - Metodika podpory vývoja - Bc. Miroslav Molnár
 - Metodika menných konvencií pre Rails - Bc. Miroslav Vojtuš
 - Diskusia o potrebe definovania produktového backlogu

- Naplánovanie ďalšieho šprintu
 - Potreba zjednotiť všetky vetvy na GitHub-e
 - Ďalšie hlbšie štúdium Rails
 - Dokončenie detailov jednotlivých funkcionalít

Stav úloh z prvého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
1.1	Definovanie úloh prvej User Story	Filip	14.10.2013	20.10.2013	Splnená
1.2	Definovanie úloh prvej User Story + návrh rozhrania aplikácie	Robo	14.10.2013	20.10.2013	Splnená
1.3	User model, autorizácia, autentifikácia používateľa	Martin, Patrik	14.10.2013	20.10.2013	Splnená
1.4	Uploader na obrázky a súbory	Miro V.	14.10.2013	20.10.2013	Splnená
1.5	Formulár na zadanie podrobností vytváratej úlohy, štruktúra úlohy, nápoveda k úlohe, definovanie kategórií úloh	Marek, Miro M.	14.10.2013	20.10.2013	Splnená
1.6	Vytvorenie rozhrania pre registráciu a prihlásenie	Martin, Patrik	14.10.2013	20.10.2013	Splnená
1.7	Vytvorenie rozhrania pre vytvorenie úlohy používateľom	Robo, Filip	14.10.2013	20.10.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
2.1	Štúdium Rails	Spoločná	21.10.2013	27.10.2013	Riešená
2.2	Zjednotenie všetkých funkcionálit do jednej vetvy na GitHubu	Spoločná	21.10.2013	27.10.2013	Riešená
2.3	Dokončenie detailov na každej vetve	Spoločná	21.10.2013	27.10.2013	Riešená

3.5 Zázpis zo stretnutia č. 4

Zázpisnica č. 4.

Téma	Dátum	Čas	Miesto
Plánovanie 3. šprintu	28.10.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčes		

Priebeh stretnutia

- Vyhodnotenie druhého šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce

- Diskusia o projekte
 - Doplnenie produktového backlogu
 - Diskusia o zmene git klienta z GitHub-u na Bitbucket
 - Dekompozícia User Stories na úlohy
 - Dohoda na potrebe pravidiel - vypracovaním metodík

- Naplánovanie ďalšieho šprintu
 - Zjednotiť produktový backlog
 - Treba napísať všetky definované metodiky
 - Je potrebné vytvoriť tímový kalendár kvôli stretnutiam
 - Vyriešiť spustenie produkčného serveru pre hostovanie aplikácie
 - Treba zrevidovať všetky doterajšie zázpisnice

Stav úloh z druhého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
2.1	Štúdium Rails	Spoločná	21.10.2013	27.10.2013	Prebieha
2.2	Zjednotenie všetkých funkcionalít do jednej vetvy na GitHubu	Spoločná	21.10.2013	27.10.2013	Splnená
2.3	Dokončenie detailov na každej vetve	Spoločná	21.10.2013	27.10.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
3.1	Usporiadanie produktového backlogu	Martin, Patrik	28.10.2013	03.11.2013	Riešená
3.2	Spísanie metodík	Spoločná	28.10.2013	03.11.2013	Riešená
3.3	Vytvorenie tímového kalendáru	Robo	28.10.2013	03.11.2013	Riešená
3.4	Spustenie produkčného servera	Patrik	28.10.2013	03.11.2013	Riešená
3.5	Revízia doterajších zápisníc	Marek	28.10.2013	03.11.2013	Riešená
3.6	Vytvorenie formulára pre vyplnenie profilu	Miro M.	28.10.2013	03.11.2013	Riešená

3.6 Zázpis zo stretnutia č. 5

Zázpisnica č. 5.

Téma	Dátum	Čas	Miesto
Plánovanie 4. šprintu	04.11.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Róbert Černý		

Priebeh stretnutia

- Vyhodnotenie tretieho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Dokončenie produktového backlogu
 - Migrácia na Bitbucket kvôli Jire
 - Rozpracované metodiky
- Diskusia o projekte
 - Riešenie názvu domény
 - Revízia integrácie projektu
 - Demo na tagovanie obrázkov
 - Testovanie Geo Lokácie
 - Prehodnotenie spôsobu komunikácie členov
 - Vytváranie sprint backlogu
- Naplánovanie ďalšieho šprintu
 - Pridanie novej funkcionality do aplikácie
 - Výpočet času, ktorý strávil pracovník plnením úloh
 - Vytvorenie filtrov pre pracovníkov
 - Vytvorenie kreditového systému pre pracovníkov
 - UI webovej aplikácie
 - Vytvorenie profilovej stránky pracovníka
 - registrácia pracovníkov cez sociálne siete
 - Migrácia na Bitbucket
 - Vytvorenie achievement systemu
 - Pridanie referral odkazov pre pracovníkov

Stav úloh z tretieho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
3.1	Usporiadanie produktového backlogu	Martin, Patrik	28.10.2013	03.11.2013	Splnená
3.2	Spísanie metodík	Spoločná	28.10.2013	03.11.2013	Riešená
3.3	Vytvorenie tímového kalendáru	Robo	28.10.2013	03.11.2013	Splnená
3.4	Spustenie produkčného servera	Patrik	28.10.2013	03.11.2013	Splnená
3.5	Revízia doterajších zápisníc	Marek	28.10.2013	03.11.2013	Splnená
3.6	Vytvorenie formulára pre vyplnenie profilu	Miro M.	28.10.2013	03.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
4.1	Vytvorenie filtrov a kreditového systému	Filip	04.11.2013	10.11.2013	Riešená
4.2	Vytvorenie UI dizajnu aplikácie	Robo	04.11.2013	10.11.2013	Riešená
4.3	Vytvorenie referral odkazov	Marek	04.11.2013	10.11.2013	Riešená
4.4	Vytvorenie profilovej stránky	Miro M.	04.11.2013	10.11.2013	Riešená
4.5	Registrácie cez soc.siete + Bitbucket	Patrik	04.11.2013	10.11.2013	Riešená
4.6	Registrácie cez soc.siete + vytvorenie subtaskov	Martin	04.11.2013	10.11.2013	Riešená
4.7	Vytvorenie achievement systému	Miro V.	04.11.2013	10.11.2013	Riešená

3.7 Zázpis zo stretnutia č. 6

Zázpisnica č. 6.

Téma	Dátum	Čas	Miesto
Plánovanie 5. šprintu	11.11.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Filip Bednárík		

Priebeh stretnutia

- Vyhodnotenie štvrtého šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce

- Diskusia o projekte
 - Riešenie achievement badges
 - Identifikovanie atribútov profilu
 - Skupiny používateľov

- Naplánovanie ďalšieho šprintu
 - Integrácia s LinkedIn a Facebook - údaje do profilu
 - Dokončenie dokumentácie
 - Vymyslieť achievementy
 - Integrovanie s ďalšími sociálnymi sieťami - Google Plus, Twitter

Stav úloh zo štvrtého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
4.1	Vytvorenie filtrov a kreditového systému	Filip	04.11.2013	10.11.2013	Splnená
4.2	Vytvorenie UI dizajnu aplikácie	Robo	04.11.2013	10.11.2013	Splnená
4.3	Vytvorenie referral odkazov	Marek	04.11.2013	10.11.2013	Riešená
4.4	Vytvorenie profilovej stránky	Miro M.	04.11.2013	10.11.2013	Splnená
4.5	Registrácie cez soc.siete + Bitbucket	Patrik	04.11.2013	10.11.2013	Splnená
4.6	Registrácie cez soc.siete + vytvorenie subtaskov	Martin	04.11.2013	10.11.2013	Splnená
4.7	Vytvorenie achievment systému	Miro V.	04.11.2013	10.11.2013	Riešená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
5.1	Vytváranie dokumentácie	Spoločná	11.11.2013	17.11.2013	Riešená
5.2	Vytvorenie UI dizajnu aplikácie	Robo	11.11.2013	17.11.2013	Riešená
5.3	Integrácia s Google účtom + referral odkazy dokončiť	Marek	11.11.2013	17.11.2013	Riešená
5.4	Integrácia s Twitter účtom	Miro M.	11.11.2013	17.11.2013	Riešená
5.5	Facebook lajky a komentáre, zmena profilu používateľa	Patrik, Martin	11.11.2013	17.11.2013	Riešená
5.6	Vymyslieť, implementovať achievmenty - merit gem	Miro V.	11.11.2013	17.11.2013	Riešená

3.8 Zápis zo stretnutia č. 7

Zápisnica č. 7.

Téma	Dátum	Čas	Miesto
Plánovanie 6. šprintu	18.11.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Patrik Štrba		

Priebeh stretnutia

- Vyhodnotenie piateho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Prezentácia metodík
 - Diskusia a pripomienky k metodikám
 - Potreba rebrandingu pre nasadenie produkčnej verzie
 - Diskusia k dokumentácii, a častiam potrebným pre vypracovanie
- Naplánovanie ďalšieho šprintu
 - Potreba zmeniť vytváranie úlohy na viackrokové formuláre
 - Spraviť rebranding produktu - zmena názvu, loga
 - Prerobenie tímového webu popri rebrandingu
 - Zmeniť organizáciu údajov v profile
 - Možnosť komentovania a lajkovania Youtube videa
 - Použitie ďalších informácií z LinkedIn profilu
 - Lokalizácia aplikácie do viacerých jazykov

Stav úloh z piateho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
5.1	Vytváranie dokumentácie	Spoločná	11.11.2013	17.11.2013	Riešená
5.2	Vytvorenie UI dizajnu aplikácie	Robo	11.11.2013	17.11.2013	Splnená
5.3	Integrácia s Google účtom + referral odkazy dokončiť	Marek	11.11.2013	17.11.2013	Splnená
5.4	Integrácia s Twitter účtom	Miro M.	11.11.2013	17.11.2013	Riešená
5.5	Facebook lajky a komentáre, zmena profilu používateľa	Patrik, Martin	11.11.2013	17.11.2013	Splnená
5.6	Vymyslieť, implementovať achievmenty - merit gem	Miro V.	11.11.2013	17.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
6.1	Vytvorenie viackrokových formulárov pre zadanie úlohy	Filip	18.11.2013	24.11.2013	Riešená
6.2	Home, práva na stránky	Robo	18.11.2013	25.11.2013	Riešená
6.3	Spojenie dokumentácie	Marek	18.11.2013	24.11.2013	Riešená
6.4	Youtube videá - like, komentár, možnosť pozretia celého videa	Miro M.	18.11.2013	24.11.2013	Riešená
6.5	Údaje z LinkedIn - vzdelanie, práca, zručnosti	Patrik	18.11.2013	24.11.2013	Riešená
6.6	Rebranding - web tímu, dizajn	Marek, Martin	18.11.2013	24.11.2013	Riešená
6.7	Zmena organizácie profilu, dizajn profilu	Miro V.	18.11.2013	24.11.2013	Riešená

3.9 Zázpis zo stretnutia č. 8

Zázpisnica č. 8.

Téma	Dátum	Čas	Miesto
Plánovanie 7. šprintu	25.11.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Martin Toma		

Priebeh stretnutia

- Vyhodnotenie šiesteho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti

- Diskusia o projekte
 - Diskusia k podstatným častiam aplikácie
 - Ochrana osobných údajov používateľov v systéme
 - Definovanie nových typov úloh v systéme, a ich reálne použite
 - Iteratívna komunikácia zadávateľa s pracovníkom o úlohe
 - Zameranie sa na vytvorenie funkčného scenára
 - Vytvorenie úloh pre použitie v primárnom prostredí - FIIT STU
 - Obmedzenie vytváraných úloh na dotazníky
 - Diskusia k dokumentácii, čo chýba, čo treba dorobiť
 - Rozdelenie používateľov do skupín v rámci systému

- Naplánovanie ďalšieho šprintu
 - Vytvorenie prezentačných materiálov aplikácie - plagát
 - Vytvorenie landing page pre aplikáciu
 - Lokalizácia do viacerých jazykov

Stav úloh zo šiesteho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
6.1	Vytvorenie viackrokových formulárov pre zadanie úlohy	Filip	18.11.2013	24.11.2013	Splnená
6.2	Home, práva na stránky	Robo	18.11.2013	25.11.2013	Splnená
6.3	Spojenie dokumentácie	Marek	18.11.2013	24.11.2013	Splnená
6.4	Youtube videá - like, komentár, možnosť pozretia celého videa	Miro M.	18.11.2013	24.11.2013	Riešená
6.5	Údaje z LinkedIn - vzdelanie, práca, zručnosti	Patrik	18.11.2013	24.11.2013	Splnená
6.6	Rebranding - web tímu, dizajn	Marek, Martin	18.11.2013	24.11.2013	Splnená
6.7	Zmena organizácie profilu, dizajn profilu	Miro V.	18.11.2013	24.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
7.1	Logovanie cez routa, id úloh/id pracovník tag	Filip	25.11.2013	01.12.2013	Riešená
7.2	Vytvorenie high chart grafov pre zobrazenie výsledkov úlohy	Robo	25.11.2013	01.12.2013	Riešená
7.3	Vytvorenie landing page obrázkov a tímového posteru	Marek	25.11.2013	01.12.2013	Riešená
7.4	Preklad lokalizačných súborov	Miro M.	25.11.2013	01.12.2013	Riešená
7.5	Vytvorenie landing page	Patrik, Martin	25.11.2013	01.12.2013	Riešená
7.6	Odmeny za vyplnenie profilu, vytvorenie reorganizovaného profilu	Miro V.	25.11.2013	01.12.2013	Riešená

3.10 Zázpis zo stretnutia č. 9

Zázpisnica č. 9.

Téma	Dátum	Čas	Miesto
Plánovanie 8. šprintu	02.12.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Molnár		

Priebeh stretnutia

- Vyhodnotenie siedmeho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Diskusia k podstatným častiam aplikácie
 - Potrebné vytvoriť logovanie
 - Identifikovanie ďalších typov úloh
 - Špeciálne domény aplikácie, namiesto viacerých typov úloh
 - Refactoring kódu
 - Vytvorenie Pie grafov
- Naplánovanie ďalšieho šprintu
 - Tvorba dokumentácie, dokumentovanie dorobených funkcionalít
 - Kontrola dokumentácie na chyby, nekonzistentnosť
 - Vytvorenie testov pre funkcionality
 - Cucumber test framework

Stav úloh zo siedmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
7.1	Logovanie cez routa, id úloh/id pracovník tag	Filip	25.11.2013	01.12.2013	Splnená
7.2	Vytvorenie high chart grafov pre zobrazenie výsledkov úlohy	Robo	25.11.2013	01.12.2013	Splnená
7.3	Vytvorenie landing page obrázkov a tímového posteru	Marek	25.11.2013	01.12.2013	Splnená
7.4	Preklad lokalizačných súborov	Miro M.	25.11.2013	01.12.2013	Splnená
7.5	Vytvorenie landing page	Patrik, Martin	25.11.2013	01.12.2013	Splnená
7.6	Odmeny za vyplnenie profilu, vytvorenie reorganizovaného profilu	Miro V.	25.11.2013	01.12.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
8.1	Refactoring kódu	Filip	02.11.2013	08.11.2013	Riešená
8.2	Vytvorenie ďalších high chartov	Robo	02.11.2013	08.11.2013	Riešená
8.3	Tvorba dokumentácie, kontrola chýb, nekonzistencií, definovanie obsahu	Marek	02.11.2013	08.11.2013	Riešená
8.4	Upravenie zobrazenie badges, vymyslieť nové badges	Miro M.	02.11.2013	08.11.2013	Riešená
8.5	Fazetta filter pre filtrovanie úloh, výsledkov, atď.	Patrik	02.11.2013	08.11.2013	Riešená
8.6	Integrácia cucumber test frameworku, grafy výsledkov	Martin	02.11.2013	08.11.2013	Riešená
8.7	Vytvorenie novej verzie profilu	Miro V.	02.11.2013	08.11.2013	Riešená

3.11 Zázpis zo stretnutia č. 10

Zázpisnica č. 10.

Téma	Dátum	Čas	Miesto
Závèrečné stretnutie	09.12.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčes		

Priebeh stretnutia

- Vyhodnotenie ôsmeho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Dokončenie dokumentácie do odovzdania

Stav úloh z ôsmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
8.1	Refactoring kódu	Filip	02.11.2013	08.11.2013	Splnená
8.2	Vytvorenie ďalších high chartov	Robo	02.11.2013	08.11.2013	Splnená
8.3	Tvorba dokumentácie, kontrola chýb, nekonzistencií, definovanie obsahu	Marek	02.11.2013	08.11.2013	Splnená
8.4	Upravenie zobrazenie badges, vymyslieť nové badges	Miro M.	02.11.2013	08.11.2013	Splnená
8.5	Fazetta filter pre filtrovanie úloh, výsledkov, atď.	Patrik	02.11.2013	08.11.2013	Splnená
8.6	Integrácia cucumber test frameworku, grafy výsledkov	Martin	02.11.2013	08.11.2013	Splnená
8.7	Vytvorenie novej verzie profilu	Miro V.	02.11.2013	08.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
9.1	Dokončenie dokumentácie	Spoločná	09.12.2013	12.12.2013	Riešená

3.12 Zázpis zo stretnutia č. 11

Zázpisnica č. 1.

Pinta Group



Téma	Dátum	Čas	Miesto
Úvodné stretnutie	25.2.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčes		

Pribeh stretnutia

- Prehodnotenie doterajších výsledkov
 - Diskutovali sme doteraz dosiahnuté výsledky a celkový pohľad na aplikáciu
 - Reorganizácia častí aplikácie
 - Konzultovali sme návrhy na podstatné zmeny celkovej štruktúry aplikácie
 - Definovali sme čo je naša pridaná hodnota, v čom bude náš systém lepší ako existujúce
- Diskusia na stretnutí
 - Definovali sme základné časti aplikácie, na ktorých zlepšenie sa musíme zamerať
 - Identifikovali sme už jednoznačné typy kampaní
 - Vybrali sme konkrétnu kampaň - survey (dotazník), ktorú rozpracujeme do detailu
 - Vytvorili sme flow použitia survey kampane a to - vytvorenie survey kampane -> vyplnenie survey kampane používateľom -> validácia -> zobrazenie výsledkov -> odmena
 - Validácia výsledkov je v tomto bode ešte nepotrebná, pridaná bude neskôr
 - Pre vytváranie kampaní sme definovali tiež základné obrazovky, vytváranie bude prebiehať pomocou upravovateľných dlaždíc
 - Dôležité je zobrazenie výsledkov, ich interpretácia - to je naša pridaná hodnota
 - Podstatné je tiež definovanie filtrov, podľa ktorých bude mať vytváraná kampaň rôzne obmedzenia

- Naplánovanie akcií do prvého šprintu
 - Potrebujeme vylepšiť predovšetkým prvý základný typ kampane - survey
 - Musíme sa zamerať na použiteľnosť, teda vytvoriť jednoduché, jednoznačné tlačidlá, dlaždice
 - Survey kampaň bude obsahovať zatiaľ 3 typy otázok: výber áno/nie, viacnásobný výber, textová odpoveď
 - Musíme údaje vhodne zobrazit - pomocou rôznych typov grafov, umožniť tiež export údajov
 - Potrebujeme kontaktovať ľudí z marketingu, ktorí nám ohodnotia použiteľnosť aplikácie

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
0.1	Vytvorenie survey kampane a spomenuté tri typy otázok	Filip, Robo	25.2.2014	4.3.2014	Riešená
0.2	Proces zefektívnenia tvorby zápisníc - automatický generátor šablóny v.1	Marek	25.2.2014	4.3.2014	Riešená
0.3	Prezentácia výsledkov - zmena layoutu, rôzne typy grafov	Miro V.	25.2.2014	4.3.2014	Riešená
0.4	Vytvorenie prezentačnej stránky aplikácie	Martin, Patrik	25.2.2014	4.3.2014	Riešená
0.5	Oprava kľúčových chýb v systéme	Robo	25.2.2014	4.3.2014	Riešená
0.6	Refaktoring spojenej úlohy	Miro M.	25.2.2014	4.3.2014	Riešená

3.13 Zázpis zo stretnutia č. 12

Zázpisnica č. 2.

