

Fakulta informatiky a informačných technológií
Slovenská technická univerzita

Group de'Cider

Dokumentácia k tímovému projektu časť 1
(Riadenie projektu)

Tím: číslo 1, sElectors

Pedagogický vedúci: Ing. Ondrej Kaššák

Členovia tímu: Bc. Lukáš Babula, Bc. Pavel Balashov, Bc. Michal Dolnák, Bc. Michal Hucko, Bc. Elena Štefancová, Bc. Miroslav Valčíčák, Bc. Andrej Vítek

Akademický rok: 2017/2018

Úvod	2
1 Role členov tímu a podiel práce	3
1.1 Podiel práce na dokumentácii	4
2 Aplikácie manažmentov	5
2.1 Manažment agilného riadenia softvérového vývoja	5
2.2 Manažment verzií	5
2.3 Manažment chýb	5
2.4 Manažment komunikácie	6
3 Sumarizácia šprintov	7
3.1 Šprint 1 (5.10. - 19.10.)	7
3.2 Šprint 2 (19.10. - 2.16.)	12
3.3 Šprint 3 (2.11. - 16.11.)	17
4 Globálna retrospektíva	22
4.1 Retrospektíva k šprintu 1 (26.10.2017)	22
4.2 Retrospektíva k šprintu 2 (9.11.2017)	22
4.3 Retrospektíva k šprintu 3 (16.11.2017)	23
5 Motivačný dokument	26
5.1 Tím	26
5.2 Témy	27
Príloha A: Zoradenie všetkých tém podľa priority	29
Príloha B: Rozvrh tímu	30
Záver	32
Príloha A Export evidencie úloh	33
Príloha B Metodiky	37

Úvod

Dokument obsahuje podrobnú správu tímu sElectors po prvých 3 šprintoch v rámci predmetu Tímový projekt. V tomto súbore sa nachádzajú informácie k časti riadenia projektu, popisy šprintov, podiel úloh práce členov tímu, metodiky k jednotlivým častiam a pohľad na jednotlivé retrospektívy počas šprintov. Rovnako tak táto správa obsahuje aj export úloh z nástroja Jira použitého pri riadení tímu.

Celý tím mal minimálne skúsenosti s aplikovaním agilného vývoja na riešenie problémov, čo sa podpísalo na úvodných fázach riešenia projektu.

1 Role členov tímu a podiel práce

Ing. Ondrej Kaššák

Vedúci tímu sElectors. Jeho úlohou bolo pripraviť product backlog a inštruovať nás k úspešnému zvládnutiu vývoja projektu. Počas sedení prezentuje svoje predstavy o finálnom produkte z pohľadu role product ownera. Pri nekorektnom postupe v práci nás vždy usmerní správnym smerom.

Bc. Lukáš Babula

Zodpovedný za kontinuálnu integráciu. Počas vývoja produktu je potrebné manažovať nástroj kontinuálnej integrácie. Náš tím za týmto účelom používa Jenkins. Lukáš nastavil tento nástroj a v dohľadnej dobe sa bude naďalej venovať jeho správe a konfigurácii.

Bc. Pavel Balashov

Pavel je zodpovedný za administráciu nástroja určeného na podporu agilného vývoja v tíme. V našom prípade sa jedná o nástroj JIRA. Jeho úlohou bolo nasadenie tohto nástroja na náš server. Tento proces bol spojený s kompletnou konfiguráciou a manažovaním, čo zahŕňa pravidelné aktualizovanie úloh počas šprintu. Okrem toho sa Pavel podieľal na vývoji frontend-ovej časti projektu.

Bc. Michal Dolnák

Michal je zodpovedný za frontend-ovú časť projektu. Jeho úlohy boli spojené hlavne s touto časťou vývoja. Mišo nastavil celé prostredie na vývoj frontendu a neskôr aj odkomunikoval celý proces nasadenia na server. Na túto úlohu bol v projekte použitý Angular. Okrem spomenutých vecí sa podieľal aj na návrhu volaní pre komunikáciu medzi backend-om a frontend-om.

Bc. Michal Hucko

Michal je zodpovedný za backend projektu. Podstatnú časť vývoja sa venoval REST API pre aplikáciu. Táto časť projektu bola naprogramovaná nad rámcom Django Rest. Podieľal sa na implementácii Docker riešenia pre automatizované nasadenie a nastavení Apache reverse proxy na školskom serveri. Michal sa podieľa aj na externej komunikácii tímu.

Bc. Miroslav Valčíčák

Miroslav je zodpovedný za implementáciu docker riešenia pre automatizované nasadenie nášho riešenia. Počas šprintov sa venoval najmä nastaveniu nginxu určeného na komunikáciu medzi backend-om a frontend-om, ktorý je súčasťou spomínaného riešenia. Okrem toho sa podieľal na úspešnom nastavení SSL certifikátu určeného na použitie Google autorizačných služieb.

Bc. Andrej Vítek

Andrej je zodpovedný za správu tímovej stránky. Počas šprintov sa venoval upravovaniu obsahu a aktualizovaniu informácií. Andrej sa ďalej značne podieľal na implementácii

funkcionality na backend-e. Tento proces zahŕňal aj niektoré kľúčové skripty potrebné na testovanie. Okrem iného sa podieľa na návrhu grafických rozhraní frontend-u.

Bc. Elena Štefancová

Ela je zodpovedná za návrh grafických rozhraní frontend-u. Táto úloha je spojená s prezentovaním mockupov a zapracovaní častých pripomienok tímu. Ela sa ďalej významne podieľa na organizácii externej komunikácie tímu, čo zahŕňa vypracovávanie dokumentov, písanie správ a podobne. Ela tiež pracuje na implementácii frontend-ových riešení v Angulari.