Pinta Group



Téma	Dátum	Čas	Miesto
Plánovanie 2. šprintu	4.3.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Molnár		

Priebeh stretnutia

- **Vyhodnotenie prvého šprintu**
 - Diskutovali sme o úlohách z predchádzajúceho šprintu, prípadných problémoch pri ich riešení
 - Prebrali sme potrebné veci, ktoré nám chýbajú pre úspešné pokrytie všetkých častí scenára použitia, teda vytvorenie úlohy zadávateľom, vyriešenie úlohy používateľom, zobrazenie štatistických výsledkov a pridelenie odmeny
- **Diskusia na stretnutí**
 - Detekcia pomalého kódu. Určenie problému ako task. Určenie priority tasku na nízku
 - Hovorenie o marketigu, aktuálnom stave
 - Komunikácia členov tímu o cielej funkcionalite programu
 - Navrhnutie nového typu tasku. Task, ktorý bude jednoduchý, ale bude schopný zastrešiť požiadavku na veľa rôznych typov úloh
 - Prehodnotenie štýlovania a výslednej kvality
 - Typy odmeňovania v projekte. Rozhodnúť, aké hodnotenie je najefektívnejšie
 - Umožnenie individuálneho ohodnotenia na základe odmeny od zadávateľa, zadaná prostredníctvom textu
- **Naplánovanie akcií do druhého šprintu**
 - Potrebujeme vytvoriť nový typ úloh (task) - inzerťová úloha, prototyp
 - Potrebujeme vytvoriť kreditný systém pre používateľov, po splnení úloh sa im bude kredit navyšovať
 - Proces vytvorenia úlohy zadávateľom - pozmeniť UI, rozloženie komponentov, logiku vvytvárania
 - Musíme upraviť vizuálnu stránku úloh a pridanie Ajax prvkov pri prechode na úlohu
 - Potrebujeme vytvoriť API pre zadávateľov úloh, ktorí si budú ľahko vedieť definovať vlasné filtre a obmedzenia pri vytváraní úloh

Splnenie úloh z prvého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
0.1	Vytvorenie survey kampane a spomenuté tri typy otázok	Filip, Robo	25.2.2014	4.3.2014	Splnená
0.2	Proces zefektívnenia tvorby zápisníc - automatický generátor šablóny v.1	Marek	25.2.2014	4.3.2014	Splnená
0.3	Prezentácia výsledkov - zmena layoutu, rôzne typy grafov	Miro V.	25.2.2014	4.3.2014	Splnená
0.4	Vytvorenie prezentačnej stránky aplikácie	Martin, Patrik	25.2.2014	4.3.2014	Splnená
0.5	Oprava kľúčových chýb v systéme	Robo	25.2.2014	4.3.2014	Splnená
0.6	Refaktoring spojenej úlohy	Miro M.	25.2.2014	4.3.2014	Splnená

Rozdelenie úloh do druhého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
1.1	Vytvorenie prototypu inzertovej úlohy	Patrik	4.3.2014	11.3.2014	Riešená
1.2	Návrh a implementácia zobrazovania úloh	Filip	4.3.2014	11.3.2014	Riešená
1.3	Prerobenie postupu vytvorenia úloh, zmena UI a rozloženia prvkov	Robo	4.3.2014	11.3.2014	Riešená
1.4	Ďalšie typy grafov, zmena bez znovunačítania dát	Miro V.	4.3.2014	11.3.2014	Riešená
1.5	Úprava vizuálnej stránky riešenia úlohy - prvky Ajax	Martin	4.3.2014	11.3.2014	Riešená
1.6	Vytvorenie základného API pre zadávateľa	Miro M.	4.3.2014	11.3.2014	Riešená
1.7	Generátor zápisníc v.2, úprava vzhľadu zápisnice - pridanie loga	Marek	4.3.2014	11.3.2014	Riešená
1.8	Zoznam aktuálnych úloh v systéme	Filip, Robo	4.3.2014	11.3.2014	Riešená

3.14 Zázpis zo stretnutia č. 13

Zázpisnica č. 3.

Pinta Group



Téma Plánovanie 3. šprintu	Dátum 11.3.2014	Čas 11:30	Miesto Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčes Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Martin Toma		

Pribeh stretnutia

- Vyhodnotenie druhého šprintu
 - Diskutovali sme o úlohách z predchádzajúceho šprintu, prípadných problémoch pri ich riešení
 - Zhodli sme sa na ďalšej iterácii procesu vytvárania úlohy, niektoré veci musíme zmeniť, aby bolo vytváranie úlohy jednoduché a intuitívne
- Diskusia na stretnutí
 - Potrebujeme vytvoriť transakcie pre pripočítavanie kreditu používateľovi, ak nesplní celú úlohu, nesmie sa mu pripočítať kredit
 - Diskusia o potrebu zachovania integrity údajov o výške kreditu, kredit môže byť pripočítaný, aj odpočítaný ak si používateľ dá vyplatiť kredit v hotovosti
 - Komunikácia o vhodnosti zobrazenia výsledkov z úlohy, aké filtre ešte potrebujeme, či sú zobrazované výsledky dostatočné
 - Zhodli sme sa, že v systéme existuje mnoho chýb, ktoré treba odstrániť
 - Dôležité je hlavne odstrániť viditeľné chyby, ktoré priamo ovplyvňujú proces riešenia úloh
 - Čím skôr treba kompletne dokončiť procesy vytvárania a riešenia úloh, aby mohol byť systém daný na testovanie, aby sme získali spätnú väzbu
- Naplánovanie akcií do tretieho šprintu
 - Potrebná ďalšia iterácia procesu vytvárania úloh
 - Potrebujeme vytvoriť transakcie nad kreditom, kvôli konzistencii dát
 - Mali by sme pridať ďalšie typy výsledkov/štatistík pre čiastkové podúlohy v rámci celej úlohy
 - Pri použití vytvoreného API musíme vyriešiť autorizáciu
 - Musíme identifikovať a odstrániť chyby v systéme
 - Potrebujeme zmeniť zobrazovanie odmiern (badges), skompletizovať profil - pridať referral odkaz, vytvoriť seedy do databázy pre následné testovanie

Splnenie úloh z druhého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
1.1	Vytvorenie prototypu inzertovanej úlohy	Patrik	4.3.2014	11.3.2014	Splnená
1.2	Návrh a implementácia zobrazovania úloh	Filip	4.3.2014	11.3.2014	Splnená
1.3	Prerobenie postupu vytvorenia úloh, zmena UI a rozloženia prvkov	Robo	4.3.2014	11.3.2014	Splnená
1.4	Ďalšie typy grafov, zmena bez znovunačítania dát	Miro V.	4.3.2014	11.3.2014	Splnená
1.5	Úprava vizuálnej stránky riešenia úlohy - prvky Ajax	Martin	4.3.2014	11.3.2014	Splnená
1.6	Vytvorenie základného API pre zadávateľa	Miro M.	4.3.2014	11.3.2014	Splnená
1.7	Generátor zápisníc v.2, úprava vzhľadu zápisnice - pridanie loga	Marek	4.3.2014	11.3.2014	Splnená
1.8	Zoznam aktuálnych úloh v systéme	Filip, Robo	4.3.2014	11.3.2014	Splnená

Rozdelenie úloh do tretieho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
2.1	Optimalizácia postupu pri vytvorení úlohy	Patrik	11.3.2014	18.3.2014	Riešená
2.2	Zlepšiť zobrazovanie badges	Filip	11.3.2014	18.3.2014	Riešená
2.3	Identifikovať chyby v systéme a odstrániť	Robo	11.3.2014	18.3.2014	Riešená
2.4	Vylepšenie zobrazenia výsledkov. Pridanie filtrov. Csv výstup dát.	Miro V.	11.3.2014	18.3.2014	Riešená
2.5	Vytvoriť prvú verziu transakcií	Martin	11.3.2014	18.3.2014	Riešená
2.6	HTTP autorizácia v API	Miro M.	11.3.2014	18.3.2014	Riešená
2.7	Pridanie referral do profilu, manualne pridanie skills, sedy do DB, zobrazenie aktuálnych úloh na dashboarde	Marek	11.3.2014	18.3.2014	Riešená

3.15 Zázpis zo stretnutia č. 14

Zázpisnica č. 4.

Pinta Group



Téma	Dátum	Čas	Miesto
Plánovanie 4. šprintu	18.3.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčes Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Róbert Černý		

Priebeh stretnutia

- Vyhodnotenie tretieho šprintu
 - Diskutovali sme o úlohách z predchádzajúceho šprintu, prípadných problémoch pri ich riešení
 - Zhodli sme sa na tom, že chýb v systéme je veľa a potrebujeme ich vyriešiť čím skôr, aby bolo možné testovanie treťou stranou
- Diskusia na stretnutí
 - Identifikovali sme spolu v systéme chyby, ktoré treba čím skôr odstrániť
 - Profil je už v použiteľnej podobe, ešte treba upraviť pridávanie jazykových znalostí, aby bolo konzistenté s LinkedIn údajmi
 - Potrebujeme na dashboarde vedieť filtrovať úlohy podľa konkrétnych preferencií používateľa - ale teraz to nie je priorita
 - Pri vytváraní survey úlohy máme tri typy otázok - textové pole, single choice, a doplniť musíme multiple choice
 - Bolo by vhodné vytvoriť ešte viac seedov kvôli rozmanitosti používateľov a úloh - aj kvôli otestovaniu zobrazovania výsledkov
- Naplánovanie akcií do štvrtého šprintu
 - Musíme pofixovať minoritné veci z predošlého šprintu
 - Pri zobrazených úlohách na dashboarde môžeme priamo zobrazovať stav kompletnosti úlohy
 - Potrebujeme spraviť refactoring kódu, aby sme ľahšie ďalej napredovali
 - Každý dostane za úlohu odstrániť zopár chýb, bugov v systéme, aby sme boli pripravený na testovanie
 - Musíme sa stretnúť s dievčatami z marketingu a dohodnúť sa na testovaní
 - Musíme dokončiť rozšírený abstrakt na TP-CUP

Splnenie úloh z tretieho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
2.1	Optimalizácia postupu pri vytvorení úlohy	Patrik	11.3.2014	18.3.2014	Splnená
2.2	Zlepšiť zobrazovanie badges	Filip	11.3.2014	18.3.2014	Splnená
2.3	Identifikovať chyby v systéme a odstrániť	Robo	11.3.2014	18.3.2014	Splnená
2.4	Vylepšenie zobrazenia výsledkov. Pridanie filtrov. Csv výstup dát.	Miro V.	11.3.2014	18.3.2014	Splnená
2.5	Vytvoriť prvú verziu transakcií	Martin	11.3.2014	18.3.2014	Splnená
2.6	HTTP autorizácia v API	Miro M.	11.3.2014	18.3.2014	Splnená
2.7	Pridanie referral do profilu, manualne pridanie skills, seedy do DB, zobrazenie aktuálnych úloh na dashboarde	Marek	11.3.2014	18.3.2014	Splnená

Rozdelenie úloh do štvrtého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
3.1	Oprava chýb, refactoring	Patrik	18.3.2014	25.3.2014	Riešená
3.2	Transakčný systém v.2	Filip, Robo	18.3.2014	25.3.2014	Riešená
3.3	Vytvorenie dashboardu	Robo	18.3.2014	25.3.2014	Riešená
3.4	Vylepšenie navigačného panelu, oprava niekoľkých chýb, rozšírený abstrakt TP-CUP	Martin	18.3.2014	25.3.2014	Riešená
3.5	Úprava načítavania úrovne jazyka z LinkedIn	Miro M.	18.3.2014	25.3.2014	Riešená
3.6	Doplniť nové seedy, zrevidovať dokumentáciu a obsah doterajších zápisníc	Marek	18.3.2014	25.3.2014	Riešená
3.7	Znovupoužitelnosť výsledkov v detaile tasku	Miro V.	18.3.2014	25.3.2014	Riešená
3.8	Global grid flow design, zoznam notifikácií na hlavnej stránke	Filip	18.3.2014	25.3.2014	Riešená

3.16 Zázpis zo stretnutia č. 15

Zázpisnica č. 5.

Pinta Group



Téma	Dátum	Čas	Miesto
Plánovanie 5. šprintu	25.3.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčes Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčes		

Pribeh stretnutia

- Vyhodnotenie štvrtého šprintu
 - Prešli sme si jednotlivé úlohy zadané v predchádzajúcom šprinte. Diskutovali sme o problémoch a ich riešeníach. Navrhli sme ďalšie kroky. K vzniknutým problémom sme pridali prioritu a zaradili sme ich do plánu.
 - Niektoré chyby v systéme s vysokou prioritou sme pridelili do nasledujúceho šprintu členom tímu, podľa znalostí zdrojových súborov v mieste detegovania chyby.

- Diskusia na stretnutí
 - Diskutovali sme o veľkom počte chýb nachádzajúcich sa v systéme. Zhodli sme sa, že ich priorita je vysoká a je potrebné ich vyriešiť čo najskôr kvôli testovaniu
 - Návrh riešenia transakcií nám nevyhovuje z dôvodu nízkej prehľadnosti. Rozhodli sme sa, že je potrebné preskúmať profesionálne techniky riešenia transakcií a následne transakcie upraviť
 - V systéme nám chýba miesto kde by si používatelia mohli zvýšiť množstvo budgetov, alebo ich premeniť na reálnu meno
 - Pre potreby testovania je potrebné mať správne testovacie prostredie. Je teda potrebné prepracovať generovanie databáze s testovacími dátami

- Naplánovanie akcií do piateho šprintu
 - Nedokončené detaily na úlohách musíme dokončiť v ďalšom šprinte podľa postupov na akých sme sa dohodli na stretnutí kvôli minimalizácii vzniku prípadných chýb

- prideliť sme nové úlohy týkajúce sa rozšírenia funkcionality systému a zlepšenie prostredia pre testovanie Sú pridelené nové úlohy týkajúce sa rozšírenia funkcionality systému a zlepšenie prostredia pre testovanie. Lepšie prostredie nám uľahčí čas strávený pri testovaní aplikácie a taktiež zaobstará kvalitnejší výsledok. Budeme schopný ľahšie odhaľovať nežiaduce správanie
- lepšie prostredie nám uľahčí čas strávený pri testovaní aplikácie a taktiež zaobstará kvalitnejší výsledok. Budeme schopní ľahšie odhaľovať nežiaduce správanie

Splnenie úloh zo štvrtého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
3.1	Oprava chýb, refactoring	Patrik	18.3.2014	25.3.2014	Splnená
3.2	Transakčný systém v.2	Filip, Robo	18.3.2014	25.3.2014	Splnená
3.3	Vytvorenie dashboardu	Robo	18.3.2014	25.3.2014	Splnená
3.4	Vylepšenie navigačného panelu, oprava niekoľkých chýb, rozšírený abstrakt TP-CUP	Martin	18.3.2014	25.3.2014	Splnená
3.5	Úprava načítavania úrovne jazyka z LinkedIn	Miro M.	18.3.2014	25.3.2014	Splnená
3.6	Doplniť nové seedy, zrevidovať dokumentáciu a obsah doterajších zápisníc	Marek	18.3.2014	25.3.2014	Splnená
3.7	Znovupoužitelnosť výsledkov v detaile tasku	Miro V.	18.3.2014	25.3.2014	Splnená
3.8	Global grid flow design, zoznam notifikácií na hlavnej stránke	Filip	18.3.2014	25.3.2014	Splnená

Rozdelenie úloh do piateho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
4.1	Stretnutie s marketérkou, oprava chýb	Patrik, Martin	25.3.2014	1.4.2014	Riešená
4.2	Hall of Fame, nastavenia používateľa, zobrazenie úloh	Filip	25.3.2014	1.4.2014	Riešená
4.3	Responzívny dizajn hlavnej stránky, zlepšenie procesu vytvárania kampaní	Robo	25.3.2014	1.4.2014	Riešená
4.4	Internacionalizácia v.2	Martin	25.3.2014	1.4.2014	Riešená
4.5	Unikátne dáta v profile - validácia	Miro M.	25.3.2014	1.4.2014	Riešená
4.6	Nové typy podúloh - multichoice, likertova škála, dropdown ranking, constant sum,...	Marek	25.3.2014	1.4.2014	Riešená
4.7	Oprava chýb v systéme	Miro V.	25.3.2014	1.4.2014	Riešená

3.17 Zápis zo stretnutia č. 16

Zápisnica č. 6.

Pinta Group



Téma	Dátum	Čas	Miesto
Plánovanie 6. šprintu	1.4.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčes Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Patrik Štrba		

Priebeh stretnutia

- Vyhodnotenie piateho šprintu
 - diskutovali sme o funkcionalite transakcií v systéme
 - novo navrhnuté riešenie bolo vytvorené podľa pravidiel, ktoré sú doporučované, no zhodli sme sa, že pre naše účely je tento návrh príliš zložitý a ťažko verifikovateľný
 - na to aby ho bolo možné použiť je potreba veľa úsilia
 - pôvodné riešenie je jednoduché a plne funkčné, preto sme sa rozhodli ostať pri pôvodnom riešení

- Diskusia na stretnutí
 - konzultovali sme o novovzniknutých problémoch a funkcionalite platobnej brány
 - bolo nám umožnené využiť znalosti marketérov pre účely zlepšenia našej aplikácie
 - zhodnotili sme aktuálny stav aplikácie a potrebné kroky pre jej upravenie na testovanie skupinou Pe-We
 - pre zlepšenie informovanosti používateľa o jeho aktuálnom stave v aplikácii a zároveň zvýšenie motivácie používateľa sme sa rozhodli do implementovať Hall of Fame, ktorý podá informácie o bodoch iných používateľov v systéme, tak používateľ uvidí rebríček používateľov a bude sa snažiť v ňom postúpiť vyššie
 - pre rozšírenie funkcionality sme sa dohodli, že treba upraviť Notice a generovať html pri jeho zobrazení
 - zistili sme nové chyby v zabezpečení aplikácie a možnostiach pri vyplňaní profilu. Aplikácia sa ukázala ako ľahko zraniteľnou voči útoku používateľmi, ktorí budú môcť vytvárať nereálne záznamy v profile.

- Naplánovanie akcií do šiesteho šprintu

- do ďalšieho šprintu sme sa rozhodli rozdeliť členom tímu nájdené chyby s najvyššou prioritou.
- k využitiu pracovných zdrojov naplno sme popridelovali aj úlohy, ktoré sme nestihli celkom dokončiť alebo na stretnutí sme rozhodli o doplnení funkcionality
- ostatné zdroje boli orientované na zlepšenie aplikácie pre testovanie skupinou pewe a doplnením funkcionality systému

Splnenie úloh z piateho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
4.1	Stretnutie s marketérkou, oprava chýb	Patrik, Martin	25.3.2014	1.4.2014	Splnená
4.2	Hall of Fame, nastavenia používateľa, zobrazenie úloh	Filip	25.3.2014	1.4.2014	Splnená
4.3	Responzívny dizajn hlavnej stránky, zlepšenie procesu vytvárania kampaní	Robo	25.3.2014	1.4.2014	Splnená
4.4	Internacionalizácia v.2	Martin	25.3.2014	1.4.2014	Splnená
4.5	Unikátne dáta v profile - validácia	Miro M.	25.3.2014	1.4.2014	Splnená
4.6	Nové typy podúloh - multichoice, likertova škála, dropdown ranking, constant sum,...	Marek	25.3.2014	1.4.2014	Splnená
4.7	Oprava chýb v systéme	Miro V.	25.3.2014	1.4.2014	Splnená

Rozdelenie úloh do šiesteho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
5.1	IIT.SRC, úprava homepage obsahu	Martin	1.4.2014	8.4.2014	Riešená
5.2	Autocomplete	Filip, Miro M.	1.4.2014	8.4.2014	Riešená
5.3	Refaktorovanie jadra aplikácie	Robo	1.4.2014	8.4.2014	Riešená
5.4	Vytvorenie funkčných a unit testov	Patrik	1.4.2014	8.4.2014	Riešená
5.5	Validácia nových typov úloh a ukladanie výsledkov	Marek	1.4.2014	8.4.2014	Riešená
5.6	Umožniť používateľovi ovplyvniť vzhľad textu otázky	Miro V.	1.4.2014	8.4.2014	Riešená

3.18 Zázpis zo stretnutia č. 17

Zázpisnica č. 7.

Pinta Group



Téma	Dátum	Čas	Miesto
Plánovanie 7. šprintu	8.4.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčes Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Vojtuš		

Priebeh stretnutia

- Vyhodnotenie šiesteho šprintu
 - členovia tímu prezentovali svoje výsledky a informovali o novovzniknutých problémoch
 - zhodnotil sa celkový stav aplikácie po šprinte
- Diskusia na stretnutí
 - diskutovali sme o výsledkoch testovania aplikácie skupinou Pe-We, testovanie nám prinieslo mnoho informácií, ktoré nám pomohli
 - overiť správnosť požiadaviek určených na začiatku vývoja. Pri testovaní sme odhalili veľký počet chýb, ktoré sa dôkladne zdokumentovali a diskutovali na stretnutí.
 - nové informácie nás prinútili prehodnotiť požiadavky na systém. Rozhodli sme sa upraviť požiadavky a následne upraviť projektový plán. U všetkých úloh v systéme sme museli prehodnotiť ich prioritu
 - kvôli blížiacej sa súťaži TP-CUP sme prehodnotili funkcionality nášho systému a jeho náchylnosť na chyby, tiež musíme vytvoriť potrebné dokumenty pre úspešnú účasť na TP-CUP
 - navrhli sme postupy testovania aplikácie a zaznamenávania chýb, rovnako sme navrhli aj postupy ich overovania a následného pridelenia
 - určili sme si presné termíny pre rozdelenie novovzniknutých chýb. Týmto spôsobom sme chceli doceliť nielen nájdenie, ale aj vyriešenie nájdených chýb jedným šprinte
- Naplánovanie akcií do siedmeho šprintu
 - rozdelili sme úlohy týkajúce sa účasti na TP-CUP, definovali sme si dokumenty a potrebné dáta pre čo najlepšiu prezentáciu nášho projektu
 - pre prezentáciu funkčnej aplikácie bolo pridelené testovanie aplikácie jednotlivým členom tímu

Splnenie úloh zo šiesteho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
5.1	IIT.SRC, úprava homepage obsahu	Martin	1.4.2014	8.4.2014	Splnená
5.2	Autocomplete	Filip, Miro M.	1.4.2014	8.4.2014	Splnená
5.3	Refaktorovanie jadra aplikácie	Robo	1.4.2014	8.4.2014	Splnená
5.4	Vytvorenie funkčných a unit testov	Patrik	1.4.2014	8.4.2014	Splnená
5.5	Validácia nových typov úloh a ukladanie výsledkov	Marek	1.4.2014	8.4.2014	Splnená
5.6	Umožniť používateľovi ovplyvniť vzhľad textu otázky	Miro V.	1.4.2014	8.4.2014	Splnená

Rozdelenie úloh do siedmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
6.1	Oprava identifikovaných chýb v systéme	Martin	8.4.2014	15.4.2014	Riešená
6.2	Oprava chýb v systéme	Filip, Miro M.	8.4.2014	15.4.2014	Riešená
6.3	Automatické uzamykanie kampaní (CRON), rozdelenie seedov	Robo	8.4.2014	15.4.2014	Riešená
6.4	Nasadzovanie aplikácie na produkčný server a riešenie chýb s tým spojených	Patrik	8.4.2014	15.4.2014	Riešená
6.5	Oprava chýb v systéme	Marek	8.4.2014	15.4.2014	Riešená
6.6	Oprava chýb v systéme	Miro V.	8.4.2014	15.4.2014	Riešená

3.19 Zázpis zo stretnutia č. 18

Zázpisnica č. 8.

Pinta Group



Téma	Dátum	Čas	Miesto
Plánovanie 8. šprintu	15.4.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčes Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Molnár		

Pribeh stretnutia

- Vyhodnotenie siedmeho šprintu
 - prehodnotili sme opravu chýb v systéme, pozreli sme sa do systému JIRA a zistili, či ešte existujú nejaké reportované chyby, ktoré neboli vyriešené
 - vyhodnotením sme zistili, že sme odstránili všetky známe chyby a môžeme pokračovať v pridávaní ďalšej funkcionality
- Diskusia na stretnutí
 - zhodli sme sa, že treba vyriešiť všetky náležitosti pre blížiacu sa súťaž TP-CUP
 - musíme vytvoriť článok na portáli Robime.it
 - pre prezentáciu produktu na TP-CUP súťaži potrebujeme mať hotové prezentačné materiály
 - musíme ešte posledný raz pred prezentáciou odhaliť prípadné ďalšie chyby v systéme
 - aplikáciu musíme nasadiť na produkčný server
 - keďže máme dvojtýždňový šprint, máme dostatok času na rôzne úpravy a doladovanie
 - musíme zabezpečiť prezentačné materiály a trička na súťaž TP-CUP
- Naplánovanie akcií do ôsmeho šprintu
 - vytvorenie článku na portáli Robime.IT
 - vytvorenie statických stránok v aplikácii
 - nasadenie aplikácie do produkcie
 - upraviť zobrazenie ocenení aby bolo konzistentné s dizajnom
 - pridanie nových funkcií na dashboard

Splnenie úloh zo siedmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
6.1	Oprava identifikovaných chýb v systéme	Martin	8.4.2014	15.4.2014	Splnená
6.2	Oprava chýb v systéme	Filip, Miro M.	8.4.2014	15.4.2014	Splnená
6.3	Automatické uzamykanie kampaní (CRON), rozdelenie seedov	Robo	8.4.2014	15.4.2014	Splnená
6.4	Nasadzovanie aplikácie na produkčný server a riešenie chýb s tým spojených	Patrik	8.4.2014	15.4.2014	Splnená
6.5	Oprava chýb v systéme	Marek	8.4.2014	15.4.2014	Splnená
6.6	Oprava chýb v systéme	Miro V.	8.4.2014	15.4.2014	Splnená

Rozdelenie úloh do ôsmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
7.1	Robime.IT článok, UI vylepšenia a bugfixy	Martin	15.4.2014	29.4.2014	Riešená
7.2	Povolenie správnych hodnôt v Education a Positions	Miro M.	15.4.2014	29.4.2014	Riešená
7.3	Doplnenie transakčnej logiky do notice kampane, automatické počítanie ceny kampane	Robo	15.4.2014	29.4.2014	Riešená
7.4	Úprava aplikácie a oprava chýb a nasadenie do produkcie	Patrik	15.4.2014	29.4.2014	Riešená
7.5	Vytvorenie statických stránok, validácia času vytvorenia kampane a nový datetimerpicker	Marek	15.4.2014	29.4.2014	Riešená
7.6	Pridanie nových typov achievementov. Vytvoriť používateľsky prívetivejšie obrázky.	Miro V.	15.4.2014	29.4.2014	Riešená
7.7	Vytvorenie prezentačného videa	Robo, Filip	15.4.2014	29.4.2014	Riešená
7.8	User management + transakcie, nové funkcie do dashboardu, filtre	Filip	15.4.2014	29.4.2014	Riešená

3.20 Zápis zo stretnutia č. 19

Zápisnica č. 9.

Pinta Group



Téma	Dátum	Čas	Miesto
Záverečné stretnutie	6.5.2014	11:30	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Miroslav Molnár Bc. Marek Lenčేశ Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčేశ		

Priebeh stretnutia

- **Vyhodnotenie ôsmeho šprintu**
 - Zhodnotili sme splnenie úloh z ôsmeho šprintu a skontrolovali sme či sú vyriešené aj všetky reportované chyby v systéme JIRA
- **Diskusia na stretnutí**
 - dohodli sme sa na záverečných krokoch v rámci vývoja aplikácie
 - rozdelili sme si úlohy pre dokončenie dokumentácie
 - každý doplní opisy svojej práce do spoločnej projektovej dokumentácie k inžinierskemu dielu
 - každý doplní prípadné ďalšie informácie do časti riadenia a konkrétnych typov manažmentov
 - všetci sa budeme podieľať na finálnej verzii dokumentácie
 - treba vyriešiť typografickú a gramatickú úpravu - Marek
 - treba spraviť zoznam všetkých náležitostí potrebných pre odovzdanie - CD, dokumentácia, zdrojové kódy, . . .

Splnenie úloh z ôsmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
7.1	Robime.IT článok, UI vylepšenia a bugfixy	Martin	15.4.2014	29.4.2014	Splnená
7.2	Povolenie správnych hodnôt v Education a Positions	Miro M.	15.4.2014	29.4.2014	Splnená
7.3	Doplnenie transakčnej logiky do notice kampane, automatické počítanie ceny kampane	Robo	15.4.2014	29.4.2014	Splnená
7.4	Úprava aplikácie a oprava chýb a nasadenie do produkcie	Patrik	15.4.2014	29.4.2014	Splnená
7.5	Vytvorenie statických stránok, validácia času vytvorenia kampane a nový datetimer picker	Marek	15.4.2014	29.4.2014	Splnená
7.6	Pridanie nových typov achievementov. Vytvorit používateľsky prívetivejšie obrázky.	Miro V.	15.4.2014	29.4.2014	Splnená
7.7	Vytvorenie prezentačného videa	Robo, Filip	15.4.2014	29.4.2014	Splnená
7.8	User management + transakcie, nové funkcie do dashboardu, filtre	Filip	15.4.2014	29.4.2014	Splnená

4 Úlohy členov tímu

4.1 Manažérske roly v tíme

Manažérske roly v tíme sme si rozdelili pri prvom šprinte, každý člen tímu má pri tvorbe projektu hlavnú manažérsku zodpovednosť za fungovanie danej časti vývoja produktu.

Meno	Zodpovednosť
Bc. Martin Toma	Manažér komunikácie a ľudských zdrojov (vedúci tímu)
Bc. Filip Bednárík	Manažér rizík
Bc. Róbert Černý	Manažér rozvrhu a plánovania
Bc. Marek Lenčes	Manažér dokumentovania
Bc. Miroslav Molnár	Manažér monitorovania
Bc. Miroslav Vojtuš	Manažér kvality
Bc. Patrik Štrba	Manažment podpory vývoja

4.2 Krátkodobé úlohy

Meno	Úloha
Bc. Martin Toma	Manažér komunikácie a ľudských zdrojov (vedúci tímu)
Bc. Filip Bednárík	Manažér rizík
Bc. Róbert Černý	Manažér rozvrhu a plánovania
Bc. Marek Lenčes	Manažér dokumentovania
Bc. Miroslav Molnár	Manažér monitorovania
Bc. Miroslav Vojtuš	Manažér kvality
Bc. Patrik Štrba	Manažment podpory vývoja

4.3 Dlhodobé úlohy

Meno	Zodpovednosť
Bc. Martin Toma	Zodpovedá za riadenie a usmerňovanie komunikácie v tíme. Rovnako má za úlohu sprostredkovať komunikáciu medzi zadávateľom projektu a tímom. Okrem toho dohliada nad nástrojom JIRA do ktorého po každom stretnutí zapíše a pridelí jednotlivé úlohy. Má prehľad o celom systéme a rieši vysoko úrovňové problémy vrámci celej produktovej domény.
Bc. Filip Bednárík	Má úlohu hlavného inšpektora pri prehliadkach zdrojového kódu. Je softvérovým architektom a stará sa o integritu softvérového riešenia. Tiež má za úlohu identifikovať včas riziká a s pomocou tímu korektne na tieto riziká reagovať. Popri tom píše zápisky a podieľa sa na identifikovaní a implementácii nových funkcií softvéru. Kontrola zmien v zdrojovom kóde pred ich vložení na globálny repozitár tímu.
Bc. Róbert Černý	Má za úlohu dozeráť na správnosť priebehu tímového stretnutia. Kontroluje termíny a dozerá na ich dodržiavanie. Navrhne dátumy a synchronizuje ich so zdieľaným tímovým kalendárom. Dozerá na plnenie krátkodobého a dlhodobého plánu.
Bc. Marek Lenčes	Má za úlohu spracovávať výslednú podobu dokumentácie, skontrolovať nekonzistentné označovanie rovnakých pojmov, opraviť gramatické chyby, spracovať výsledné formátovanie dokumentu, a zabezpečiť, aby obsahoval všetky náležitosti potrebné pri jeho odovzdaní. Tiež má za úlohu spracovávať zápisky zo stretnutí, kontrolovať gramatické chyby, zabezpečiť pravdivosť údajov v zápisnici, a aktualizovať tímový web pravidelným pridávaním zápisníc a aktuálnych verzií obidvoch častí dokumentácie na stránku tímového webu.
Bc. Miroslav Molnár	Zabezpečenie monitorovania aktuálneho stavu projektu voči dlhodobému plánu projektu. Informovať členov tímu o stave projektu. Navrhnuť optimálne riešenia pri zmene plánu z hľadiska nazbieraných informácií z predchádzajúceho monitorovania.
Bc. Miroslav Vojtuš	Zabezpečiť kvalitu produktu podľa požiadaviek. Robí preventívne testy funkčných častí systému. Rozhoduje o zavedení a hĺbke automatizovaného testovania.
Bc. Patrik Štrba	Správa spoločného repozitára a procesu nasadenia projektu na produkčný server. Kontrola pull-requestov a zabezpečovanie korektnosti dát v repozitári. Vytváranie master vetvy pre produkčný server a vytváranie nových verzií produktu.

4.4 Podiel členov na tvorbe dokumentácie k inžinierskemu dielu

4.4.1 Zimný semester

Meno	Podiel
Bc. Martin Toma	13 %
Bc. Filip Bednárík	13 %
Bc. Róbert Černý	13 %
Bc. Marek Lenčes	22 %
Bc. Miroslav Molnár	13 %
Bc. Patrik Štrba	13 %
Bc. Miroslav Vojtuš	13 %

4.4.2 Letný semester

Meno	Podiel
Bc. Martin Toma	12,5 %
Bc. Filip Bednárík	12,5 %
Bc. Róbert Černý	12,5 %
Bc. Marek Lenčes	22 %
Bc. Miroslav Molnár	15,5 %
Bc. Patrik Štrba	12,5 %
Bc. Miroslav Vojtuš	12,5 %

4.5 Podiel členov pri tvorbe diela

4.5.1 Zimný semester

Meno	Podiel
Bc. Martin Toma	14,3 %
Bc. Filip Bednárík	14,3 %
Bc. Róbert Černý	14,3 %
Bc. Marek Lenčes	14,3 %
Bc. Miroslav Molnár	14,3 %
Bc. Patrik Štrba	14,3 %
Bc. Miroslav Vojtuš	14,3 %

4.5.2 Letný semester

Meno	Podiel
Bc. Martin Toma	18 %
Bc. Filip Bednárík	20 %
Bc. Róbert Černý	18 %
Bc. Marek Lenčes	11 %
Bc. Miroslav Molnár	11 %
Bc. Patrik Štrba	11 %
Bc. Miroslav Vojtuš	11 %

5 Manažment rozvrhu a plánovania

Autor: Bc. Róbert Černý

V tejto kapitole sú opísané procesy, ktoré prebiehajú počas plánovania. Náš tím si vybral agilnú metódu vývoja. Používame teda *Scrum*, čo je agilný, prírastkový a iteratívny projektový manažment.

5.1 Procesy

5.1.1 Tvorba a udržiavanie produktového backlogu

Vstup: žiadny alebo nevyhovujúci produktový backlog

Výstup: produktový backlog so všetkými náležitosťami

Produktový backlog vytvára produktový vlastník na základe priorít a požiadaviek zákazníkov. Obsahuje používateľské príbehy, ktorá sa zväčša dajú rozdeliť do jednoduchších podúloh. Forma zápisu týchto príbehov je obsiahnutá v šablóne nižšie. Backlog sa nachádza v systéme *Atlassian JIRA*®, ktorý je vhodný na jeho reprezentáciu.

	Popis	Príklad
ID	Identifikátor, začíname od 0001	0212
Názov	Maximálne jedna veta, stručná, jasná, vyjadrujúca podstatu	Autentifikácia používateľa
Dôležitosť	Určuje produktový vlastník a je to číselná hodnota, čím vyššia, tým vyššia priorita	110
Odhad prácnosti	Slúži na porovnanie s inými príbehmi, ide o odhad, určuje ho spoločne celý tím <i>hodnota = počet ľudí x počet hodín</i>	10
Demonštrovanie	Definujeme, ako je možné daný príbeh prezentovať	Registrujeme sa, autentifikujeme sa pomocou odkazu z e-mailu, prihlásime sa, vstúpime na domovskú stránku
Poznámky	Doplňujúce informácie	Použijeme vhodný GEM

Odhad prácnosti je atribút, ktorý sa odhaduje až pri rozdeľovaní úloh medzi členmi pri tvorbe backlogu šprintu. Počas vývoja sa totižto odhad členov tímu spresňuje, čo zefektívni proces plánovania a dodržiavanie termínov.

5.1.2 Tímové stretnutia a šprinty

Stretnutia prebiehajú opakovane, konkrétny deň v týždni. Jednotlivé šprinty začínajú od ukončenia stretnutia a trvajú až do začatia nasledujúceho stretnutia. Nakoľko nie je možné šprintovať neustále, každý si vyberie ľubovoľný deň v týždni,

kedy oddychuje. Počas šprintu sa predpokladá, že všetci členovia chápu priradené úlohy.

Stretnutia prebiehajú v miestnosti s pripojením na internet a tabuľou pre jednoduchšiu spoluprácu pri vizualizácii. Trvanie tímového stretnutia sa zbytočne nepredlžuje a trvá v priemere hodinu. V tomto časovom intervale prebiehajú postupne podľa poradia tieto procesy:

Revízia šprintu a jeho výsledky	5.1.2.A
Príprava a plánovanie šprintu	5.1.2.B

5.1.2.A Revízia šprintu a jeho výsledky

Vstup: backlog ukončeného šprintu

Výstup: aktualizovaný produktový backlog, úspešnosť šprintu, vyhodnotenie

Týmto procesom začína každé stretnutie a slúži na vyhodnotenie šprintu. Každý člen sa v priebehu pár minút vyjadrí ako sa mu v šprinte darilo. Zhodnotí svoje výsledky a predvedie demo na ktorom preukáže svoju prácu. Vedúci tímu označí úlohu za hotovú ak predvedenie spĺňa požiadavky úlohy. Na konci revízie sa nedokončené úlohy vrátia do produktového backlogu a v prípade potreby sa upraví ich dôležitosť alebo sa zaradia do nasledujúceho šprintu.

Ďalšou súčasťou tohto procesu je prehliadka kódov, ktorá má informatívnu, ako aj kontrolnú funkciu. Pomocou metodiky na prehliadku kódov je možné informovať programátorov o nových algoritmoch či zmenách v implementácii. Metodika písania zdrojových kódov zrýchli tento proces, nakoľko bude mať zdrojový kód uniformnú štruktúru.

Vedúci tímu má na konci priestor pre vznesenie postrehov a problémov ktoré sa vyskytli počas šprintu. Je potrebné aby navrhol riešenie a tím sa začne riadiť podľa týchto pravidiel.

5.1.2.B Príprava a plánovanie šprintu

Prvým krokom v tomto procese je definovať cieľ šprintu. Príkladom môže byť: „dokončenie 3 používateľským príbehov“. Vedúci tímu vyberie člena, ktorý bude zapisovať informácie o stretnutí a šprinte do zápisnice. Zverejnenie zápisnice na tímovom webe je povinnosťou manažéra dokumentácie, ktorý robí poslednú revíziu a opravu zápisnice. Pred rozchodom tímu sa dohodne termín ďalšieho stretnutia. Priestor tu majú hlavne členovia, ktorí pracujú v skupine. Všeobecne však platí, že hlavné stretnutie sa koná presne o týždeň, kedy končí šprint.