1.1 Podiel práce na dokumentácii

Tab. 1: Prehľad podielu práce na dokumentácii k riadeniu (%)

	Lukáš	Pavel	Mišo D	Mišo H	Miro	Andrej	Ela
Úvod	10%	0%	0%	90%	0%	0%	0%
Role členov tímu a podiel práce	13%	13%	13%	22%	13%	13%	13%
Aplikácie manažmentov	10%	55%	0%	35%	0%	0%	0%
Sumarizácie šprintov	15%	15%	15%	15%	15%	10%	15%
Globálna retrospektíva ZS/LS	5%	30%	0%	35%	0%	30%	0%
Motivačný dokument	30%	5%	5%	20%	5%	5%	30%
Metodiky	10%	10%	10%	10%	10%	10%	40%
Export evidencie úloh	0%	95%	0%	5%	0%	0%	0%

Tab. 2: Prehľad podielu práce na dokumentácii k inžinierskemu dielu (%)

	Lukáš	Pavel	Mišo D	Mišo H	Miro	Andrej	Ela
Úvod/Záver	5%	0%	0%	5%	0%	90%	0%
Globálne ciele	5%	0%	0%	0%	0%	90%	5%
Celkový pohľad na systém	0%	0%	20%	15%	0%	65%	0%
Modul hlas. stratégií	90%	0%	0%	0%	0%	10%	0%
Modul hlasovania	5%	5%	60%	0%	5%	25%	0%
Modul autentifikácie	5%	0%	0%	0%	95%	0%	0%
Prílohy	5%	0%	25%	0%	0%	70%	0%

2 Aplikácie manažmentov

2.1 Manažment agilného riadenia softvérového vývoja

V rámci tímového projektu sme sa pravidelne stretávali v štvrtky od 12:00 spolu s produktovým vlastníkom. Tieto stretnutia boli v miestnosti s veľkou obrazovkou a tabuľou na písanie, ktoré sme aktívne využívali. Postupom času sme si osvojili zopár pravidiel, ktoré každý musí počas stretnutia dodržiavať:

- Meškanie nie je povolené.
- Notebooky a mobily sú zakázané (okrem prezentačného a zapisovateľského notebooku).

Stretnutia boli vedené Scrum mastrom, ktorý pred každým napísal na tabuľu program v bodoch. V zásade sa postupovalo podľa dvoch formátov: vyhodnotenie šprintu + plánovanie (planning) a stretnutie v strede šprintu.

Počas plánovania sme si vybrali používateľské príbehy z backlogu a nasledoval plánovací poker. V tejto časti každý ohodnotil používateľský príbeh počtom story pointov, následne sme sa dohodli buď na rozbití používateľského príbehu na menšie, alebo sme prešli na delenie úloh. Ak sme sa pri hlasovaní nezhodli, prebehla krátka debata, po ktorej sa hlasovanie opakovalo. Takto sme postupovali, až kým sme sa nezhodli. Po tomto procese bol úlohám pridelený za zodpovedný člen. Každú z nich sme evidovali v systéme JIRA, kam má každý člen prístup. V prípade stretnutia v strede šprintu sme na začiatku prešli stav jednotlivých úloh a vyťaženie členov tímu. Ak sa našiel niekto, kto mal problém so svojou úlohou, snažili sme sa mu čo najefektívnejšie pomôcť.

Okrem štandardných stretnutí v čase predmetu sme si každý druhý týždeň vymedzili čas na retrospektívu. Termín tohto stretnutia bol v nasledujúcom týždni. Retrospektíva prebiehala v troch kolách, pred ktorými mal každý člen čas na rozmyslenie. V prvom kole sme sa vyjadrovali k tomu, čo sa nám páčilo. V druhom sme odpovedali na otázku toho čo sa nám na šprinte nepáčilo a v poslednom sme predostreli riešenie. Počas jednotlivých kôl mohol rozprávať len jeden človek a reakcie boli povolené až po skončení kola. Viac detailov tohto postupu obsahuj **metodika pre agilné riadenie softvérového vývoja**.

2.2 Manažment verzií

Pre správu verzií kódu sme použili GIT a GitLab. Každý člen tímu počas vyvoja pracoval s vetvami v tomto nástroji a vytvoril niekoľko commitov. Viac o tom možno nájsť v **metodike práce s GitLabom**.

2.3 Manažment chýb

GitLab sme taktiež použili pre manažovanie chýb (angl. issue tracking). Ak pri práci na úlohe člen tímu zistí, že aplikácia alebo jej časť nefunguje správne, ale v minulosti funkčná bola alebo bola za funkčnú označená, toto chybné správanie podrobne opíše do

GitLab issues. Počas prvých dvoch šprintov sme podcenili potrebnosť tohto nástroja, ale od tretieho šprintu sme ho začali pravidelne používať.

2.4 Manažment komunikácie

Jedným z kľúčových prvkov manažmentu je komunikácia. V rámci projektu ju vieme rozdeliť na internú (medzi členmi), kde ako nástroj používame Slack a externú (medzi vedením a potenciálnymi zákazníkmi), kde postačuje mail. Na komunikáciu s produktovým vlastníkom používame taktiež Slack.

3 Sumarizácia šprintov

3.1 Šprint 1 (5.10. - 19.10.)

Úvodný šprint bol najmä o zoznamovaní sa s agilným vývojom pomocou SCRUMU. V tomto šprinte sme story pointy pridelovali jednotlivým úlohám, čo sa v neskoršej fáze zmenilo. Planning poker prebiehal ku každému jednému z nich.

Definícia POST/GET

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Michal Hucko	Áno

Definovanie rozhrania a procesu tvorby rozhrania medzi frontend-om a backend-om.

Návrh DB schémy

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Áno

Navrhnutie dátového modelu pre funkcionality a jeho vizualizácia pomocou UML diagramu.

Implementácia backend v Django REST - jednoduché hlasovanie

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Implementácia základného vytvárania udalostí spolu s funkcionalitou hlasovania. Bude sa dať vytvoriť hlasovanie spolu s množinou možností. Následne je možné anonymne hlasovať za jednu z nich. Táto úloha je spojená aj s úvodnou inicializáciou projektu nad frameworkom.

Docker setup

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Áno

Vytvorenie konfiguračných súborov pre docker-compose a jednotlivé moduly.