Tvorba backlogu šprintu	5.1.2.B.a
Priradenie úloh členom	5.1.2.B.b
Priradenie úloh členom	5.1.2.B.c

5.1.2.B.a Tvorba backlogu šprintu

Vstup: produktový backlog je v poriadku

Výstup: nový backlog nasledujúceho šprintu

V tomto procese sa vytvára backlog pre nastávajúci šprint. Pred začiatkom je však potrebné aby bol produktový backlog aktuálny. Nové úlohy vyberá produktový vlastník, ktorý zároveň komunikuje s tímom o tomto výbere. Členovia tímu spoločne odhadujú, koľko úloh bude možné v jednom šprinte splniť. Tento odhad je subjektívny a najdôležitejšou súčasťou tohto procesu je vzájomná komunikácia. Po výbere úloh je potrebné určiť ich priradenie.

5.1.2.B.b Priradenie úloh členom

Vstup: backlog nasledujúceho šprintu

Výstup: úlohy sú priradené, každý vie na čom pracuje

Jednotlivé úlohy sa v systéme JIRA priradia v priebehu prvého dňa šprintu. Vedúci tímu postupne priradí tieto úlohy priamo na stretnutí. Priradenie sa uchová v zápisnici. Úlohy prideliujeme podľa žiadostí členov. Inak ich prideli vedúci tímu podľa vlastného uváženia.

5.1.2.B.c Rozdelenie úloh na podúlohy

Vstup: úlohy priradené konkrétnemu členovi (skupine)

Výstup: podúlohy jednotlivých úloh

Tento proces je vhodné použiť v prípade, že priradená úloha je zložitejšia a je možné ju logicky rozdeliť na jednoduchšie časti. V prípade že konkrétnu úlohu majú priradenú viacerí členovia, je vhodné aby si navzájom priradili podúlohy podľa svojho uváženia. Tento proces prebehne priamo na tímovom stretnutí. Výsledkom je priradenie elementárnejších úloh jednotlivým členom tímu, ktoré sú zapísané do zápisnice.

Vedúci tímu sa snaží čo najskôr aktualizovať systém JIRA a nastaviť aktuálnu tabuľu šprintu.

5.1.3 Komunikácia počas šprintu

Počas šprintu sú všetky relevantné informácie o priebehu a stave uvedené v systéme JIRA. Toto je prvé miesto ktoré treba navštíviť v prípade nejasností. Tabuľka šprintu je prístupná pre všetkých a je vždy aktuálna.

Pri objavení novej chyby v zdrojových kódach sa jej reprezentácia pridá do systému JIRA ako nový problém. Samotné priradenie zabezpečuje vedúci tímu.

V prípade implementačných nejasností sa člen tímu primárne obracia na zadávateľa danej úlohy. Ak na úlohe spolupracuje viac členov, komunikácia prebieha priamo medzi nimi.

Komunikačné kanály podľa priority:

- JIRA (komentovanie, priradovanie úloh, tabuľka šprintu)
- Google Hangouts (možnosť konferenčných hovorov)
- Google Groups (vytváranie príspevkov podľa tém)

- Google Calendar (termíny PintaGroup)
- mobilný telefón alebo iný osobný kontakt

Pri problémoch s dodržaním termínov je nutné upozorniť vedúceho tímu, ktorý zasiahne do priebehu šprintu a jeho zmenu zverejní v systéme JIRA, pričom sa odošle upozornenie o zmene formou e-mailu. O tejto zmene upozorní aj produktového vlastníka.

6 Manažment rizík

Autor: Bc. Filip Bednárík

Riziká rozdelujeme na rôzne typy. Podľa základného delenia na generické a špecifické. Generické riziká sú podobné pri všetkých softvérových projektoch. Vybrali sme najčastejšie a popísali sme ako tieto generické riziká vplývajú priamo na náš projekt a ako chceme znížiť tieto riziká na minimum.

Riziko redundantnej práce

Typ rizika: Generické, Organizačné, Zdroje

Pravdepodobnosť: Vysoká

Popis rizika: Pri používaní nových technológií sa často stretávame s nepreskúmanými problémami, na ktoré je potrebné hľadať riešenia, ktoré môžu byť časovo náročné. Je neefektívne ak sa na tom istom probléme zaseknú dvaja vývojári.

Škody a dôsledky rizika: Oneskorenie termínov, Demotivácia a frustrácia zamestnancov, Vyššie náklady na vývoj.

Riešenie rizika: Pre zníženie rizika pochybenia plánujeme organizovať pravidelné stretnutia tímu s prezentáciami problémov, ktoré sa vyskytnú pri vývoji alebo návrhu aplikácie a tiež o potencionálne vhodných technológiách vhodných pre tento projekt a existujúcich modulov, tzv. „gemov“.

Nerealistické rozvrhy a rozpočty

Typ rizika: Generické, Manažment projektu, Plánovanie

Pravdepodobnosť: Stredná

Popis rizika: Plány a rozvrhy môžu byť nerealistické, rozpočty môžu byť prehnané alebo nedostatočné.

Škody a dôsledky rizika: Príliš tesný plán môže spôsobiť: Nesplnenie termínov, Demotivácia a frustrácia zamestnancov. Naopak príliš voľný plán: Pomalý vývoj, vyššie náklady, nespokojnosť zákazníkov pri pomalom riešení problémov s aplikáciou. Prehnaný rozpočet: Nezískanie investora. Príliš nízky rozpočet: Problémy s dodatočnými požiadavkami o zvýšenie, nespokojnosť zamestnancov.

Riešenie rizika: Pri plánovaní počítame s rôznymi možnosťami. Vývoj bude agilný a inkrementálny, a využijeme systém pre správu úloh, do ktorého pravidelne členovia tímu budú vkladať podrobné odhady. Chceme robiť prototypy aplikácie.

Zlý návrh aplikácie.

Typ rizika: Generické, Technické, Kvalita

Pravdepodobnosť: Stredná

Popis rizika: Aplikácia môže mať zlú architektúru, jednotlivé časti zle oddelené alebo neoddelené vôbec.

Škody a dôsledky rizika: Nekonzistentnosť aplikácie, Redundancia kódu, Opakovanie chýb a z toho vyplývajúce vyššie náklady na vývoj.

Riešenie rizika: Jednotlivé časti aplikácie budú vo forme modulov a niektoré moduly budú použiteľné aj osobitne. Modulárnosť nám zabezpečí možnosť využitia rovnakých modulov v rôznych častiach aplikácie. Na návrhu aplikácie sa bude podieľať skúsený vývojár.

Vypracovanie nevyhovujúcej funkcionality.

Typ rizika: Generické, Manažment projektu, Komunikácia

Pravdepodobnosť: Vysoká

Popis rizika: Používateľom sa funkcionalita nemusí páčiť, môže byť zbytočná alebo v aplikácii dokonca prekážať.

Škody a dôsledky rizika: Úbytok počtu používateľov a teda nižší zisk, Zbytočne vynaložené zdroje na vývoj nepotrebných funkcií, oddialenie vývoja potrebnej funkcionality.

Riešenie rizika: Pred vytváraním novej funkcionality vypracujeme prieskum, či by takáto funkcionalita vyhovovala. Máme tiež záujem o overené technológie a funkcionalitu. Aplikáciu, vzhľadom na to, že beží na našom hardvéri, môžeme aktualizovať a opravovať priebežne a často. Aplikácia bude navrhnutá tak aby bola používateľsky prívetivá a nepotrebovala množstvo dokumentácie, aj keď pomocníka do aplikácie zahrnieme. Používatelia budú mať možnosť odoslať spätnú väzbu priamo v aplikácii.

Vytvorenie nevyhovujúceho používateľského rozhrania.

Typ rizika: Generické, Technické, Zložitosť a rozhrania

Pravdepodobnosť: Stredná

Popis rizika: Pri návrhu aplikácie sa nepočíta s prostredím, kde sa aplikácia bude používať.

Škody a dôsledky rizika: Úbytok používateľov a teda nižšie zisky.

Riešenie rizika: Základom kvalitného používateľského rozhrania je správna charakteristika používateľa. Používateľské rozhranie preto budeme musieť prispôbiť tak, aby bolo vhodné pre technicky nezdatného používateľa ale zároveň zachovať prvky ako skratky pre expertov. Pre všetky spôsoby použitia našej aplikácie preto vypracujeme scenáre, na základe ktorých prispôbíme aj používateľské rozhranie. Vývoj rozhrania bude prebiehať v šablónach a teda v budúcnosti si bude môcť používateľ vybrať používateľské rozhranie, ktoré mu bude vyhovovať.

Neustále zmeny požiadaviek.

Typ rizika: Generické, Externé, Zákazník

Pravdepodobnosť: Nízka

Popis rizika: Zdrojom požiadaviek bude spätná väzba od používateľov vo forme prieskumov ale, zo samotnej aplikácie pomocou formulára ale aj logmi o používaní aplikácie.

Škody a dôsledky rizika: Zvýšené náklady na vývoj, väčšia šanca na vypracovanie nevyhovujúcej funkcionality.

Riešenie rizika: Na základe týchto informácií vytvoríme požiadavky na zmeny aplikácie, ktoré budeme postupne zahŕňať do jednotlivých inkrementov aplikácie. Jednotlivé verzie aplikácie budú vydávané často a nepravidelne podľa požiadaviek. Jednotlivé funkcie však budú pred vydaním otestované, v prípade veľkej funkcionality prístupníme aj beta verziu produktu.

Nedostatky výkonu v reálnom čase.

Typ rizika: Generické, Technické, Výkon a spoľahlivosť

Pravdepodobnosť: Vysoká

Popis rizika: Je možné, že zvyšujúcim množstvom používateľov nebude stačiť výkon serverov. Niektorí používatelia môžu mať problém s pomalým pripojením alebo s veľkosťou obrázkov a skriptov.

Škody a dôsledky rizika: Spomalenie aplikácie, nespokojnosť používateľov, odrezanie počtu používateľov od funkčnosti. Spomalenie alebo nedostupnosť serveru. Úbytok používateľov a straty v ziskoch.

Riešenie rizika: Pravidelné benchmarky aplikácie a záťažové testy serverov. Kvalitné logovanie akcií používateľa v našej aplikácii nám umožní lepšie odhaliť problémy s jednotlivými modulmi a tieto problémy rýchlo odstrániť. Naša aplikácia bude mať rôzne nastavenia a režimy, v ktorých môže fungovať, čím chceme zabezpečiť pohodlné používanie pre rôzne typy používateľov.

Nedostatok zadávateľov úloh.

Typ rizika: Špecifické, Externé, Zadávatel

Pravdepodobnosť: Vysoká

Popis rizika: Môže sa stať, že nebude dostatočný záujem zo strany zadávateľov úloh, alebo zadávatelia nebudú potrebovať použiť náš systém na zadanie úlohy, pretože nebudú mať potrebu riešiť svoju úlohu externým spôsobom.

Škody a dôsledky rizika: Nedostatok úloh. Nespokojnosť používateľov. Narastajúca nedôveryhodnosť aplikácie. Neschopnosť ďalej prevádzkovať aplikáciu. Úbytok používateľov a straty v ziskoch.

Riešenie rizika: Pre používanie aplikácie musíme zadávateľov dostatočne motivovať. Podstatou je dostatočne oboznámiť potencionálnych zadávateľov s funkčnosťou a princípom zadávania úloh. Dôležité je tiež zdôraznenie výhod, ktoré zadávateľ získa, ak úlohu do systému zadá. Podstatná časť je teda vytvorenie reklamy a prezentovanie aplikácie rôznymi spôsobmi, a zdôraznenie jej predností.

Nedostatok záujmu pracovníkov.

Typ rizika: Špecifické, Externé, Zákazník

Pravdepodobnosť: Stredná

Popis rizika: Ak nastane situácia, že zadávateľov aj úloh bude dostatok, ale nebude dostatok pracovníkov na riešenie úloh, nastáva problém. Pracovníkom sa odmeny môžu zdať príliš nízke, za ktoré sa im neoplatí danú úlohu vypracovať.

Škody a dôsledky rizika: Odškodnenie zadávateľov. Narastajúca nedôveryhodnosť aplikácie. Neschopnosť ďalej prevádzkovať aplikáciu. Úbytok používateľov(zadávatelov) a straty v ziskoch.

Riešenie rizika: Pre čo najväčšie motivovanie pracovníka je potrebné udržať pracovníka v napätí, a v túžbe po vyššom zisku. Preto je riešením vytvorenie odmeňovacieho systému, v ktorom bude pracovník postupne napredovať na vyššie pozície, čím bude mať možnosť vypracovávať úlohy za väčšie odmeny. Riešením je tiež vytvorenie zaujímavých funkcionalít, ktoré spestria pracovníkovi používanie aplikácie, a umožnia mu získať odmenu aj za elementárne činnosti. napr. získanie nového člena, prihlásenie cez sociálnu sieť, a iné.

Neschopnosť validovať splnenie úloh.

Typ rizika: Špecifické, Technické, Kvalita

Pravdepodobnosť: Vysoká

Popis rizika: V systéme poskytujeme viacero typov úloh, tieto typy je možné aj

rozšíriť o ďalšie typy. Pri každej úlohe je potrebné validovať, či pracovník splnil základné požiadavky pre vypracovanie úlohy. Ak nevieme skontrolovať a verifikovať údaje zadané pracovníkom, predstavuje to vážne riziko.

Škody a dôsledky rizika: Nespokojnosť zadávateľov. Vypracované úlohy nevyhovujú podmienkam vypracovania. Odmena za úlohu bola vyplatená neprávom, práca nebola dostatočná. Narastajúca nedôveryhodnosť aplikácie. Neschopnosť ďalej prevádzkovať aplikáciu. Úbytok používateľov(zadávateľov) a straty v ziskoch.

Riešenie rizika: Na to, aby sme vedeli úlohy správne validovať, musíme mať detailne dekomponovaný každý typ úlohy, a zachytiť podstatu pri vypracovávaní úlohy. Je potrebné, aby sme vytvorili dostačujúce algoritmy, ktoré budú spoľahlivo validovať základné splnenie požiadaviek pri vypracovaní úloh pracovníkmi.

Nekonzistentnosť pri zadávaní úloh.

Typ rizika: Špecifické, Technické, Kvalita

Pravdepodobnosť: Stredná

Popis rizika: Každá úloha má pri vytváraní svoje špecifické rozhranie, podľa svojho typu. Ak sa zadávateľ pokúsi vložiť nekompatibilné údaje do systému, nastane problém, a aplikácia sa môže správať neočakávane.

Škody a dôsledky rizika: Nespokojnosť zadávateľov. Narušená integrita aplikácie. Narastajúca nedôveryhodnosť aplikácie. Úbytok používateľov(zadávateľov) a straty v ziskoch.

Riešenie rizika: Pre vyriešenie problému nekonzistentnosti údajov je potrebné presne špecifikovať formát údajov prijateľných aplikáciou. Dôležité je tiež ošetrovanie všetkých možných chybových vstupov, ktoré môžu pri zadávaní úlohy nastať. Je potrebné identifikovať všetky prípady, v ktorých môže nastať narušenie integrity aplikácie a tieto prípady ošetriť.

Zhrnutie rizík

Názov rizika	Typ rizika	Pravdepodobnosť	Stručné riešenie
Redundantná práca	Generické	Vysoká	Stretnutia vývojárov
Nerealistické rozvrhy a rozpočty	Generické	Stredná	Agilný vývoj, Nástroj pre projektové plánovanie, Beta verzie
Zlý návrh aplikácie	Generické	Stredná	Modulárnosť, Skúsený architekt
Nevyhovujúca funkcionality	Generické	Vysoká	Prieskumy a spätná väzba z aplikácie, Časté aktualizácie, Rozdelenie aplikácie.
Nevyhovujúce UI	Generické	Stredná	UI na základe scenárov použitia
Zmeny požiadaviek	Generické	Nízka	Inkrementálny vývoj, Testovanie, Beta testy
Pochybenie zdroju dát	Špecifické	Stredná	Viac zdrojov a prekrývanie, Komunikácia
Nedostatky výkonu	Generické	Vysoká	Modulárnosť aplikácie, Benchmarky, Rôzne režimy aplikácie, Zátťažové testy
Nedostatok zadávateľov úloh	Špecifické	Vysoká	Motivácia zadávateľov, Reklama, Prezentácia
Nedostatok záujmu pracovníkov	Špecifické	Stredná	Motivácia pracovníkov, Odmeny, Sociálne siete
Neschopnosť validovať splnenie úloh	Špecifické	Vysoká	Dekompozícia, Kvalitné algoritmy
Nekonzistentnosť pri zadávaní úloh	Špecifické	Stredná	Špecifikácia formátov, Ošetrenie chýb

6.1 Zmeny v letnom semestri

Riziká ohrozujúce výsledný produkt detegované v zimnom semestri ostali. Zmenila sa ale ich pravdepodobnosť výskytu nakoľko sa časom zmenili naše znalosti a zručnosti.

Názov rizika	Typ rizika	Pravdepodobnosť LS	Pravdepodobnosť ZS
Redundantná práca	Generické	Vysoká	Stredná
Nerealistické rozvrhy a rozpočty	Generické	Stredná	Nízka
Zlý návrh aplikácie	Generické	Stredná	Nízka
Vypracovanie nevyhovujúcej funkcionality	Generické	Vysoká	Stredná
Vytvorenie nevyhovujúceho používateľského rozhrania	Generické	Stredná	Stredná
Neustále zmeny požiadaviek	Generické	Nízka	Nízka
Nedostatky výkonu v reálnom čase	Generické	Vysoká	Stredná
Nedostatok zadávateľov úloh	Špecifické	Vysoká	Vysoká
Nedostatok záujmu pracovníkov	Špecifické	Stredná	Stredná
Neschopnosť validovať splnenie úloh.	Špecifické	Vysoká	Vysoká
Nekonzistentnosť pri zadávaní úloh.	Špecifické	Stredná	Stredná

Redundantná práca

Zmiernenie pravdepodobnosti rizika spôsobili nadobudnuté skúsenosti z letného semestra kedy sme sa dobre oboznámili z novými technológiami a osvojili sme si ich. Okrem skúseností nám pomohla lepšia tímová spolupráca medzi členmi tímu. Pri zadávaní nových úloh sme vedeli lepšie odhadnúť znalosti jednotlivých členov a tím zmierniť riziko a zvýšiť efektívnosť.

Nerealistické rozvrhy a rozpočty

Pri zmiernení pravdepodobnosti rizika nám pomohli tiež znalosti schopností jednotlivých členov tímu. Hlavnou príčinou bola manipulácia s úlohami v pláne. Jednotlivé úlohy v pláne boli rozdelené na menšie pod-úlohy, pri ktorých sa nám ľahšie odhadoval čas trvania. Pri pod-úlohách sme často narazili aj na problémy a prekážky, ktoré z pohľadu úlohy neboli známe. Rozdelenie úloh malo aj psychologický charakter, kedy člen tímu mal dojem zväčšenia efektivity jeho práce.

Zlý návrh aplikácie

Zlepšenie schopností návrhu aplikácie sa značne zlepšilo. Poučenie sa na chybách a lepšie porozumenie pravidiel vývoja v použitých technológiách prispelo k tomu, že sa pravdepodobnosť rizika znížila zo strednej na nízku. Hlavný problém z ktorým

sme museli v tom to semestri riešiť bol opravenie chybných návrhov spôsobených v letnom semestri. Ďalšie používanie takto zle navrhutej aplikácie by pravdepodobnosť výskytu rizika nezmiernil. Preto sme museli previesť komplexnú úpravu návrhu a zmeniť ho. Tento krok nás stál veľa času, ale z pohľadu do budúcnosti ušetril problémy a námahu spojenú s ich odstraňovaním. Zmena návrhu v neskoršom čase by nás stála omnoho viac času ako keď sme ju urobili na začiatku semestra.