Frontend - infraštruktúra

Zodpovedný	Overovateľ	Splnené
Michal Dolnák		Áno

Vytvorenie nového projektu a nastavenie technológie. Pridanie prekladov, volania backend-u. Nastavenie ciest v aplikácii.

Frontend - zaučenie, zdokumentovanie

Zodpovedný	Overovateľ	Splnené
Michal Dolnák		Áno

Podpora ostatných členov tímu pri svojich úlohách. Predanie svojich vedomostí o technológii.

Frontend - hlasovacia tabuľka a zobrazenie výsledkov

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Nie

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností. Prvky grafického rozhrania:

- tabuľka
- statický text pre možnosti
- textové pole pre meno
- hlasovacie tlačidlá pre možnosti
- tlačidlo pre potvrdenie hlasovania

Dôvod nesplnenia: Úloha nebola splnená z dôvodu komplikácií pri integrácii frontend a backend.

Frontend - vytvorenie hlasovacieho formulára

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- Názov
- Možnosti hlasovania v textovom bloku pod sebou
- API volanie po potvrdení
- Vygenerovanie URL na admin a hlasovacie rozhrania

Dôvod nesplnenia: Úloha nebola splnená z technických príčin - Elenin operačný systém nebol kompatibilný s požiadavkami implementačných nástrojov.

Nasadenie JIRA

Zodpovedný	Overovateľ	Splnené
Andrej	Pavel	Áno

Nasadenie kolaboračného systému JIRA na pridelený server.

Požiadat' o navýšenie RAM na tímovom serveri

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Požiadat' P. Lacka o zvýšenie parametrov servera.

Migrovanie úloh

Zodpovedný	Overovateľ	Splnené
Pavel	Ela	Áno

Migrovanie úloh z Asana. Zahŕňa prepísanie úloh do JIRA a kontrolu ďalším členom tímu.

Prihlásenie sa do TP Cup

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Ela	Áno

Vytvorenie prihlášky na TP cup, ktorá bude zahŕňať všetky body zo stránky predmetu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi. Po finalizácii sa projekt riadne odovzdá do AISu.

Metodika - dokumentovanie frontend kódu

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

Vytvorenie metodiky pre dokumentovanie frontend-u.

Metodika - práca s GitLab

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Vytvoriť metodiku ku práci s GitLabom:

- vetvy a ich názvy
- komentáre ku commit-u
- merge request a komentáre

Inštalácia a konfigurácia Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Áno

Nainštalovanie Jenkins na server. Nastavenie a zabezpečenie Jenkins. Nastavenie Gitlab notifikácie pre Jenkins. Vytvorenie Jenkins projektu. Nastavenie SCM v projekte. Rozšírenie Slacku o Jenkins agenta. Integrácia Slack kanála do Jenkins.

Vytvorenie tímového webu

Zodpovedný	Overovateľ	Splnené
Andrej, Lukáš	Pavel	Áno

Vytvorenie tímovej stránky:

- rozloženie elementov
- informácie o tíme a projekte
- šprinty
- dokumenty
- kontakt

Prieskum ohľadom stratégií hlasovania

Zodpovedný	Overovateľ	Splnené
Lukáš	Pavel	Nie

Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.

Dôvod nesplnenia: Analýza mnohých používaných a zaujímavých stratégií bola časovo výrazne náročnejšia, než bolo odhadované. Na konci šprintu bola preto úloha dokončená len na približne 60 %.

Vytvorenie mockupov pre aspoň 5 ďalších stratégií

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Vytvorenie mockupov obrazoviek ku hlasovanie pomocou rôznych stratégií.

Dôvod nesplnenia: Úloha nebola splnená, pretože priamo nadväzuje na úlohu *Prieskum ohľadom stratégií hlasovania*, ktorá nebola dokončená.

Návrh grafických rozhraní

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Vytvoriť prvotné mockupy pre rozhrania + vygenerovať CSS.

Dôvod nesplnenia: Úloha sa odložila na ďalší šprint, keďže nebola sfinalizovaná vízia produktu. Bola vykonaná len analýza v oblasti podobných webov.

3.2 Šprint 2 (19.10. - 2.16.)

Na predchádzajúcom šprinte sa prejavila neznalosť agilného vývoja, čo zapríčinilo nesprávny odhad úloh, ktorých sme si dali viac ako sme boli schopní zvládnuť. Všetky nedokončené úlohy sa preniesli do tohto šprintu.

Frontend - hlasovacia tabuľka a zobrazenie výsledkov

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Nie

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností.

Prvky grafického rozhrania:

- tabuľka
- statický text pre možnosti
- textové pole pre meno
- hlasovacie tlačidlá pre možnosti
- tlačidlo pre potvrdenie hlasovania

Dôvod nesplnenia: Úloha nebola splnená z dôvodu komplikácii pri integrácii frontend a backend.

Frontend - vytvorenie hlasovacieho formulára

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- Názov
- Možnosti hlasovania v textovom bloku pod sebou
- API volanie po potvrdení
- Vygenerovanie URL na admin a hlasovacie rozhrania

Dôvod nesplnenia: Technické problémy pretrvávali, z týchto dôvodov bolo nutné používať na vykonávanie úlohy VirtualBox, na ktorého efektívne využitie nemá Elena vhodný počítač. Z tohto dôvodu bola síce funkcionálna naprogramovaná, ale nestihli sa napísať testy.

Nasadzovanie projektu cez Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Nie

Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.

Dôvod nesplnenia: Problémy s nastavením adekvátnych práv pre komunikáciu Jenkins a Docker, čo výrazne oneskorilo ich integráciu. To bolo jednou z príčin nepreviazania Jenkins nástroja s projektom. Ďalšou príčinou bola nepripravenosť a neúplnosť častí projektu, ktoré mali byť použité na testovanie a nasadenie. Na konci šprintu preto nebolo možné odhadnúť, na koľko percent bola úloha splnená, pretože pokrok nebolo možné kvantifikovať.

Vytvorenie mockupov pre aspoň 5 ďalších stratégií

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.

Dôvod nesplnenia: Stále neboli vybrané stratégie, úloha sa teda odložila na neurčito.

Návrh grafických rozhraní

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Áno

Vytvoriť prvotné mockupy pre rozhrania + vygenerovať CSS.

Poznámka: Po doplnení prvého náčrtu finálnej vízie produktu boli využité poznatky z analýzy a úloha dokončená.

Prieskum ohľadom stratégií hlasovania

Zodpovedný	Overovateľ	Splnené
Lukáš	Pavel	Áno

Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.

Metodika testovania backend-u

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

V tejto metodike pokryjeme všetky detaily spojené s testovaním backend-u. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.

Metodika - dokumentovanie backend kódu

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

V tejto metodike pokryjeme všetky detaily spojené s dokumentáciou backend-u. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.

Metodika - testovanie frontend-u

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

Vytvorenie metodiky pre testovanie frontend-u.

Vyjednať navýšenie RAM + certifikát

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Napísať Peťovi Lackovi o zvýšenie RAM na serveri a zistiť či je možné získať SSL certifikát pre projekt. SSL je potrebný pre integráciu modulu prihlasovania cez Google+.

Dokonfigurovať Docker

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Dokonfigurovanie docker na serveri. Tento krok predchádza samotnému spusteniu dockeru na serveri. Je potrebné doinštalovať docker-compose. Nastaviť práva pre skupinu docker a otestovať spustenie kontajneru.

Backend - Úprava voľby hlasovania

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Používateľsky príbeh: Úprava voľby akéhokoľvek používateľa

Backend pre zmenu hlasovania používateľa cez HTTP PUT metódu.

Backend - Ukončenie hlasovania

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Používateľsky príbeh: Ukončenie hlasovania

Umožniť ukončenie hlasovania cez backend:

- úprava modelu hlasovania
- vytvorenie URL pre administráciu
- zabezpečiť, že sa nedá hlasovať na uzavretom hlasovaní

Metodika - dokumentovanie frontend kódu

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Andrej	Áno

Vytvorenie metodiky pre tvorbu a dokumentáciu frontend-ového kódu.

Metodika - vývoj a integrácia

Zodpovedný	Overovateľ	Splnené
Miroslav	Pavel	Nie

Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.

Dôvod nesplnenia: Až do konca šprintu nebol istý konkrétny postup pri vývoji kvôli problémom s prípravou jednotlivých prostredí a nástrojov.

Nastavenie Apache na serveri

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Nie

Nastavenie Apache na serveri, aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.

Dôvod nesplnenia: Nesplnenie 2 frontend-ových úloh, na ktorých bola úloha priamo závislá: hlasovacia tabuľka a zobrazenie výsledkov, vytvorenie hlasovacieho formulára.

Merge do mastra

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Úloha zahŕňa zodpovednosť nad tým aby sa stihli spojiť kódy frontend-u a backend-u. V prípade akýchkoľvek problémov zodpovedná osoba poskytne pomoc.

Frontend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Nie

User story: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google OAuth:

- vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu
- odhlásenie používateľa
- udržiavanie stavu prihlásenia na strane klienta
- vytvorenie sekcie prístupnej len pre prihláseného používateľa
- integrácia riešenia s backend-om

Dôvod nesplnenia: Úloha sa počas analýzy a implementácie ukázala zložitejšia ako pôvodný odhad z dôvodu integrácie viacerých technológií.

Backend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Nie

User story: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google

- vytvorenie používateľa cez Google účet (OAuth2)
- pridanie nášho hostname do overených hostanemov (Google)
- vytvorenie tajných kľúčov (Google)
- vygenerovanie SSL certifikátu
- nakonfigurovanie Apache s certifikátom
- nakonfigurovanie nginx
- vytvorenie autentifikačného modulu na backend-e

Dôvod nesplnenia: Úloha sa počas analýzy a implementácie ukázala zložitejšia ako pôvodný odhad z dôvodu integrácie viacerých technológií.

3.3 Šprint 3 (2.11. - 16.11.)

V poslednom šprinte boli všetky nedokončené úlohy z minulých šprintov prenesené. V tomto šprinte sme sa snažili dotiahnuť problémy s nasadením a frontend-ové úlohy.

Frontend - hlasovacia tabuľka a zobrazenie výsledkov

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Áno

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Implementácia frontend-u pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností.

Prvky grafického rozhrania:

- tabuľka
- statický text pre možnosti
- textové pole pre meno
- hlasovacie tlačidlá pre možnosti
- tlačidlo pre potvrdenie hlasovania

Frontend - vytvorenie hlasovacieho formulára

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Áno

Používateľsky príbeh: Vytvorenie jednoduchého hlasovania

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- Názov
- Možnosti hlasovania v textovom bloku pod sebou
- API volanie po potvrdení
- Vygenerovanie URL na admin a hlasovacie rozhrania

Frontend

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Áno

Používateľsky príbeh: Uzavrieť hlasovanie

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- API volania pre prerušenie hlasovania
- Nová admin obrazovka (názov hlasovania, jeho prerušenie)
- Testy

Nasadzovanie projektu cez Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Nie

Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.

Dôvod nesplnenia: Tesne pred dokončením vyplávala na povrch skutočnosť, že na serveri nie je možné spustiť end to end testy pre frontend. Kým teda neprebehne konfigurácia testov so serverovým prostredím, nebude možné vykonávať tieto testy pomocou Jenkins. Úloha je preto síce teoreticky splnená, ale po praktickej stránke je závislá na serverovom testovacom prostredí, takže kým nebude táto skutočnosť vyriešená, nie je možné považovať ju za splnenú, pretože riešenie počas testovania nevykonáva end to end frontend testy.

Nastavenie Apache na serveri

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Nastavenie Apache na serveri, aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou. Je potrebné otestovať komunikáciu s vytvoreným kontajnerom.

Frontend - úprava voľby hlasovania

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Áno

User story: Úprava voľby akéhokoľvek používateľa

Implementácia frontend pre úpravu voľby akéhokoľvek používateľa v hlasovacej tabuľke. Zahŕňa možnosť prepnutia voľby do stavu upravovania a uloženia zmien.