Vypracovanie nevyhovujúcej funkcionality

K zníženiu pravdepodobnosti tohto rizika nám pomohlo hlavne to, že po letnom semestri sme mali hotový prototyp, ktorý sme mohli ukazovať cieľovým používateľom. Následne sme dostávali spätnú väzbu, ktorú sme mohli prehodnocovať a zapracovávať do nášho plánu.

Vytvorenie nevyhovujúceho používateľského rozhrania

Po vytvorení prototypu a prezentácií cieľovým používateľom sme si vedeli rýchlo overiť vlastnosti používateľského rozhrania. Konkrétne testovanie osôb mimo náš tím, ktoré nemali s aplikáciou skúsenosti nám priniesli informácie na ktoré sme mohli reagovať.

Neustále zmeny požiadaviek

Vďaka dobrej taktike navrhutej už v zimnom semestri, kedy vydávaním menších verzii sme dokázali flexibilne reagovať na zmeny požiadaviek sa pravdepodobnosť nezhoršila. Požiadavky sa stále menia a je to spôsobené hlavne spájaním častí do celého projektu kedy sa často pôvodné požiadavky menia. Vďaka overení požiadaviek na častiach projektu sa tento problém nevyskytuje často.

Nedostatky výkonu v reálnom čase

Vplyvom prepracovania návrhu aplikácie, ktorý sme na začiatku semestra vytvorili a skúsenostiam s technológiami sme schopný vyhýbať sa tomuto riziku. Pri návrhu zohľadňujeme šetrenie zdrojov a to nielen výkonových ale aj dátových. Darí sa nám efektívnejšie využívať vlastnosti dobrého programovania v použitých technológiách. Na zmiernenie tohto rizika vplyva aj riziko Zlý návrh aplikácie. Pri jeho zlepšení sa automaticky zlepšil aj výkon v reálnom čase našej aplikácie.

Pravdepodobnosť špecifických rizík v našej aplikácii sa nám nepodarilo zlepšiť. Má to za následok hlavne to, že aplikácia nie je nasadená a reálne používaná. Pomocou testov prototypu na pár nezávislých používateľoch nevieme získať dostatočné množstvo informácií, ktoré by sme mohli brať ako validné. Rovnako nikto z tímu nemá potrebné skúsenosti s podobnými rizikami. Napriek tomu sme sa pokúsili skontaktovať s odborníkmi v oblasti manažmentu a získať viac informácií, ktoré by nám pomohli pri vyhnutí sa týmto rizikám. Na základe týchto informácií sme urobili zmeny v pláne. Výsledky a úspešnosť sa ale dajú reálne vyhodnotiť až po nasadení do reálnej prevádzky.

Zhodnotenie

Manažment rizík sme mali veľmi dobre navrhnutý čo sa ukázalo aj na neskoršom

zmiernení pravdepodobnosti výskytu u niektorých rizík. Na začiatku sme mali viaceré problémy, avšak členovia tímu si rýchlo osvojili postupy, ktoré sme v zimnom semestri navrhli a v letnom semestri ešte doplnili o informácie získané z testovania prototypy používateľmi a zlepšenými znalosťami s používaných technológií.

7 Manažment monitorovania projektu

Autor: Bc. Miroslav Molnár

Monitorovanie projektu predstavuje neustále kontrolovanie aktuálneho stavu projektu voči plánovanému cieľu. Pre presnejšie monitorovanie sa využíva projektový plán stanovený na začiatku projektu. Faktory vplyvajúce na monitorovanie sú čas, náklady, kvalita a riziká. Hlavná charakteristika týchto faktorov je, že sa navzájom ovplyvňujú. Ak sa kvalita zvýši, predĺži sa spotrebovaný čas a náklady na projekte. Naopak sa znížia možné riziká nekvalitného návrhu. Z toho dôvodu je rýchlosť vývoja dynamická. Monitorovanie nám prináša informácie o aktuálnom stave a tým umožňuje prispôbiť postup vývoja k úspešnému dosiahnutiu cieľa projektu. Zodpovednosť za monitorovanie projektu má manažér monitorovania spolu s vedúcim tímom. Pre získanie kvalitných a aktuálnych výsledkov monitorovania je potrebné, aby všetci členovia tímu dodržiavali základné postupy pri tvorbe vývoja a vypracovávaní úloh popísané bližšie v metodike „Metodika – Zaznamenávanie práce a sledovanie postupu“.

7.1 Navrhovaný prístup

Kontrola aktuálneho stavu projektu

Kontrola stavu projektu počas jeho vývoja a overovanie dokončenia a aktuálneho stavu úloh.

stav úloh - vyhodnotenie aktuálneho výsledku úlohy a zistenie jej správnosti s požiadavkami zadanými pri jej vytvorení.

spotreba času - kontrola spotrebovaného času na úkoly u jednotlivcov ale aj u celého tímu.

úprava harmonogramu - riadenie zmien spôsobených neočakávanými problémami počas vývoja.

projektový plán - pri výskyte problémov treba upraviť a prispôbiť k úspešnému dosiahnutiu cieľa projektu. Problémy vplyvajúce na zmenu projektového plánu:

- výskyt problému, ktorý nebol detekovaný pri tvorbe projektového plánu a teraz ho treba do neho zaradiť,
- zmena zamerania člena tímu na inú oblasť. Schopnosti člena tímu sa viac hodia na inú problematiku, alebo sa člen tímu špecializoval na inú problematiku. Takéto preorganizovanie zdrojov je veľmi nebezpečné, ale ak je dobre premyslené, prinesie efektívne a kvalitné výsledky pri riešení úloh. Manažér monitorovania je často prvý, ktorý má dostatok informácií na to, aby vedel odporučiť vedúcemu tímu správnu preorganizáciu zodpovednosti členov.

Typy monitorovania

Monitorovanie projektu delíme podľa stavu dokončenia projektu na:

- priebežné - monitorovanie prebieha počas vývoja projektu. Monitoruje sa aktuálny stav projektu v porovnaní s projektovým plánom. Priebežné monitorovanie prebieha neustále a vykonáva ho manažér monitorovania s hlavným vedúcim tímom. Hlavné monitorovanie činností členov tímu a stavu projektu prebieha na stretnutiach tímu po každom šprinte. Na týchto stretnutiach

sa kontrolujú výsledky jednotlivcov, ale aj celého tímu. Rozdeľujú sa ďalšie úlohy z plánu. Ak vznikli nejaké problémy, celý tím ich zhodnotí a spoločne sa dohodnú na možnostiach ich riešenia. Problémy sú následne zahrnuté do celkového plánu projektu,

- záverečné - zahŕňa zhodnotenie priebehu vývoja projektu. Monitorovanie zhodnotí prácu jednotlivcov a celého tímu. Detegujú sa chybné postupy vývoja. Problémy, ktoré vznikli behom vývoja sú prehodnotené a hľadajú sa príčiny ich vzniku. Nad aktuálnym stavom projektu sa urobí kompletný monitoring. Prejde sa zoznam úloh, ktoré neboli uskutočnené a prehodnotí sa ich význam. Naplánuje sa nový šprint do ktorého sa nevyriešené úlohy zahrnú,
- následné - po nasadení projektu monitorovanie pokračuje. Aktuálny stav projektu sa porovnáva s novým plánom zhotoveným pred nasadením projektu. Monitorovacie techniky sa nemenia. Mení sa dynamickejší plán, ktorý sa tvorí z prevažne úloh typu „chyba“, alebo rozšírenia funkcionality projektu. Monitorovanie tejto časti je zložitejšie. Dôvodom je vysoká priorita chýb, ktoré musia byť riešené hneď po ich detekcii v projekte. Vysokú priority úloh typu „chyba“ vytvára fakt, že systém je používaný používateľmi, zatial čo pri vývoji s projektom pracovali iba členovia tímu.

7.2 Aplikácia prístupu

Pre monitorovanie práce jednotlivcov a vyhodnocovanie práce celého tímu je navrhnutý prístup, ktorý sa zaviedol aj do praxe. Pre rozdeľovanie, pridelenie a monitorovanie stavu pridelených úloh členom tímu sme použili systém JIRA.

7.2.1 Dokumentovanie práce

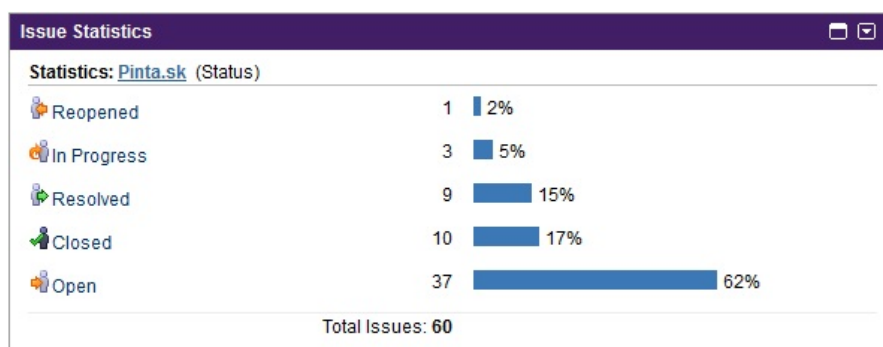
Každý člen tímu je povinný zdokumentovať v systéme JIRA prácu, ktorú vykoná na úlohe. Členovia tímu sú na základe funkcionality ktorú im JIRA prináša schopní monitorovať sami seba. Dokumentácia je potrebná pri odhalovaní zdroja chýb, modifikácii programu. Je treba zachytiť všetky kľúčové fakty, ktoré budú v budúcnosti zúročené pri modifikácii projektu. Dokumentácia je veľmi dobrým zdrojom monitorovania práce členov tímu. Pri väčšom množstve dokumentov je možné k úlohe pripojiť súbory, odkazy a komentáre. Komentáre k úlohe môžu pridávať aj iní členovia tímu. Všetky tieto informácie evidujú nielen aktuálny stav úlohy, ale aj jej priebeh vypracovania.

7.2.2 Vyhodnotenie stavu projektu

Stav projektu je možné vyhodnocovať aj priebežne z aktuálneho stavu zaznamenanom v systéme JIRA. Jednotlivé informácie pomáhajú nielen manažérovi monitorovania a hlavnému vedúcemu tímu, ale aj všetkým ostatným členom tímu získať aktuálny stav projektu, ale aj stav ich samých v rámci tímu. Najrýchlejšie zobrazenie a pochopenie veľkého množstva údajov je grafickým spôsobom. Pomocou grafického zobrazenia sa môže zobrazit veľké množstvo dát. Systém JIRA umožňuje členom tímu zobraziť viaceré informácie pomocou grafu.

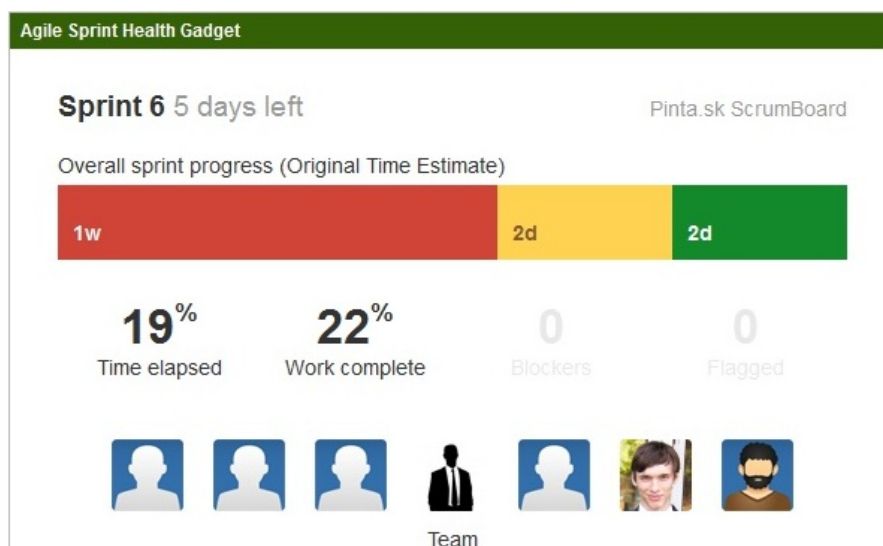
Pre zobrazenie štatistiky vytvorených úloh slúži graf zobrazený na Obrázku 1. Graf triedi úlohy do skupín podľa ich aktuálneho stavu. Manažér monitorovania z grafu deteguje aktuálny stav úloh a ich množstvo v projekte. Z grafu manažér

monitorovania neodhaduje čas potrebný na spravenie všetkých úloh, lebo úlohy majú rôznu dĺžku trvania nezohľadnenú v grafe.



Obr. 1: Stav úloh v systéme.

Stav úloh je možné monitorovať aj v rôznych časových intervaloch. V našom projekte sa tím stretáva každý týždeň a preto máme nastavený sedem dňový šprint. Dole nižšie je zobrazený graficky aktuálny stav úloh rozdelených v jednom šprinte. Vidíme koľko z úloh je spravených a koľko sa práve robí a nakoľko sa ešte nezačalo robiť. Manažér monitorovania má takto aktuálny stav úloh kedykoľvek potrebuje. Bližšie Obrázok 2.



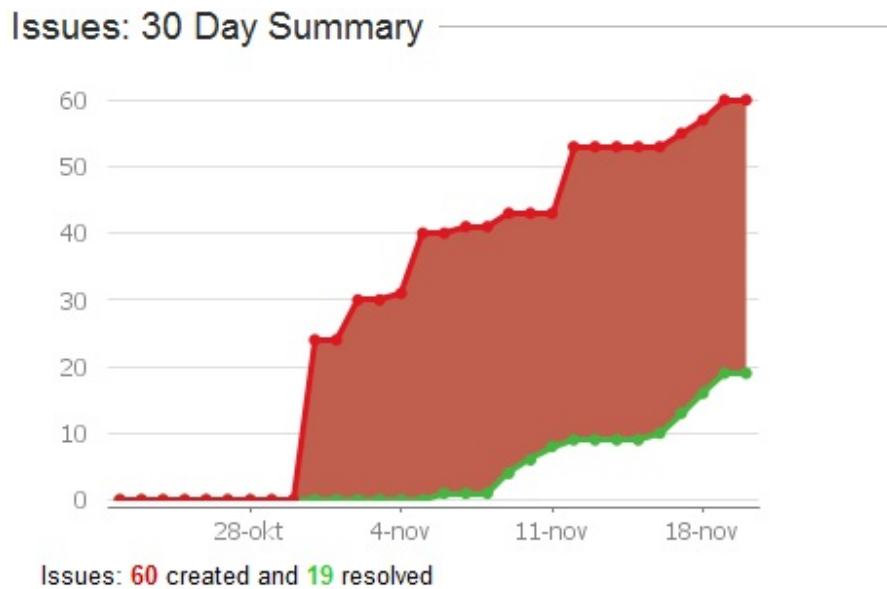
Obr. 2: Stav úloh počas jedného šprintu.

Je však dôležité aby všetci členovia tímu dodržiavali pravidlá práce so systémom JIRA, ktoré sú bližšie popísané v „Metodika – Zaznamenávanie práce a sledovanie postupu“.

Pre monitorovanie celkovej správy úloh nám slúži graf na Obrázku 3, ktorý popisuje aktuálny stav úloh v systéme. Na tomto grafe je jasne vidieť novovzniknuté úlohy. Na začiatku sa pre lepšiu prehľadnosť a orientáciu v monitorovaní aktuálneho stavu zvolia iba hlavné úlohy. Tie sa následne delia na podúlohy.

7.3 Zmeny v letnom semestri

Bol detegovaný nový problém vplyvajúci na zmenu projektového plánu. Týmto problémom je spätná väzba od používateľov, ktorí testovali našu aplikáciu a určujú nové.



Obr. 3: Stav úloh v projekte

Monitorovanie v letnom semestri zlepšilo dodržiavanie postupov práce členov tímu vo všetkých ostatných metodikách. Hlavným zlepšením bolo rozdeľovanie úloh na pod-úlohy. Tie sa následne podľa potreby mohli ďalej deliť. Toto sa opakuje až pokiaľ nezískame úlohu dostatočne jasnú. V zimnom semestri sa úlohy delili, ale toto delenie nebolo dostačujúce. Často úloha skrývala problémy, ktoré sa jej rozdelením mohli detegovať a riešiť. Rozdelenie úloh prebieha na stretnutiach za účasti všetkých členov tímu. Tak sa docieli správneho rozdelenia a odhalenia možných problémov. Takto rozdelené úlohy sa ľahšie monitorujú.

Monitorovanie sa zlepšilo aj kvôli lepšiemu a stabilnejšiemu projektovému plánu prepracovanom na začiatku letného semestra. Problémom ale bolo, že projektový plán bol navrhnutý na základe informácií, ktoré sme mali lev v rámci tímu. Po vytvorení prototypu a testovaní na používateľoch, sa zistili nedostatky v návrhu a musela sa aplikácia prerobiť, následne sa upravoval aj projektový plán.

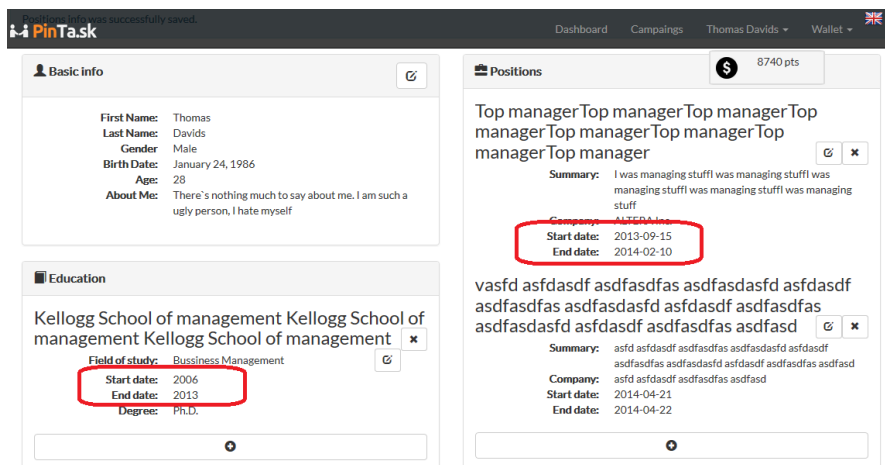
Monitorovanie funkčnosti aplikácie

Je nutné aplikáciu testovať na používateľoch, pomocou ktorých získame potrebné informácie o funkcionalite projektu. Tiež sa overujú požiadavky na aplikáciu. V prípade vzniku nových požiadaviek je nutné ich prehodnotiť s členmi tímu a následne zaradiť do projektového plánu s prioritou. Je dôležité aby testovanie bolo vykonávané na viacerých typoch používateľoch s rôznym vekom, schopnosťou práce na počítači a znalosťou našej aplikácie.

Monitorovanie chýb v systéme.

Pred nasadením aplikácie do prevádzky sme musela dôkladne otestovať pred výskytom chýb. Testovanie prináša z hľadiska monitorovania veľa informácií o aktuálnom stave aplikácie. Nevedomosť o existencii chýb môže skresľovať aktuálny stav projektu a tým pádom aj monitorovanie je nepresné. Je treba dôkladne testovať aplikáciu na viacerých typoch prehliadačoch. Pri nájdení chyby sa musí overiť v zozname chýb, či už daná chyba neexistuje, aby nevznikala duplicita. Zápis chýb je striktné zapísaný do Altassian Jira, kde je nutné výstižne pomenovať chybu a

dôkladne zrozumiteľný ju popísať. V niektorých prípadoch je lepšie k chybe pridať aj obrázok chyby. Chyba sa môže na obrázku zvýrazniť pomocou červenej farby, ako môžeme vidieť na Obrázku 4.