Frontend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

User story: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google OAuth:

- vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu
- odhlásenie používateľa
- udržiavanie stavu prihlásenia na strane klienta
- vytvorenie sekcie prístupnej len pre prihláseného používateľa
- integrácia riešenia s backend-om

Backend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Áno

User story: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google:

- vytvorenie používateľa cez Google účet (OAuth2)
- pridanie nášho hostname do overených hostanemov (Google)
- vytvorenie tajných kľúčov (Google)
- vygenerovanie SSL certifikátu

- nakonfigurovanie Apache s certifikátom
- nakonfigurovanie nginx
- vytvorenie autentifikačného modulu na backend-e

Frontend - Voľba hlasovacej stratégie

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

User story: Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.

Používateľ si po vybratí názvu hlasovania a zadania možností vyberie, ktorú z dostupných hlasovacích stratégií si chce vybrať.

- Pridanie kroku zvolenia stratégie do procesu vytvárania hlasovania.
- Automatická zmena popisu hlasovacej stratégie.

Backend - Voľba hlasovacej stratégie

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Andrej	Áno

User story: Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.

Doplní sa funkcionalita pri vytváraní hlasovania, ktorá používateľovi umožní výber konkrétnej hlasovacej stratégie. V tomto prípade ide o nastavenie backend-ovej časti.

Backend - Skript testovacej databázy

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Vytvorenie príkazov na vytvorenie testovacej databázy pre frontend s možným zálohovaním a obnovením databázy.

Metodika - vývoj a integrácia

Zodpovedný	Overovateľ	Splnené
Miroslav	Pavel	Áno

Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.

Metodika - agilný vývoj

Zodpovedný	Overovateľ	Splnené
Pavel	Elena	Áno

Vytvorenie metodiky pre riadenie vývoja softvéru, ktoré je založené na metodológii Scrum.

- role a zodpovednosti
- udalosti
- artefakty

Vytvoriť metodiku pre metodiky

Zodpovedný	Overovateľ	Splnené
Pavel	Elena	Áno

Vytvorenie metodiky pre zjednotenie vzhľadu a obsahu ostatných metodík.

Časti:

- Definícia a účel
- Štýl
- Obsah

Vytvoriť šablónu pre metodiky

Zodpovedný	Overovateľ	Splnené
Elena	Pavel	Áno

Vytvorenie LaTeX šablóny pre metodiky:

- Titulná strana
- Jazykové nastavenia
- Rozloženia a formátovanie

Návrh UX

Zodpovedný	Overovateľ	Splnené
Elena	Andrej	Áno

- Návrh farebnej schémy
- Pozadie pre aplikáciu
- Bez rozloženia elementov
- Branding

4 Globálna retrospektíva

Na lepšie pochopenie zmien v tíme a samotnej globálnej retrospektívy poskytujeme podrobný prehľad ku každému šprintu.

4.1 Retrospektíva k šprintu 1 (26.10.2017)

Problémy:

1. Máme vedúcich pre oblasti/sme rozdelení na podtímy (frontend, backend).
2. Hovoríme príliš dlho k jednotlivým bodom, aj keď už je zrejmá pointa.
3. Odbočujeme od témy.
4. Dlhو riešime veci ohľadom scrumu.
5. Chaos ohľadom story pointov.
6. V Jire je veľa taskov bez opisu.
7. Namiesto zápisnice písať veci do Jiry.
8. Zlé odhadovanie taskov.
9. Nie sme jedna masa. Nie sme my, ale každý je sám na probléme.
10. Chýba nám tímové cítenie.
11. Chýba nám disciplína. Častokrát meškáme.
12. Dôslednejšie a presnejšie metodiky a chaos v slacku.
13. Nestíha sa obed po tímovom stretnutí

Možné riešenia:

- Nebudeme sa tak oslovovať [k bodu 1].
- Gesto zdvihnutie ruky, aby niekto skončil/predtým ako bude niekto rozprávať dostane časový limit [k bodu 2].
- Napíšeme metodiku [všeobecne].
- Zápisnica je užitočná a ostáva [k bodu 7].
- Časom sa to zlepší. Nie je, že *ja* som nestihol, ale že *my* sme nestihli [k bodu 9 a 10].
- Pivo aspoň raz za mesiac [k bodu 10].
- Tresty (nosenie jedla za meškanie) [k bodu 11].
- Používať channely a thready [k bodu 12].
- Ješť na FIIT [k bodu 13].

Ďalšie pripomienky:

- Nebáť sa prehovoriť aj keď nevieme, aký je problém.

4.2 Retrospektíva k šprintu 2 (9.11.2017)

Pozitíva:

- Prijali sme problémy z minula.
- Zlepšenie disciplíny. Že pracujeme všetci.
- Neodbočujeme od tém.

- Spolupráca v tíme. Prejavilo sa to hlavne v integrácii.
- Spôsob komunikácie.
- Prístup k tomuto predmetu. Neberieme to ako nutné zlo.
- SSL certifikát.
- Ondrejov čas na retrospektívy (extra). Vedenie tímu.

Problémy:

1. Chýbajú priority taskov.
2. Chýbajú popisy taskov.
3. Neprečítali sme celú stránku.
4. Scrum poker je o bodoch.
5. Treba si písať viac progres do Jiry.
6. Utorok a Sobota statusy do Slacku (niekedy sa pozabudne).
7. Dlho čakali merge requesty.
8. Neskoro chodíme na stretnutia.
9. Závidíme si story pointy.

Možné riešenia:

- Mali by sme viac myslieť na ostatných pri vykonávaní svojich úloh (niekto na nás môže čakať) [všeobecne].
- Presunieme retrospektívu na štvrtok hneď po šprinte [všeobecne].
- Každý si bude viac upravovať tasky [k bodu 1 a 2].
- Budeme viac písať issues (gitlab) [všeobecne].
- Pripomienky do gmailu o statusoch [k bodu 6].
- Scrum master bude písať statusy ako prvý (pokúsi sa), pridá @channel [k bodu 6].
- Scrum master rozdá cukríky každému kto ho predbehne po 23:00 so statusom v utorok a sobotu [k bodu 6].
- Pri merge requeste budeme posúvať zodpovednosť ďalej (hľadať ďalšiu osobu) [k bodu 7].

4.3 Retrospektíva k šprintu 3 (16.11.2017)

Pozitíva:

- Sústredili sme sa na to, aby sme stihli šprint.
- Ela včas nahlásila to, že nestíha dokončiť svoju úlohu kvôli chorobe.
- Paľo venuje veľa času nastavovania a aktualizovaniu JIRA.
- Andrej inžinierske dielo.
- Miro ochotný komunikovať.
- Jenkins vyzerá sľubne, Lukáš aktívne komunikuje.
- Dokumentácia projektu od chalanov.
- Tomuto projektu venujeme veľa času.
- Zodpovedne makáme.
- Makáme jeden za druhého.
- Andrej stihol spraviť všetky svoje úlohy o dosť skôr ako na konci šprintu.
- Veľké množstvo pripomienok k metodike riadenia agilného vývoja od Ely.
- Zmena vnímania ohľadom písania metodík, zlepšujeme sa.
- Andrej je dobrý spolupracovník pri práci na UX.

- Rozpracovanosť metodík.
- Začali sme logovať prácu v JIRA.

Problémy:

1. Cítim sa ako keby som pracoval s ľuďmi v Amerike, lebo pracujeme v iných časových intervaloch (niekto cez deň, niekto neskoro v noci alebo skoro ráno).
2. Niektorí začínajú robiť na svojich úlohách až na konci šprintu.
3. Máme problém odkomunikovať prekážky v rámci našich úloh.
4. Pred implementáciou úloh nerobíme návrh.
5. Naháňame SP.
6. Treba komunikovať pri práci na úlohách, ktoré na seba nadväzujú.
7. Tomuto projektu venujeme veľa času.
8. Nevieme dobre naplánovať prácu na úlohách.
9. Neuvedomujeme si to, čo obnášajú niektoré úlohy.
10. Máme problémy so spoluprácou.
11. Hneď sme sa vrhli do implementácie.
12. Nenastavili sme si procesy.
13. Definition of done je dosť implicitný.
14. Chýba nám dokumentovanie úloh.
15. Potrebujeme sa lepšie pripravovať na stretnutia.
16. Pdlíšné vnímanie zodpovednosti JIRA administrátora.
17. Odlišné vnímanie toho, čo má obsahovať metodika pre agilné riadenie softvérového vývoja.
18. Nevydaril sa nám pokus s použitím papierikov počas plánovania šprintu.
19. vysoká technické požiadavky pre prácu s Angluarom spôsobujú neefektivitu pri vývoji na Windows platforme.
20. Na začiatku šprintu bol problém s reakciou na merge requesty.
21. Predmet TP nevnímame všetci rovnocenné.
22. Úprava cudzieho kódu podliehajúceho aktívnemu vývoju autora.
23. Nemáme dobré premyslený model použitia vetiev.

Možné riešenia:

- Mohli by sme skúsiť zaviesť nový typ tímového stretnutia, aby sme mali viac času osobne rozobrať riešenie úloh [k bodom 3, 6, 10].
- Nemali by sme riešiť TP po polnoci [k bodu 1].
- Mohli by vyskúšať párové programovanie [k bodom 6, 22].
- Mali by sme dohodnúť si deadline na časti tasku [k bodu 2].
- Všetci si máme vyskúšať úlohu scrum mastera [všeobecne].
- Ľudia zodpovední za úlohy v rámci jedného používateľského príbehu budú priebežne spolupracovať na dokumentovaní postupu riešenia [k bodom 6, 10, 22].
- Prioritizácia backlogu [všeobecne].
- Rozloženie práce na úlohách podľa *definition of ready* a *definition of done* [k bodom 2, 8, 12].
- Scrum master by mal viac komunikovať s členmi tímu, kontrolovať pokrok pri riešení úloh a pomáhať s odstraňovaním prekážok pri práci na nich, s tým, že by dostával menej implementačných úloh [k bodom 2, 3].
- Zlepšiť komunikáciu medzi ľuďmi (frontend, backend) [k bodom 6, 22].
- Je potrebné zjednotiť sa a nastaviť rovnaké priority [všeobecne].

- Je potrebné zaviesť development vetvu [k bodu 23].
- Je potrebné nahlasovať issues a nebáť sa o to, že to nie je issue [všeobecne].
- Zmeniť charakter úloh pre ďalší šprint – viac riešiť funkcionality, menej integráciu [všeobecne].
- Treba vyriešiť frontend testy - najprv treba spustiť v Docker na serveri [všeobecne].

5 Motivačný dokument

5.1 Tím

Už pri formovaní tímu sme sa dohodli, akým smerom sa chceme v práci neskôr uberať. Dávali sme si pozor na to, aby každý z nás mal rovnaké preferencie a ambície, ktoré sa stretli v oblasti orientovanej na strojové učenie, pri témach súvisiacich s umelou inteligenciou. Naša konečná zostava je plná ľudí s rôznymi skúsenosťami, avšak kolektívne najbohatšie sú práve v tejto oblasti, čo je podporované najmä našimi bakalárskymi prácami, ktoré sme vypracovali počas minulého akademického roka.

Pri výbere technológií, sme taktiež nemali problém zhodnúť sa na jednej, a to na Pythone. Ďalšou z našich silných stránok je skúsenosť s knižnicami najmä na umelú inteligenciu (zahŕňa pandas, numpy, nltk, scikit-learn atď.). Čo sa týka webových frameworkov, máme skúsenosti s pythonovským Django a taktiež Ruby on Rails - pričom preferujeme prvú možnosť. V rámci tvorby klientskych aplikácií máme skúsenosti s JavaScriptom a jeden člen tímu má bohaté skúsenosti s Angularom 2 a Ionicom.

V oblasti relačných databáz si najviac rozumieme s PostgreSQL a MySQL. Nerelačné databázy zahŕňajú Elasticsearch, Redis a MongoDB. Máme skúsenosti aj s nasadením aplikácie do produkcie, napríklad Heroku, DigitalOcean, AWS či Docker. Niektorí členovia majú skúsenosti s automatizovaným testovaním, konkrétne Selenium Framework a tiež vývojom mobilných aplikácií pre Android.