Obr. 4: Zvýraznenie chýb

Pri zápise chyby bolo nutné určiť aj jej prioritu. Chyby po zapísaní nie sú hneď riešené. Je nutné aby sa overila ich existencia ostatnými členmi tímu. Mohlo sa stať, že chyba bola už odstránená, ale člen tímu, ktorý ju našiel mal staršiu verziu aplikácie. Rovnako sa mohlo stať, že chyba je iba lokálna a je spôsobená zlou konfiguráciou systému programátora. Po odstránení chýb, pri ktorých nebola overená ich existencia sa ostatným chybám prehodnotí a podľa potreby upraví ich priorita. Následne sa upraví aj projektový plán. Väčšina nájdených chýb má najvyššiu prioritu a tak je hneď pridelená členom tímu podľa ich skúseností so zdrojovým kódom v mieste výskytu chyby.

Zhodnotenie

Monitorovanie bolo veľmi ťažké počas oboch semestrov. V prvom semestri sťažovalo monitorovanie zle navrhnutý a tím pádom stále sa meniaci projektový plán, spolu s nedržiavaním pravidiel definovaných v jednotlivých metodikách u členov tímu. V letnom semestri projektový plán bol stabilný a dobre navrhnutý. Rovnako členovia tímu si osvojili pravidlá popísaných v metodikách. Naša aplikácia obsahovala chyby, ktoré boli odhalené až po testovaní aplikácie. Po odhalení týchto chýb sa projektový plán často upravoval a monitorovanie bolo veľmi komplikované. Pri testovaní aplikácie s používateľmi sa detegovali nové požiadavky na aplikáciu, ktoré tiež vplývali na zmenu projektového plánu. Napriek tomu sme boli schopný nájdené chyby dobre monitorovať a rýchlo odstrániť a tím pádom splniť ciele stanovené v projektovom pláne.

8 Manažment komunikácie a ľudských zdrojov

Autor: Bc. Martin Toma

V tejto kapitole popisujeme procesy spojené s riadením komunikácie a s manažmentom ľudských zdrojov. Vedúci tímu má teda za úlohu pracovať ako spojka medzi produktovým vlastníkom (zadávatelom projektu) a zvyškom tímu. Jeho úlohou je rozdeľovať úlohy a agilne reagovať na vzniknuté problémy či požiadavky z oboch strán.

Hlavným pracovným nástrojom pri týchto procesoch je JIRA, softvér na evidenciu a sledovanie úloh v projektoch (v našom prípade vo verzii 5.1) od spoločnosti Atlassian. Okrem sledovania úloh ponúka tento nástroj možnosti ako sledovať postup práce na konkrétnom šprinte a zároveň sa dá využiť ako miesto na komunikáciu špecificky určenú pre konkrétnu pridelenú úlohu.

Vedúci tímu sleduje priebeh aktuálneho šprintu, odhadovaný zostávajúci čas prác a iné metriky, ktoré mu umožňujú vytvoriť si obraz o fungovaní tímu. Okrem toho, spostredkováva komunikáciu medzi členmi tímu a snaží sa aby bol celý tím vrátane zadávateľa projektu v obraze. Vedúci tímu je zároveň vedúcim projektu v kontexte nástroja JIRA. Jeho úlohy súvisiace s týmto nástrojom sú:

- Udržiavanie a aktualizácia projektových metadát, verzií a komponentov,
- správa, evidovanie a sledovanie všetkých úloh, primárne však tých ktoré nepatria pod konkrétny komponent alebo neboli správne priradené,
- náprava chybné špecifikovaných atribútov a zabezpečenie konzistencie v rámci projektu.

Na to aby mohol vedúci tímu túto prácu vykonávať je potrebná aktivita aj ostatných členov tímu. Čo sa týka nástroja JIRA, medzi ich úlohy patrí primárne:

- Participácia na identifikovaní všetkých chýb v zdrojovom kóde a ich zaevidovanie v systéme,
- participácia na identifikovaní nových funkcionalít a ich evidencia v systéme,
- správa, evidovanie a sledovanie všetkých úloh ktoré patria pod môj komponent,
- aktualizácia stavu (aj zostávajúceho času) mne pridelených úloh v systéme (špeciálne pri úlohách na aktuálny šprint (Metodika SCRUM)).

Vedúci tímu má povinnosť byť dostupný na komunikačných kanáloch ktoré tím najčastejšie používa. V našom prípade ide o Google Hangouts, e-mailovú komunikáciu a nástroj JIRA. Vedúci tímu má prehľad o tom, kto čo robil, robí, a v budúcnosti plánuje robiť. Má mať prehľad o tom, n koho je dobré sa obrátiť v prípade špecifickejších otázok. Vedúci tímu zároveň plní úlohu SCRUM master-a. Nie je však klasickou autoritou. Nevykonáva rozhodnutia samostatne ale len po dohode väčšiny tímu. Jeho úlohou je jednotlivým členom tímu ponúknuť podporu a motivovať ich k lepším výkonom.

Vedúci tímu by sa mal zúčastniť na všetkých tímových formálnych aj neformálnych stretnutiach. Každé dôležité rozhodnutie ohľadom akéhokolvek procesu spojeného s projektom by malo byť prediskutované s celým tímom, minimálne však s vedúcim tímu ktorý ďalej informuje zainteresované osoby.

Všetky vyššie spomenuté zodpovednosti a úlohy sa v súčasnosti vedúci tímu snaží zabezpečiť avšak vzhľadom na náročnosť a prvú skúsenosť s danou pozíciou nie sú všetky procesy ideálne. Dôležité však je, že rovnako ako všetky procesy, tak aj proces riadenia sa kontinuálne zlepšuje a tým sa zvyšuje aj celková kvalita tímu a jeho výsledkov.

Vedúci tímu má v našom prípade na starosti aj prezentáciu tímu na verejnosti. Má za úlohu pracovať na značke tímu a budovať jeho identitu. Mal by byť rovnako ako zadávateľ projektu v kontakte s externým prostredím. Prenáša ťarchu na seba a tým ponúka ostatným členom tímu viac možností na nerušenú a produktívnu prácu na projekte. Samozrejme aj on sám sa zúčastňuje vývojových aktivít. Jeho prínos však nespočíva len v počte implementovaných funkcionalít. Aj keď je v prípade tohto tímového projektu funkcia vedúceho tímu obmedzenejšia ako v reálnom prostredí, stále však plní dôležitú úlohu.

Z hľadiska riadenia ľudských zdrojov je vedúci tímu zodpovedný za to, aby v prípade preťaženia jedného člena tímu dokázal zabezpečiť rovnováhu v tíme. Po dohode so zadávateľom projektu konzultuje zásahy do aktuálneho šprintu. Či už ide o pridanie alebo naopak odstránenie niektorých úloh. Musí však zabezpečiť to, aby sa ku každému členovi tímu pristupovalo rovnako. Nesmie sa stať že niektorí členovia tímu neustále nestíhajú dokončiť dohodnutú prácu. V takomto nepríjemnom prípade sa opäť po porade so zadávateľom projektu rozhodnú ako budú postupovať.

Pri týchto úlohách je hlavným pracovným nástrojom vedúceho tímu už spomínaný nástroj JIRA. Medzi jeho možnosti patrí už spomínané zaznamenávanie práce. Generujú sa tzv. „burndown“ grafy, ktoré ukazujú reálny vývoj prác na projekte. Na to, aby tieto grafy mali výpovednú hodnotu je nutné, aby každý člen tímu zaznamenával každú, aj nedokončenú prácu na jemu pridelenú úlohu. Proces zaznamenania práce v nástroji JIRA je popísaný podrobne v nasledujúcej podkapitole.

8.1 Metodika – Zaznamenávanie práce a sledovanie postupu

Ciel: Dokončil som časť práce na mne pridenej úlohe. Chcem do systému založovať ako dlho som na nej pracoval z dôvodu sledovania postupu implementácie.

Vstup: Mne pridelená issue v nástroji JIRA, v stave vykonávania (ang. „In progress“).]

Výstup: Uzavretá issue so zaznamenaným časom práce alebo aktualizovaná issue o hotovú časť. Aktuálny graf postupu (ang. „Burn-down chart“).

Na logovanie práce budeme používať funkcionality „log work“. Nebudeme teda používať postup pri ktorom JIRA automaticky počíta čas od zmeny statusu issue z prebiehajúcej na hotovú.

Na stránke danej issue v JIRE cez záložku „More actions“ → „Log Work“. Zobrazí sa formulár zobrazený na Obrázku 5. Ten vyplníme podľa Tabuľky 1.

Správnym a pravidleným zaznamenávaním práce použitím tohto dohodnutého postupu sme schopní získať reálny obraz o priebehu šprintu. Na základe týchto dát sa generuje spomínaný „burndown“ graf, ktorého príklad je zobrazený na Obrázku 6. V tomto prípade ide o veľmi dobrý a pomerne priamočiary graf skúseného SCRUM

Log Work

Time Spent * (eg. 3w 4d 12h) ?
An estimate of how much time you have spent working.

Date Started * 📅

Remaining Estimate Adjust automatically
the estimate will be reduced by the amount of work done, but never below 0.

Use existing estimate of 1 day

Set to (eg. 3w 4d 12h)

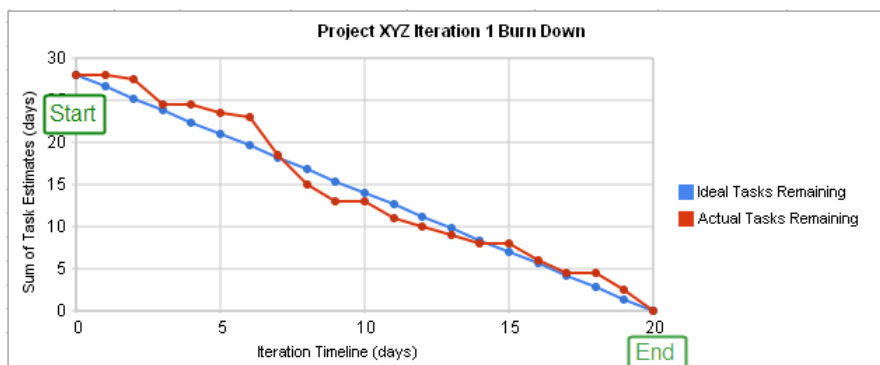
Reduce by (eg. 3w 4d 12h)

Work Description

🔍 ? 🔒 Viewable by All Users

Obr. 5: Log Work

tímu. V našom prípade však vďaka neúplnosti dát a nie úplne zvládnutej SCRUM metodike vyzerajú tieto grafy značne horšie. V najbližších šprintoch však už dbáme na správne postupy a hlavne zodpovedné zaznamenávanie práce čo značne zlepší použiteľnosť týchto dát.



Obr. 6: Burndown graf

V prípade, že budú dáta správne tak už nič nebráni tímu v rýchlejšom napredovaní a vývoji, napríklad aj v oblasti kvality a presnosti odhadovania časovej náročnosti úloh. Nakoľko to je jeden z hlavných dôvodov problémov, objavených použitím tohto grafu. Vedúci tímu je teda zodpovedný za to aby sa výsledky a zistenia aplikovali tak, aby zlepšili proces akým tím funguje v nasledujúcej iterácii.

Vedúci tímu však samozrejme nie je jediný kto spomenuté veci sleduje. V prípade „burndown“ grafov spolupracuje primárne s manažérom plánovania. S manažérom rizík, ktorý je súčasne aj jeho zástupcom, spolupracuje skoro pri všetkých procesoch. Samozrejme je jeho spolupráca so zadávateľom projektu. Vedúci tímu má

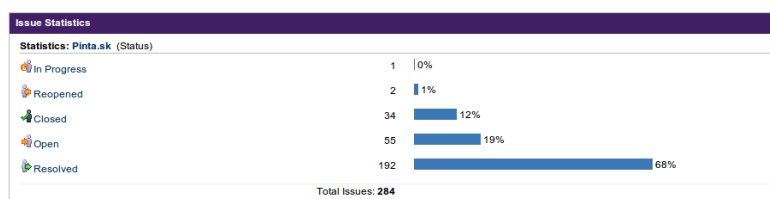
Tab. 1: Vyplnenie Log Work-u

Atribút	Ako vyplniť
Time spent (uplynutý čas)	Čo najpresnejšie zadefinovaný čas strávený pri práci na danej issue. Ide o celkový čas vrátane analýzy problému, hľadania riešenia, a implementácie. Príklad: „4h 30m“.
Date started (dátum začatia práce)	Dátum a čas kedy sme začali na danej issue pracovať (Môže sa odlišovať od času a dátumu pridelenia úlohy).
Remaining estimate (Odhad ešte potrebného času)	Odhadovaný zostávajúci čas. Vždy treba vyplniť. Na základe týchto údajov sa kontroluje stav šprintu. Skreslené odhady skresľujú celý obraz o šprinte.
Work description (Popis vykonanej práce)	Komentár k vykonanej práci. Súvisí priamo s konkrétnym issue, ktoré logujeme. V prípade, že úloha ešte nebola dokončená, obsahuje popis toho, čo už bolo implementované a čo je ešte potrebné dokončiť.

teda za úlohu sledovať všetky procesy pričom dbá na to, aby boli všetci členovia tímu spokojní, aby mali všetky potrebné informácie, a aby celý tím fungoval efektívne a aby dosahoval vytýčené ciele.

8.2 Zmeny v letnom semestri

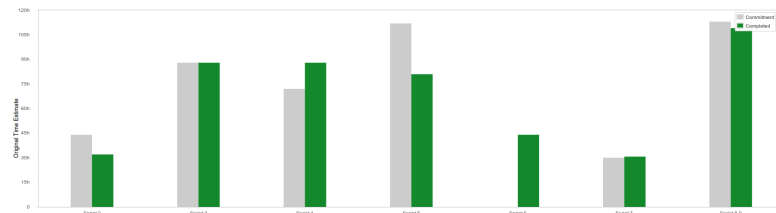
Úlohy stanovené v zimnom semestri som sa snažil plniť. Zúčastnil som sa takmer každého stretnutia a snažil som sa všetko organizovať spravodlivo. Okrem toho som po každom šprinte zadal do systému JIRA úlohy pre každého člena na nasledujúci šprint. Celkový obraz stavu všetkých úloh v systéme ku dnešnému dňu je zobrazený na obrázku nižšie.



Obr. 7:

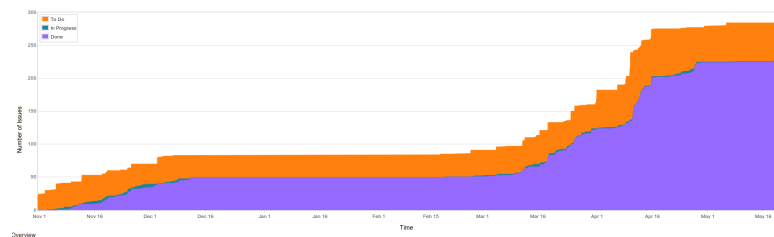
Všetky dôležité rozhodnutia som konzultoval s vedúcim tímu ale vždy som sa snažil informovať o tom aj všetkých členov tímu. Mojim cieľom bolo spraviť s týmu čo najlepšie fungujúcu skupinu ľudí kde každý robí primárne to čo ho baví a v čom je dobrý a zároveň tým pomáha tímu. Na obrázku nižšie vidíme graf rýchlosti v letnom semestri. Ku koncu vidíme mierne spomalenie, to však bolo spôsobené inými úlohami (dokumentácia, IIT.SRC) a podobne, ktoré sme v tomto systéme nevidovali. Celkovo sme však dosahovali asi o tretinu vyššie rýchlosť ako v prípade zimného semestra.

Okrem toho sme v letnom semestri celkovo vytvorili a vyriešili omnoho viac úloh a priniesli sme aj vďaka zvýšenej rýchlosti oveľa viac pridanej hodnoty. Na obrázku nižšie vidíme celkový graf vytvorených a vyriešených úloh v systéme JIRA za



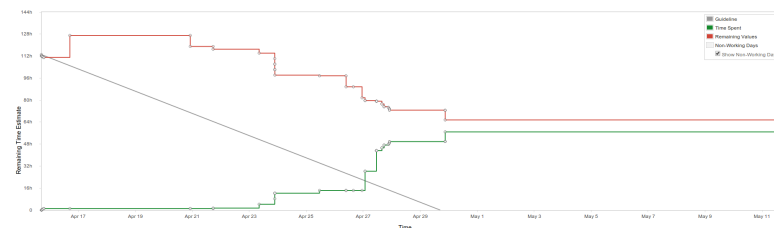
Obr. 8:

celý rok (zimný aj letný semester). Ako môžeme vidieť, v druhej polovici grafu prišlo k výraznému vzostupu vytvorených ale aj vyriešených úloh. Samozrejme že



Obr. 9:

počas prác na jednotlivých šprintoch som sledoval iné metriky a graf, konkrétne burndown chart, ktorý vidíme na obrázku nižšie a ktorý ponúka aktuálnejší obraz na vec. V prípade, že veci nešli podľa plánu som sa snažil konzultovať dôvody s tými ktorých sa to najviac týkalo.



Obr. 10: Stĺpcový graf

Na koniec môžem povedať, že som s celkovou prácou tímu spokojný. Niektorí členovia tímu ma mierne sklamali a iní zasa príjemne prekvapili. Myslím si, že ako tím sme mali konštruktívne debaty a snažili sme sa aj za ceny miernych ústupkov robiť to, čo bolo najlepšie pre tím.

Z pohľadu nášho výstupu si myslím, že vzhľadom na to, že sme sa vybrali do nám dovedy neznámych technológií a veľa z nás aj prvý krát pracovalo vo väčšom tíme, sme projekt zvládli. Samozrejme, že výstup mohol byť aj lepší. Myslíme si však, že poskytuje dostatočné možnosti a funkcionality na to, aby sme preukázali čo sme za tieto dva semestre urobili.

9 Manažment dokumentácie

Autor: Bc. Marek Lenčes

9.1 Roly účastníkov

Rola	Povinnosti
Produktový vlastník	<ul style="list-style-type: none"> • čítanie a kontrola dokumentácie • návrhy na zlepšenie dokumentácie • hodnotenie kvality dokumentácie
Manažér dokumentácie	<ul style="list-style-type: none"> • kontrola konzistentnosti a úplnosti dokumentácie a zápisníc • kontrola gramatickej správnosti • úpravy formátovania dokumentov
Všetci členovia	<ul style="list-style-type: none"> • vytváranie projektovej dokumentácie • vytváranie zápisníc • komentovanie zdrojových kódov • snaha o gramatickú jednoznačnosť a bezchybnosť

9.2 Projektová dokumentácia

Dokumentovanie vytváranej aplikácie bude prebiehať priebežne v rámci každého šprintu. Každá nová funkcionálna v podobe konkrétnej feature v rámci user story bude zdokumentovaná členom tímu, ktorý mal túto funkcionálnu v danom šprinte na starosti. Každý člen tímu je povinný pravidelne aktualizovať dokumentáciu, keď vytvorí novú feature, ktorá bude funkčná a bude zlúčená s hlavnou vetvou v git repozitári.

Projektová dokumentácia sa bude nachádzať v spoločnom adresári v repozitári Google Drive, kde budú mať k nej všetci členovia tímu prístup na úpravy. Na Google Drive sa budú nachádzať nasledovné dokumenty:

- Projektová dokumentácia - verzia na úpravy,
- Projektová dokumentácia - publikovateľná, finálna verzia, opravená a správne naformátovaná,

- Zápisnice zo stretnutí - verzia na úpravy,
- Zápisnice zo stretnutí - publikovateľná, finálna verzia, opravená a správne naformátovaná
- Všetky metodiky, ktorými sa členovia tímu riadia
- UML diagramy vo formáte *.xml* dostupné na úpravy každým členom tímu
- UML diagramy vo formáte *.xml* vo finálnej verzii, a vo formáte *.jpg* vo finálnej verzii

Manažér dokumentácie je zodpovedný za publikovanie finálnych verzií jednotlivých dokumentov, jeho úlohami sú:

- oprava gramatických chýb
- úprava štruktúry a vzhľadu (L^AT_EX)
- jednotnosť v používaní pojmov
- konzistencia jazyka, prípadne definovanie neznámych pojmov
- korektné začlenenie obrázkov a tabuliek
- kontrola správnosti zápisníc
- finalizácia zápisníc
- kontrola komentárov v zdrojovom kóde

Pri vytváraní návrhu architektúry v UML (diagramy - dátový model) sme sa dohodli na modelovacom nástroji *Visual Paradigm*. Diagramy budú vytvárané a upravované všetkými členmi tímu, budú ukladané do repozitára Google Drive vo formáte *.xml*, ktorý je možné upravovať importovaním do modelovacieho nástroja. Finálna verzia diagramov bude uložená do osobitného priečinka, a okrem formátu *.xml* bude v ňom dostupná aj inštancia diagramu vo forme obrázku vo formáte *.jpg*.