Niektorí z našich členov sa v minulosti aktívne podieľali na projektoch, ktoré využívali prvky metodológie agilného vývoja softvéru Scrum. Veríme, že tato skúsenosť nám pomôže ľahšie zvládnuť nábeh celého tímu na tento spôsob organizácie vývoja.

Naša voľba predmetov na zimný semester podporuje zameranie. U niektorých členov obsahuje Neurónové siete, Pokročilé databázové technológie, Spracovanie obrazu, grafika a multimédiá a u všetkých Vyhľadávanie informácií. Veríme, že nám najmä tento predmet pomôže nadobudnúť skúsenosti potrebné pre našu budúcu prax.

Sme skupina študentov, ktorí majú záujem vzdelávať sa a rozširovať svoje schopnosti a vedomosti. Štúdium na tejto fakulte rozhodne neberieme len ako cestu na získanie vysokoškolského titulu, ale najmä ako príležitosť získať poznatky a cenné skúsenosti s najnovšími technológiami, ktoré sa v bežnom živote ešte len budú využívať. Zároveň sa chceme aktívne podieľať na budovaní nástrojov zlepšujúcich kvalitu života bežného človeka.

Členovia nášho tímu medzi sebou v rôznych menších skupinkách (v závislosti podľa možnosti) už spolupracovali na viacerých spoločných projektoch. Okrem toho sa niektorí stretávajú aj v súkromnom živote, a to vedno považujeme za základ spolupatričnosti a dobrých vzťahov medzi nami. Práve takéto dobré vzťahy a vzájomná ochota, v kombinácii s rovnakou predstavou o smerovaní tímu, sú podľa nášho názoru najlepším predpokladom pre rozsiahlejšiu spoluprácu, aká je potrebná pri tímovom projekte.

5.2 Témy

Téma č. 2: Group de'Cider [Group]

Pri snahe čo najlepšie využiť náš vlastný čas a dohodnúť sa na spoločnom termíne stretnutia už aj v týchto raných fázach tímového projektu cítime obrovskú potrebu aplikácie, ktorá by bola výsledkom tohto projektu. To je práve najväčším dôvodom, prečo by sme sa radi stáli za jej návrhom a vznikom, aby mohla čo najvernejšie sledovať reálne potreby, ktoré my, tak ako aj neúrekom ďalších ľudí reálne máme. Pri takomto druhu aplikácie je navyše možnosť testovať ju už počas vývoja priamo pre potreby nášho všedného života.

Nemenej podstatný pre nás je aj fakt, že projekt ešte nezačal, a mohli by sme stáť za vznikom tejto aplikácie od jej samého počiatku, čo nám ponúka väčší priestor pre výber technológií a prostriedkov, ktoré by nám umožnili čo najlepšie presadiť silné stránky nášho tímu. Tento aspekt taktiež úzko súvisí aj s odporúčanými technológiami, ktoré v značnej miere pokrývajú oblasť našich preferencií a skúseností, no zároveň ponechávajú dostatočnú voľnosť pre potrebu prispôsobenia si konkrétnych technológií dokonale na mieru.

Viacerí z nás máme rozsiahle skúsenosti so súčasnými riešeniami, a práve preto sa vieme prirodzenejšie vyvarovať nedostatkom, ktoré najčastejšie majú. Tieto skúsenosti nám rovnako tak otvárajú cestu ku dôkladnej a hĺbkovej analýze existujúcich riešení, ktorá podľa nášho názoru zásadným spôsobom dopomôže ku čo možno najvyššej kvalite nášho návrhu.

Láka nás zavše aj výskumná časť tohto projektu, keďže máme všetci záujem aj o inovácie a prínosy. Užitočné predmety uvedené pri tomto projekte sme si už vopred zvolili do centra našej pozornosti, a preto máme za to, že by sme práve my mohli byť pre tento projekt aspoň takým prínosom, akým by bol on pre nás.

Stihli sme si už dokonca spraviť prieskum ohľadom hlasovania v súvislosti s teóriou hier a napadlo nám niekoľko potencionálnych smerov a vylepšení oproti aktuálnym riešeniam. Určite by sme chceli, aby sa projekt nevyužíval na značenie udalostí v kalendári, ale ich čerpal z už existujúcich riešení (napr. Google calendar, Facebook udalosti), a pritom vyžadoval od používateľa čo najmenšiu interakciu. Navrhujeme tiež, aby informácie vyťažené takýmto štýlom využíval maximálne efektívne, teda napríklad bral ohľad na miesto výskytu (keď je niekto od 13:00 do 14:00 v Košiciach, asi nebude môcť byť o 15:00 v Bratislave).

Téma č. 9: Odporúčanie pre e-biznis (Recommendation for eCommerce) [reCommers]

Táto téma v zásade zodpovedá všetkým bodom, ktoré sme si ako tím na začiatku vytýčili, že očakávame od témy. Menovite sa to týka predovšetkým backend oblastí, ktorá má byť v tomto projekte implementovaná v Pythone (a nám vyhovujúcom PostgreSQL), no zaujímavý je pre nás aj cieľ a ostatné technológie, ktoré tento projekt so sebou obnáša.

Veľmi nás motivuje fakt, že pri práci na tejto téme bude možné získať hlbšie poznatky a praktické skúsenosti z oblasti ako softvérové inžinierstvo, strojové učenie a vizualizácia dát. Čo nás na nej ale skutočne zaujalo, je priestor vypracovať sa v oblasti Big Data, keďže tento projekt je úzko spätý s technológiou Hadoop. Vieme, že táto technológia je momentálne na vzostupe a nielenže je veľmi príťažlivá, ale i prinajmenšom rovnako tak žiadaná. Na Slovensku je zatiaľ len málo produktov, ktoré ju plne využívajú, alebo ju aspoň využívajú rozumne.