9.3 Zápisnice

Tvorba zápisníc bude prebiehať vždy na stretnutí v aktuálnom šprinte, a bude tvorená vždy iným členom tímu. Zápisnica má presne stanovenú štruktúru, ktorú treba dodržať, čo bude mať na starosti zapisovateľ. Zápisnica ešte bude po stretnutí zrevidovaná zapisovateľom, prípadne doplnená a odovzdaná bude do repozitára dokumentov, kde bude ešte raz zrevidovaná manažérom dokumentácie. Ten sa postará o opravu pravopisných chýb a kontrolu dodržania stanovenej štruktúry. Následne bude zápisnica uložená na prezentačnú stránku tímu, stane sa tak maximálne do konca posledného dňa príslušného šprintu.

9.4 Dokumentácia zdrojového kódu

Komentovanie zdrojového kódu nám uľahčí orientáciu v jednotlivých častiach zdrojového kódu, a umožní aj identifikovať, kto vytvoril daný kód. Preto sa bude na začiatku každého vytvoreného súboru uvádzať kto ho vytvoril, krátky názov, a čo daný kód reprezentuje, príklad:

```

1 =begin
2 Bc. Buduci Inzinier
3 Users Controller
4 obsahuje funkcie na spravu pouzivatelov
5 =end

```

Ďalší príklad, v HTML zdrojovom kóde

```

1 <!--
2 Bc. Buduci Inzinier
3 Task Create View
4 zobrazuje pouzivatelovi formular pre vytvorenie ulohy
5 -->

```

Komentáre v rámci zdrojového kódu sú nutné iba v prípade, že kód obsahuje nejasné jazykové konštrukcie, alebo zložitejšie mechanizmy, ktoré budú ľahšie pochopiteľné pre ostatných členov tímu pomocou komentárov.

9.5 Procesy

Číslo	Názov procesu
8.1	Dokumentovanie dokončenej funkcionality
8.2	Revízia dokumentácie
8.3	Dokumentovanie stretnutí

9.5.1 Dokumentovanie dokončenej funkcionality

Vstup: Nezdokumentovaná časť práve naprogramovanej funkcionality

Výstup: Vytvorená dokumentácia pre hotovú funkcionality

Postup:

1. Člen tímu otvorí príslušný program s dokumentáciou v repozitári Google Drive
2. Člen tímu zdokumentuje vytvorenú funkcionality
3. Člen tímu skontroluje správnosť a pravdivosť opísanej funkcionality

9.5.2 Revízia dokumentácie

Vstup: Vytvorená dokumentácia alebo jej časť

Výstup: Opravená dokumentácia

Postup:

1. Manažér dokumentácie otvorí príslušný dokument s dokumentáciou
2. Manažér dokumentácie skontroluje pridanú časť dokumentácie, opraví chyby, skontroluje konzistenciu

3. Manažér dokumentácie presunie skontrolovanú časť dokumentácie do finálnej verzie dokumentu

9.5.3 Dokumentovanie stretnutí

Vstup: Témy rozoberané na stretnutí tímu

Výstup: Zápisnica z aktuálneho stretnutia

Postup:

1. Člen tímu otvorí šablónu zápisnice
2. Člen tímu zapisuje všetky potrebné údaje do zápisnice
3. Ostatní členovia tímu na záver zápisnicu skontrolujú a potvrdia
4. Člen tímu uloží zápisnicu do repozitára Google Drive a na prezentačnú stránku tímu

9.6 Zmeny v letnom semestri

Najväčšie zmeny pri písaní dokumentácie nastali pri vytváraní zápisníc z jednotlivých šprintov. Tento proces sme automatizovali vytvorením nástroja pre generovanie zápisníc prezentovaného v kapitole 3.11.5 dokumentácie k inžinierskemu dielu. Zapisovateľ už iba vpisuje detaily rozoberané na stretnutí a na konci automaticky vygeneruje hotovú zápisnicu.

Zmeny nastali aj pri vytváraní projektovej dokumentácie, pre lepšiu orientáciu v texte a jednotlivých kapitolách sme použili hlavičky na každej strane, v ktorých je vždy názov aktuálnej kapitoly, v dokumente takéhoto rozsahu to značne zjednodušuje orientáciu. Tiež sme sa rozhodli, kvôli rozsahu dokumentu zvoliť obojstranné rozloženie a tým vytvorili pocit, že čitateľ číta „knihu“, nejaký ucelený text.

Snažili sme sa dokumentovať patrične aj zdrojový kód, ak sme vytvorili funkcionality, ktorá obsahuje špeciálne implementačné mechanizmy, jej opis sa nachádza priamo v zdrojovom kóde. Pri praktizovaní Code Review v každom šprinte sme navyše pridávali pri Pull Requestoch množstvo komentárov a otázok ku kódu, ktorý sme nevedeli jednoznačne opísať a následne sme poznámky zahrnuli do zdrojového kódu.

Okrem týchto zmien nenastali žiadne špeciálne zmeny v písaní dokumentácie, každý člen tímu vytváral dokumentáciu v aktuálnom šprinte opisom svojej implementovanej funkcionality a spoločné časti sme upravovali spoločnými silami pre vytvorenie ucelenej dokumentácie k našej aplikácii.

V rámci manažmentu dokumentácie som ako manažér dokumentácie kontroloval typografickú a gramatickú úpravu celého dokumentu. Väčšina chýb sa podarila odstrániť, no stále ešte existujú nekonzistentosti v pomenovávaní tých istých vecí rôznymi názvami, najčastejšie zámena anglických slov za slovenské, napr. - task-úloha, pracovník-worker, feature-funkcionalita.

10 Manažment kvality

Autor: Bc. Miroslav Vojtuš

10.1 Konvencie v jazyku Ruby a programovom rámci RoR

S kvalitou kódu určite súvisí aj čitateľnosť kódu. Pre zabezpečenie čitateľnosti máme vypracovanú metodiku konvencií pre jazyk Ruby a programový rámec Ruby on Rails. Zvýšením čitateľnosti zároveň znižujeme riziko odchodu člena tímu, keďže dobre čitateľný kód sa aj dobre absorbuje novým človekom. Taktiež pri dodržaní spoločných postupov a konvencií daných Ruby on Rails rámcom, zabezpečíme konzistenciu kódu, a tým zredukujeme duplicitný kód, čo opäť vedie k vyššej kvalite kódu.

V ďalších kapitolách sa nachádza upresnenie(ohraničenie) konvencií daných programovým rámcom RoR a jazykom Ruby. Opísané sú konvencie písania názvov, štandardných postupov a organizácie zdrojových kódov. Účelom je zabezpečiť kvalitu kódu a jeho čitateľnosť.

10.1.1 Všeobecné

- nikdy neodovzdávajte do zdieľaného úložiska zakomentovaný zdrojový kód. V prípade, že vykonanie kódu blokujete len dočasne, osamostatnite si ho lokálne,
- minimalizujte komplexnosť kódu dekompozíciou do samostatných funkčných blokov(napr. súkromné metódy),
- opisné komentáre používajte len na komplikované (ťažko pochopiteľné) časti kódu. Aj v takomto prípade preferujte dekompozíciu a plnovýznamové názvy premenných a metód,
- inštančné premenné používajte len kde je nevyhnutné (napr. na zdieľanie informácií s pohľadom)
- na prenos informácií medzi metódami používajte výhradne predávanie pomocou parametrov
- na tvorbu kódu uprednostnite dostupné generátory. Tie zabezpečia správne vytvorenie/umiestnenie súborov so zdrojovými kódmi. Nepoužité vygenerované súbory a časti kódu odstráňte,
- vlastné knižnice umiestňujte do priečinka '`<rails_root>/lib`'
- knižnice tretích strán uprednostňujte vo forme gem-ov pred manuálnym sťahovaním zdrojových súborov. Ak je to nutné, tieto umiestnite do priečinka '`<rails_root>/lib`'

10.1.2 Písanie názvov

- všetky názvy v zdrojových kódach musia byť písané výhradne v anglickom jazyku
- musia byť krátke a dostatočne opisné na vyjadrenie účelu

- každá premenná musí spĺňať kritéria dané jazykom Ruby a zároveň formát daný pravidlami opísanými v nasledujúcej tabuľke:

oblasť	príklady	pravidlá zápisu
inštančné, globálne premenné	variable longer_variable	slová malými písmenami oddelené znakom '_' slová <u>musia</u> byť plnovýznamové regex: <code>^[a-z]+([a-z][_])?*\$</code>
lokálne premenné	local_variable i	slová začínajúce malým písmenom oddelené znakom '_' regex: <code>^[a-z]+([a-zA-Z0-9][_])?*\$</code>
konštanty	CONSTANT LONG_CONST	slová veľkými písmenami oddelené znakom '_' slová <u>musia</u> byť plnovýznamové regex: <code>^[A-Z]+([A-Z][_])?*\$</code>
triedy, moduly	Module1 Controller::One	slová začínajúce veľkými písmenami slová <u>musia</u> byť plnovýznamové regex: <code>^[A-Z]+([a-z0-9]+([A-Z]::[A-Z])?)?*\$</code>
verejné metódy	method_one method?	slová malými písmenami oddelené znakom '_' slová <u>musia</u> byť plnovýznamové regex: <code>^[a-z]+([a-z][_])?*[!]?+\$</code>
mená súborov	model_one.rb controller_name.rb view.erb	slová malými písmenami oddelené znakom '_' zakočené platnou príponou regex: <code>^[a-z]+([a-z][_])?*\.[a-z]+\$</code> <i>výnimkou sú migračné skripty, ktoré môžu obsahovať číslu na začiatku názvu</i>

Obr. 11: Pravidlá formátovania

10.1.3 Ruby konvencie

Jazyk Ruby často umožňuje zápis niektorých typických úloh rôznymi prístupmi. V nasledujúcich odsekoch sú opísané preferované postupy, ktoré zabezpečia čitateľnosť a konzistenciu zdrojových kódov. Tieto je nutné dodržiavať, pokiaľ nie je nevyhnutné konať inak, v takom prípade treba mať primerané argumenty,

- nepoužívajte metódu `.nil?` - pokiaľ to nie je explicitne nutné. Na testovanie či bol objekt inicializovaný používajte `if object` nie `if !object.nil?` ani `if object != nil`.
- používajte `unless` - Na testovanie, či objekt nie je inicializovaný používajte `unless object` namiesto `if !object`,
- používajte jednoriadkové podmienené vykonanie:

```

1 if element.processable?
2   element.process
3   puts "element #{element} processed"
4 end
5
6 element.process if element.processable?
```

- `do end` - v prípade viacriadkových blokov

```

1 elements.each do |element|
2   element.process
3   puts "object #{element} processed"
4 end
```

- `{}` - v prípade jednoriadkových blokov

```
1 elements.each {|element| element.process}
```

- *zátvorky používajte len ak je to nutné* - zvyčajne len pri vnorenom volaní funkcií. Jednoduchý zoznam parametrov nemusí byť ohraničený zátvorkami:

```
1 return do_a_job(param, get_job(args, :console))
2 do_a_job param, :exec
```

10.1.4 Migrácie

- pre každý zásah do databázovej schémy píšete migračný kód
- na vytvorenie migrácie uprednostňujte dostupné generátory
- názvy generačných skriptov musia byť samo-opisné
- nikdy neupravujte staršie migrácie (neplatí pre lokálne, nezdieľané)
- pre dlhé migračné skripty píšete sprievodný text na výstupnú konzolu
 - pre výstup na konzolu používajte výhradne metódu *say*
 - neinformujte o každej zmene, len sumárne výpisy (napr. „*User → 30 rows has been removed*“, „*— added column to Users*“)

10.1.5 Model

- organizácia kódu
 - relačné príkazy
 - validačné príkazy
 - ďalšie skupiny príkazov
 - deklarácie verejných metód
- využívaj už existujúce validačné metódy (zdroj: dokumentácia RoR)
- pri každej zmene modelu je nutné upraviť testy (viac v časti testovanie)

10.1.6 Ovládač (ang. Controller)

- organizácia kódu
 - filtrovacie príkazy
 - verejné metódy prístupné cez adresu
 - ostatné verejné metódy
 - súkromné metódy (príkaz *private* oddeliť desiatimi znakmi '#')
- z dôvodu ochrany pred SQL-Injection útokom pre dopyty na databázu používajte podmienky s oddelenými premennými

```
1 User.find(:conditions => ["name like ? and age >= ?",
2 @user.name, @user.age])
```

10.1.7 Pohľad (ang. View)

- z dôvodu prevencie pred XSS útokom, akékoľvek dáta prijaté od používateľa zobrazte cez metódu `h()`
- jednoduché reťazce nikdy nevypisujte cez `<%= "reťazec"%>`

10.2 Testovanie

Ďalším podstatným prvkom zabezpečenia kvality je testovanie. Čím lepšie je otestovaný kód, tým lepší je výstup a nižšia pravdepodobnosť výskytu chýb. Kód je možné testovať na viacerých úrovniach preto v tejto kapitole opisujem tri základne úrovne:

- jednotkové testovanie
 - najnižšia úroveň, kde sa testujú jednotlivé funkčné bloky podľa špecifikácie
 - základná forma testovania, ktorou sa predchádza kalkulačným a multi-variantovým chybám
- funkčné testovanie
 - stredná úroveň kde sa testujú komplexnejšie funkcionálne prvky na úrovni ovládača
 - každá akcia ovládača by mala mať aspoň základné funkčné testy
- integračné testovanie
 - vyššia úroveň, kde sa testuje správanie aplikácie na úrovni používateľských prípadov, prípadne používateľských príbehov
 - časovo náročné definovanie i vykonanie týchto testov
 - najlepšie overí fungovanie systému naprieč jednotlivými inkrementami

V nasledujúcich kapitolách sú opísané metodické pokyny, ktoré musí dodržiavať každý člen tímu podieľajúci sa na tvorbe zdrojových kódov.

10.2.1 Procesy

V tejto kapitole sú definované procesy, ktoré je nutné dodržiavať pre zabezpečenie kvality kódu. V nasledujúcej tabuľke sú vymenované:

#	Názov postupu	Kapitola
1	Tvorba jednotkových testov	10.2.1.A
2	Tvorba funkčných testov	10.2.1.B
3	Tvorba integračných testov	10.2.1.C
4	Modifikovanie jednotkových testov	10.2.1.D
5	Modifikovanie funkčných testov	10.2.1.E
6	Modifikovanie integračných testov	10.2.1.F
7	Spustenie testov	10.2.1.G

10.2.1.A Tvorba jednotkových testov

Vstup: Neotestovaný komplexný funkčný kód

Výstup: Testovacia jednotka pre každú komplexnú funkcionálnosť

Proces:

1. Vytvoriť príslušný súbor s názvom a umiestnením podľa konvencií RoR
2. Vytvoriť prípravky pre testovanie
3. Vytvoriť podmienené jednotkové testy spĺňajúce formát definovaný v kapitole 10.2.2 pre každú komplexnú funkciu/metódu

10.2.1.B Tvorba funkčných testov

Vstup: Neotestovaná akcia ovládača

Výstup: Funkčný test pre akciu ovládača

Proces:

1. Vytvoriť príslušný súbor s názvom a umiestnením podľa konvencií RoR
2. Vytvoriť funkčné testy spĺňajúce formát definovaný v kapitole 10.2.2 pre každú verejnú akciu ovládača

10.2.1.C Tvorba integračných testov

Vstup: Neotestovaný používateľský prípad

Výstup: Integračný test pre základné hranice používateľského prípadu

Proces:

1. Vytvoriť príslušný súbor s názvom a umiestnením podľa konvencií RoR
2. Vytvoriť integračné testy spĺňajúce formát definovaný v kapitole 10.2.2 pre každú komplexnú funkciu/metódu

10.2.1.D Modifikovanie jednotkových testov

Vstup: Zmenená funkcionálnosť s existujúcim jednotkovým testom

Výstup: Rozšírená testovacia jednotka

Proces:

1. Podľa zmeny funkcionality upraviť prípravky
2. V závislosti na zmene očakávaných výstupov upraviť podmienené testy, aby zahrnuli túto zmenu a zároveň zahrnuli nové krajné prípady ktoré vznikli

10.2.1.E Modifikovanie funkčných testov

Vstup: Zmenená logika správanie akcie ovládača

Výstup: Rozšírený funkčný test

Proces:

1. V závislosti na zmene upraviť funkčný test tak, aby zahrňoval upravenú logiku.

10.2.1.F Modifikovanie integračných testov

Vstup: Zmenená logika používateľského prípadu

Výstup: Rozšírený integračný test

Proces:

1. V závislosti na zmene upraviť integračný test aby zahrňoval upravenú logiku.

10.2.1.G Spustenie testov

Vstup: Zdrojový kód s vytvorenými testovacími jednotkami a integračnými testami

Výstup: Zdrojový kód, ktorého kvalita je overená vykonanými testami

Proces:

1. Pred zdieľaním zdrojových kódov spustiť automatizované testy - v konzole spustiť príkaz `rake test` nad koreňovým priečinkom aplikácie.
2. Ak upravený kód neprejde testami, upraviť zdrojový kód alebo testy.

10.2.2 Konvencie pre testovanie

Nasledujúce kapitoly definujú interné konvencie pre jednotlivé typy testov. Spoločné ohraničenia:

- test musí mať plnovýznamové pomenovanie
- každý test testuje len jeden prípad
- čitateľnosť a samo-opisnosť má prednosť pred efektivitou
- využívajte generátory pre správne umiestnenie a pomenovanie testov
- na prerušenie testu nepoužívajte výnimky

10.2.2.A Jednotkové testy

Minimálne prípady pre testovanie:

- nový alebo upravený validačný skript
- nová metóda v modeli, ktorá má komplexnú funkcionálnosť
- nové ohraničenie modelového objektu

Ohraničenia:

- testovanie výhradne pomocou `ActiveSupport::TestCase`

```

1 class UserTest < ActiveSupport::TestCase
2   fixtures :users
3
4   test "user name can't be null" do
5     assert_not_nil users(:user).name
6   end
7 end

```


10.2.2.B Funkčné testy

Minimálne prípady pre testovanie:

- úspešnosť dopytu na akciu
- správne presmerovanie používateľa
- používateľ bol autentifikovaný
- správne nastavené objekty na výstupe
- používateľovi sa zobrazila požadovaná správa

Ohraničenia:

- testovanie výhradne pomocou *ActionController::TestCase*

```
1 class UsersTest < ActionController::TestCase
2   test "should create user" do
3     assert_difference('User.count') do
4       user :create, user: {name: 'some name'}
5     end
6
7     assert_redirected_to user_path(assigns(:user))
8   end
9 end
```

10.2.2.C Integračné testy

Minimálne prípady pre testovanie:

- dokončený používateľský prípad, ktorý nemá naplánované významne úpravy logiky vykonania

Ohraničenia:

- testovanie výhradne pomocou *ActionDispatch::IntegrationTest*

```
1 class UserFlowsTest < ActionDispatch::IntegrationTest
2
3   fixtures :users
4
5   test "login and browse site" do
6     https!
7     get "/sign_in"
8     assert_response :success
9
10    post_via_redirect "/sign_in",
11                      username: users(:user).username,
12                      password: users(:user).password
13    assert_equal '/tasks', path
14    assert_equal 'Welcome user!', flash[:notice]
15
16    https!(false)
```

```

17   get "/posts/all"
18   assert_response :success
19   assert_assigns(:tasks)
20 end
21 end

```

10.3 Spätná kontrola kvality

Kvalita kódu sa dá do určitej miery merať. Existujú metriky ktoré umožňujú sledovať kvalitatívne nedostatky a zistiť ich zdroj(či už ľudský alebo technický). Jedným z hlavných ukazovateľov je ukazovateľ pokrytia kódu testami (z ang. code coverage). Na získanie tohto ukazovateľa pre jednotlivé triedy a metódy v jazyku ruby používame gem (plugin/knižnica pre ruby programy) *MetricFu*. Tento gem kombinuje metriky omnoho viac metrík z viacerých zdrojov:

Názov gemu (zdroja)	Opis
Cane	Blokje zostavenie projektu v prípade zistených nedostatkov v kóde. (parametrizovateľný)
Churn	Zaznamenáva históriu zmien jednotlivých funkčných blokov. Výstupom je frekvencia zmien.
Flog	Reportuje najzložitejší kód. Vytvára reporty, ktoré sú dobrým ukazovateľom zdrojov potenciálnych problémov.
Reek	Vyhľadáva pachy v kóde.
Saikuro	Analyzátor cyklomatickej zložitosti pre jazyk Ruby.
Roodi	Analyzuje kvalitu Objektovo Orientovaného Dizajnu.
CodeMetrics	Analyzuje základné metriky kódu, ako napríklad počet riadkov, metód, tried.
Rails best practices	Analyzuje kvalitu z pohľadu najlepších praktík v programovom rámci RoR.
SimpleCov	Reportuje mieru pokrytia kódu testami

10.4 Zmeny v letnom semestri

V nasledujúcich sekciách je opísaná zmena v manažmente kvality oproti zimnému semestru. Sú opísané učené rozhodnutia a ich dôvody.

Testovanie

Počas letného semestra sme sa v značnej miere venovali aj kvalite. Už zo stanovených globálnych cieľov je zrejmé, že zabezpečenie kvality vyvíjaného produktu hralo podstatnú rolu. Pre dosiahnutie stanovených cieľov sme však museli urobiť ústupky. TDD prístup, ktorý je silne závislý na testovaní aplikácie, predlžuje dobu potrebnú na vývoj aj najmenšieho inkrementu. Keďže vyvíjaná aplikácia nemá zásadné dopady z pohľadu spoločenských dopadov, mohli sme si dovoliť testovanie presunúť až na záver vývoja. Toto rozhodnutie odporuje agilným prístupom, avšak s prihliadnutím na časové obmedzenia a schopnosti vývojárov sme toto rozhodnutie museli urobiť. Tým pádom sme zadané metodické pokyny pre testovanie vylúčili s vývoja.

Aj napriek tomuto rozhodnutiu, sme do jedného šprintu zakomponovali úlohu definovania základných funkčných testov. Potrebovali sme pokryť navýznamnejšie procesy vo vyvíjanom systéme, pretože sa nám viac krát stalo, že sme pri spájaní zdrojových kódov zanesli do aplikácie runtime-ové chyby. Tieto sa prejavili často v nevhodnom čase - tesne pred prezentáciou a pod.

Ako kompenzáciu absencie TDD (najprv test potom funkcionality) sme ako finálnu fázu zaviedli testovací šprint. V tomto šprinte mali jednotlivci za úlohu podstivo prejsť maximálny možný počet kombinácií prípadov použitia. Keďže niektoré chyby spôsobili predčasné prerušenie testovania, bol tento v rámci jedného šprintu aplikovaný dva krát. Vždy so zadaním blokujúceho bugu do systému Jira zodpovednej osobe. Pred ukončením šprintu sme ohodnotili vážnosť jednotlivých chýb, ktoré boli priradené podľa oblasti výskytu najskúsenejšej osobe.

Okrem interného testovania sme sa zahrnuli aj testovanie tretími stranami. Toto je opísané v kapitole 6 dokumentácie k inžinierskemu dielu.

Konvencie písania zdrojových kódov

Túto metodiku sme zaviedli, hlavne za účelom zachovania konzistencie naprieč zdrojovými kódmi. Vďaka pravidelnej kontrole prostredníctvom Code Review sme dokázali vynútiť dodržiavanie konvencií podľa pokynov.

V rámci metodiky a následne CodeReview sa nám nepodarilo udržať konzistenciu názvov s pohľadu správnosti pomenovania (semantickej). Keďže sme si nevytvorili slovník pojmov používaných v problémovej oblasti a často sme pri preklade z angličtiny do slovenčiny a naopak využívali odlišnejšie ekvivalenty, dostali sme sa do problémového stavu (napr. jedna osoba pomenovala úlohu Task a následne druhá Campaign). Pojmy sme niekedy používali skoro až náhodne. Časom, ako rozsah zdrojových kódov rástol, sa tento problém prehĺbil a citelne sa prejavil hlavne v časoch hlbších refaktoringov. Ak sa bude vo vývoji pokračovať aj po skončení letného semestra, bude nutné vyhradiť čas na definovanie kvalitného slovníku pojmov problémovej oblasti a taktiež zjemniť metodiku konvencií.

Code review

Aj keď sme tento proces v dokumente spomenuli už viackrát, má svoje opodstatnenie hlavne z pohľadu kvality. Všetci sme sa na záver zhodli, že akákoľvek forma spätnej kontroly napomáha nie len predchádzaniu chybám (najmä runtime-ového charakteru) ale aj zachovaniu tzv. Big Picture. Každý účastník vďaka tomu vie o funkcionalite systému ako celku a nie len svojej problémovej oblasti (“Budujem hrad, nie len jednu vežu!”).

11 Manažment podpory vývoja

Autor: Bc. Patrik Štrba

V prvých šprintoch sme si nastavili procesy, ktoré majú zabezpečovať kvalitu produktu. Medzi tieto procesy patrí dodávanie kódu, vytváranie verzií produktu alebo vytváranie produkčných verzií a ich nasadenie.

V procese dodávania kódu sme použili architektúru založenú na spoločnom tímovom repozitári, kde má každý člen tímu právo čítať zmeny, ktoré boli vykonané na danom repozitári. Každý člen v tíme má potom svoj vlastný repozitár, na ktorý už má právo zápisu. Architektúra je postavená na spoločnom repozitári a samostatných repozitároch pre každého člena, kde sa zmeny vykonávajú pomocou *pull-request*, kde každý člen samostatne pracuje na svojom repozitári. Do svojho repozitára si ukladá zmeny vykonané v systéme potrebné pre úspešné dokončenie úlohy pridelenej na stretnutí. Po dokončení sa všetky tieto zmeny sa zabalia do požiadavky (*pull-request*), ktorá je následne spracovávaná v procese schvaľovania takýchto požiadaviek. Týmto sme dokázali zabezpečiť prísun kvalitného kódu, kedy sa schvaľujú len kódy, ktoré sú čitateľné a nie je v nich na prvý pohľad vidno vznik chýb a podobne. Týmto zároveň vykonávame aj *code review*, na ktorom sa podieľajú všetci členovia tímu.

Ďalší proces, ktorý vznikol pre zvýšenie kvality a jednoduchšie nasadenie produktu, je produkčné nasadenie, kedy je dôležité dodávať zmeny rýchlo, avšak je potrebné produkt otestovať ešte pred nasadením do ostrej prevádzky. Preto sme použili balík nástrojov, ktoré nám umožňujú efektívnejšie a zároveň nám pridávajú možnosť jednoduchšej manipulácie pri vytváraní produkčných verzií. Zároveň nám umožňuje jednoducho prezerať vytvorené predchádzajúce produkčné verzie a v prípade vytvorenia nefunkčnej produkčnej verzie, nasadiť staršiu verziu, avšak bez chýb.

V každom zložitejšom softvérovom produkte sa vyskytujú chyby a preto musíme tento problém riešiť. Ideálne je udržiavať počet nájdených chýb na nule a preto je dôležité si tieto chyby evidovať. Na evidovanie chýb používame nástroj JIRA. Ten nám umožňuje jednoducho vytvárať a ďalej manipulovať s chybami, ktoré objavil jeden z členov alebo nám boli nahlásené cudzou osobou. Tieto chyby vieme pridelovať osobám, ktoré mali na starosť funkcionality v ktorej sa chyba vyskytla a preto je z pohľadu efektívnosti najlepšie aby chybu riešil tvorca danej funkcionality. V prípade ak sa jedná o naliehavú chybu, je možné ju riešiť priamo členom, ktorý chybu našiel alebo bol o nej informovaný.

11.1 Použitie vetiev (branch) v tímovom repozitári

V tímovom repozitári sa nachádzajú vetvy:

- master
- develop

Master vetva obsahuje funkcionality, ktorá je otestovaná a pripravená na proces nasadenia do produkcie. Tento proces vykonáva administrátor repozitára.

Develop vetva obsahuje funkcionality na ktorej sa pracuje, avšak sa zároveň považuje za dokončenú. Do tejto vetvy nesmie byť pridaná funkcionality, ktorá obsahuje chyby alebo nie je dokončená.

11.2 Potvrdenie funkcionality do tímového repozitára

Vstup: Vytvorený pull-request

Výstup: Spracovaný pull-request

Proces:

1. O potvrdzovanie pull-requestov sa starajú správcovia repozitára. Každý pull-request je najskôr potreba analyzovať. Po analýze sa následne zmeny potvrdia, alebo zrušia a doplnia poznámkou, čo nie je vhodné a treba prerobiť.
2. Schvaľovanie prebieha formou označenia požiadavky za schválenú. Pre začlenenie požiadavky musia obaja správcovia repozitára udeliť schválenie aby mohla byť požiadavka schválená.
3. Ak je pull-request v poriadku, tak sa označí ako schválený a následne je zlúčený. V prípade ak sa jedná o súrne opravy, tak je možné proces schvaľovania preskočiť a požiadavku schváliť priamo.

11.3 Vytvorenie novej verzie

Vstup: Vytvorené commity

Výstup: Vytvorenie značky (tag) označujúca daný commit za novú verziu

Proces:

1. Vytvorenie prebieha vždy po uzatvorenom úseku dokončených úloh, kde už nie je známa žiadna chyba.
2. Číslo verzie sa bude deliť na 2 časti, pričom časti budú oddelené bodkou.
3. Prvé číslo udáva verziu produktu. Toto číslo sa zmení len pri dokončení všetkých zmien, ktoré sú naplánované do novej verzie.
4. Druhé číslo udáva verziu, ktorá zmení len pri dokončení všetkej funkcionality, ktorá bola naplánovaná na daný šprint.
5. Osoba poverená správou verzií je administrátor repozitára.

11.4 Vytvorenie novej funkcionality

Vstup: Pridelená úloha v systéme na evidenciu úloh

Výstup: Vytvorená vetva s požadovanou funkcionality

Proces:

1. Vytvoriť novú vetvu a zvoliť názov s pridaním prefixu „feature-“ názov popisujúci danú funkcionality
2. Vytvoriť novú funkcionality
3. Vytvoriť commit a napísať správu opisujúcu funkcionality (čo a ako bolo implementované)
4. Nahrať zmeny na vzdialený repozitár (push)
5. Vytvoriť požiadavku na pridanie zmien do tímového repozitára, konkrétne do vetvy *develop*

11.5 Oprava chyby na vývojovej verzii aplikácie

Vstup: Objavená chyba a nahlásená v systéme pre správu chýb

Výstup: Vytvorená vetva s požadovaným názvom

Proces:

1. Vytvoriť novú vetvu a zvoliť názov s pridaním prefixu „bug-“ a číselný identifikátor chyby
2. Vytvoriť opravu pre danú chybu
3. Vytvoriť commit a napísať správu opisujúca chybu
4. Nahrať zmeny na vzdialený repozitár (push)
5. Vytvoriť požiadavku na pridanie zmien do tímového repozitára, konkrétne do vetvy *develop*

11.6 Oprava chyby na produkčnej verzii aplikácie

Vstup: Objavená chyba a nahlásená v systéme pre správu chýb

Výstup: Vytvorená vetva s požadovaným názvom

Proces:

1. Vytvoriť novú vetvu a zvoliť názov s pridaním prefixu „hotfix-“ a číselný identifikátor chyby
2. Vytvoriť opravu pre danú chybu
3. Vytvoriť commit a napísať správu opisujúca chybu
4. Nahrať zmeny na vzdialený repozitár (push)
5. Vytvoriť požiadavku na pridanie zmien do tímového repozitára, konkrétne do vetvy *master*

11.7 Spolupráca na spoločnej funkcionalite

Vstup: Oprávnenie na repozitáre ostatných členov s ktorými je potrebná spolupráca
Nahrané zmeny na vzdialených repozitároch (push)

Pridané vzdialené repozitáre potrebných členov v lokálnom repozitári

Výstup: Možnosť spolupracovať na úlohe spoločne

Proces:

1. Načítať (fetch) si zmeny z repozitára z ktorého chcem pridať funkcionalitu
2. Zlúčiť (merge) lokálnu vetvu (local branch), na ktorej chcem pridať požadovanú funkcionalitu so vzdialenou vetvou (remote branch), kde sa nachádza požadovaná funkcionalita.
3. Nahrať zmeny na svoj repozitár (push)
4. Prvé tri kroky opakovať, pokiaľ nie je funkcionalita hotová.

11.8 Zmeny v letnom semestri

Vývoj prebieha ako bol stanovený v zimnom semestri. Používame jeden spoločný repozitár a každý má svoj samostatný repozitár kde si zálohuje svoju prácu. Z tohoto repozitára sa potom zmeny spájajú do spoločného repozitára cez pull-requests. Toto nám poskytuje kontrolu nad pridanými zmenami a zabezpečuje jednoduchší manažment.

Správca repozitára sa pre leto zvolili aj ďalší dvaja členovia a tým pádom sme štyri a to Patrik Štrba, Martin Toma, Filip Bednárík a Róbert Černý. Títo členovia majú právo zahrňovať vytvorené pull-requests do spoločného repozitára.

V stanovený čas a po dohode sa vytvorí verzia aplikácie, ktorá sa následne nasadí na produkčný server. Na nasadenie používame nástroj Jenkins, ktorý dostane pokym od repozitára pokiaľ sa niečo zmení vo vetve master. Následne Jenkins vytvorí build, ktorý po úspešnom stiahnutí a vytvorení novej verzie aj nasadí.

V rámci vytvorenia novej verzie prebieha stiahnutie najnovšej verzie aplikácie a aktualizácia všetkých balíkov (gemov). Následne sa spustí databázová migrácia a skontroluje databáza. Potom sa vygenerujú všetky statické veci a spustia testy. V prípade ak toto všetko zbehne v poriadku tak sa nasadí nová verzia do produkcie.

12 Plány

12.1 Zimný semester

Tab. 2: Plán na zimný semester

Týždeň	Dátum začiatku	Dátum konca	Popis činnosti
1	Začiatok semestra	15.10.2012	<ul style="list-style-type: none"> • Tvorba webovej stránky • Dizajn plagátu • Analyzovanie domény Digital Sweats-hop a podobných riešení • Výber vhodných technológií a programovacích jazykov • Vytvorenie šablón pre dokumentáciu • Vytvorenie jednotnej šablóny pre zápisky
4-6	17.10.2013	31.10.2013	<ul style="list-style-type: none"> • Vytvorenie backlogu projektu • Rozbehnutie dohodnutých technológií • Implementovať prihlasovanie používateľa a základný dátový model
6-8	31.10.2013	14.11.2013	<ul style="list-style-type: none"> • Vytvorenie profilu používateľa • Integrácia so sociálnymi sieťami • Vytvoriť UI pre aplikáciu • Analyzovať a vybrať vhodný názov pre našu aplikáciu
8-10	14.11.2013	28.11.2013	<ul style="list-style-type: none"> • Vytvorenie finálnych verzií metodík • Zaradenie vytvorených metodík do procesu vývoja • Vytvoriť a analyzovať kategórie úloh v aplikáciách
10-12	28.11.2013	12.12.2013	<ul style="list-style-type: none"> • Doladovanie a refaktoring • Testovanie • Doladovanie dokumentácie • Prezentovanie stavu projektu na konci semestra

12.2 Letný semester

12.2.1 Krátkodobý plán

Počas skúškového obdobia sme sa ako tím dohodli na týchto povinnostiach, ktoré do začatia letného semestra splní každý náš člen. Výsledky sa budú prezentovať na prvom tímovom stretnutí v letnom semestri.

- Refaktorovanie a upratovanie projektu
- Ošetrovanie a doladovanie doteraz implementovaných funkcionalít
- Testovanie aplikácie
- Úprava grafických chýb
- Zjednotenie dizajnu (farby, tvary, navigácia, logá, motívy)
- Vytvorenie funkčnej verzie do vetvy Master
- Vo vetve develop na do začiatku semestra aktualizuje funkčná upravená verzia so všetkými zmenami, ktoré prebehli počas skúškového obdobia
- Migrácia členov na jednotnú databázovú technológiu (MySQL)
- Študovanie metódik
- Aspoň 3 stretnutia celého tímu prostredníctvom Google Hangouts
- Každý člen spíše 3 nové user story, ktoré mu v systéme chýbajú
- Každý člen spíše 3 user story, ktoré su podľa neho najmenej podstatné z doteraz implementovaných

12.2.2 Dlhodobý plán

Tab. 3: Plán na letný semester

Týždeň	Dátum začiatku	Dátum konca	Popis činnosti
1	20.02.2014	26.02.2014	<ul style="list-style-type: none"> • Retrospektíva, čo sa spravilo za skúškové obdobie
2-3	27.02.2014	12.03.2014	<ul style="list-style-type: none"> • Vylepšenie grafickej podoby aplikácie • Práca na dátovom modeli a jeho zapracovanie do aplikácie • Optimalizácia a zrýchľovanie aplikácie • Doplňovanie rozhraní, pre externé služby • Testovanie aplikácie model
4-5	13.03.2014	25.03.2014	<ul style="list-style-type: none"> • Funkcionalita orientovaná na zadávateľa • Úprava rozhrania pre zadávanie nových úloh • Vytvorenie prehľadnej štatistiky výsledkov pre zadávateľov • Úprava admin panelu, úprava kompetencií rolí v systéme • Štatistické údaje pre administrátora, kontrola nad financiami
6-7	26.03.2014	09.04.2014	<ul style="list-style-type: none"> • Funkcionalita orientovaná na pracovníkov • Prehodnotenie štruktúry profilu pracovníka • Pridanie nových ocenení za dobre odvedenú prácu • Finančný model a odmeňovanie
8	10.04.2014		<ul style="list-style-type: none"> • Práca na plagáte IIT.SRC • Oprava chýb a testovanie • Vytvorenie kompletného prekladu aplikácie pre SK a ENG
9-10	17.04.2014	29.04.2014	<ul style="list-style-type: none"> • Príprava na TPCup • Testovanie aplikácie v reálnych podmienkach • Stretnutia s potenciálnymi klientmi a používateľmi • Dohoda o pokračovaní vo vývoji projektu po ukončení predmetu
11 - 12	30.04.2014	Koniec semestra	<ul style="list-style-type: none"> • Údržba, posledné úpravy zdrojového kódu • Dokončenie dokumentácie • Odovzdanie a prezentovanie výsledkov • Víťazstvo v súťaži TPCup

PRÍLOHY

A: Preberací protokol

Preberací protokol

Tímový projekt 2013/2014

Tím 11 - Pinta.sk

Predmet odovzdávania:

- Dokumentácia k riadeniu projektu - letný semester
- Dokumentácia k inžinierskemu dielu - letný semester

Pedagogický vedúci - Ing. Dušan Zeleník

Podpisom potvrdzuje prevzatie vyššie uvedených častí projektu a dokumentácie

dátum _____

podpis _____