Chceli by sme túto tému využiť na naučenie sa pracovania s touto, a rovnako tak s príbuznými technológiami, pričom by sme mali aj príležitosť pracovať s reálnymi dátami, čo sa nie vždy na akademickej pôde v rámci zadaní dá. Napriek mierne pozvoľnejšiemu začiatku, ktorý by určite táto pre nás nová technológia zapríčinila, by sme sa chceli plnohodnotne zapojiť do takéhoto projektu, a tým docieľiť požadovaný a nie len pre nás osožný výsledok.

Príloha A: Zoradenie všetkých tém podľa priority

2. Group de'Cider [Group]
9. Odporúčanie pre e-biznis (Recommendation for eCommerce) [reCommers]
24. Investment Portal [Invest]
3. Otvorené zmluvy: Budovanie prepojení vo verejných dátach [Zmluvy]
15. DeepSearch, alebo nájdeme to, čo práve potrebujete [DeepSearch]
4. Inteligentný bazár [IBazar]
26. Inteligentné parkovanie [SmartParking]
25. Artificial Intelligence: Voice Channel [VirtualAsist]
1. Importér verejných datasetov [PubDatasets]
20. Behaviorálna biometria na mobilných zariadeniach [Behametrics]
18. Rozpoznávanie cloudových služieb [OntoSEC]
13. Analýza správania sa používateľa v mobilných aplikáciách [Mob-UX]
12. Kolaboratívne prototypovanie používateľských rozhraní [Collab-UI]
11. Databanka otázok a úloh [FIIT-DU]
14. 3D UML, optimized version [3D-UML]
5. Monitorovanie a vyhodnocovanie fyziologických procesov človeka [StresMonitor]
8. Vzdelávanie vo Virtuálnej realite [EduVirtual]
7. Vizualizácia informácií v rozšírenej realite [VizReal]
10. Podpora diagnostiky [Look-Inside-Me]
19. Pohlcujúci Web [iWeb]
17. Vnorený systém monitorovania osôb [Breyslet 2.0]
23. Návrh systému MOD [Future MOD]
21. 3D simulovaný robotický futbal [3D futbal]
6. Internet vecí v našich životoch [IoT]
16. Softvérovo definované siete pre budúci Internet [SDN4Fut!]
22. Softvérovo riadené siete rozšírené o WiFi štandard [SDWN]
27. Digitalizácia pokladničných bločkov [Blocky]

Príloha B: Rozvrh tímu

Z rozvrhu je zrejmé, že máme spoločné voľno počas pondelka (preferujeme neskoré poobedie), vo štvrtok ráno, cez obed a večer. Vo všeobecnosti najviac preferujeme štvrtkové termíny, najvhodnejší sa javí blok od 11 do 15 hodiny - momentálne tam má jeden náš člen ešte cvičenie, ale pracujeme na jeho preložení (mali sme predvolené cvičenia študijným oddelením, preto nám nebolo umožnené prispôbiť si ich pre potreby tímu).

Pre lepšie zobrazenie podrobný rozvrh dostupný tu: [Rozvrhy](#)

Deň	Čas	7:00 - 7:55	8:00 - 8:55	9:00 - 9:55	10:00 - 10:55	11:00 - 11:55	12:00 - 12:55	13:00 - 13:55	14:00 - 14:55	15:00 - 15:55	16:00 - 16:55	17:00 - 17:55	18:00 - 18:55	19:00 - 19:55	20:00 - 20:55
Pon	Michal H														
	Elena														
	Michal D														
	Pavel														
	Lukáš														
	Miro														
	Andrej														
Ut	Michal H										VISS_I	TP1_I		ASS_I	
	Elena			PDT		VINF_I					VISS_I	TP1_I			
	Michal D			PDT		VINF_I					VISS_I	TP1_I			
	Pavel										VISS_I	TP1_I			
	Lukáš					VINF_I			SOGAM_I		VISS_I	TP1_I			
	Miro					VINF_I			SOGAM_I		VISS_I	TP1_I			
	Andrej										VISS_I	TP1_I			
Str	Michal H		NSIETE_I					ASS_I		MTS_I		MTS_I		VINF_I	
	Elena							ASS_I		MTS_I		MTS_I			
	Michal D							ASS_I		MTS_I		MTS_I			
	Pavel					ZKGRA_I		ASS_I		MTS_I		MTS_I		VINF_I	
	Lukáš					SOGAM_I		ASS_I		MTS_I		MTS_I			
	Miro					SOGAM_I		ASS_I		MTS_I		MTS_I			
	Andrej		NSIETE_I					ASS_I		MTS_I		MTS_I		VINF_I	
Štv	Michal H					PEWE				VINF_I		NSIETE_I			
	Elena					PEWE				VINF_I		ASS_I			
	Michal D									VINF_I		ASS_I			
	Pavel							ASS_I		VINF_I					
	Lukáš									VINF_I		ASS_I			
	Miro									VINF_I		ASS_I			
	Andrej									VINF_I		NSIETE_I			
Pia	Michal H														
	Elena			PDT				PDT							
	Michal D			PDT				PDT							
	Pavel											ZKGRA_I			
	Lukáš														
	Miro														
	Andrej					ASS_I									

- Prednáška
 - Cvičenie
 - Aktivita

Zhrnutie rozvrhov:

[illegible]

Záver

Pri práci na prvých troch šprintoch sa značne prejavil fakt, že náš tím nemá skúsenosti s prácou v Scrum. Úlohy z prvého šprintu boli značne podcenené, čo zapríčinilo ich presun do ďalších týždňov. Okrem toho nám trvalo podstatne dlhšie zvyknúť si štýl akým sú stretnutia plánovania vedené. Napriek všetkým týmto problém sa nám však podarilo splniť ciele zo všetkých troch šprintov.

Neskúsenosť sa prejavila aj v nástroji Jira, kde sme úlohy písali rôzne počas šprintu a nie v jednotnom čase. Tento fakt zapríčinil nekonzistenciu v burndown grafoch.

Veríme, že nasledujúce šprinty budú pokračovať plynulejšie. V poslednom šprinte sme kvôli časovej náročnosti a neefektívnosti značne zmenili granularitu plánovania a začali pridávať story pointy jednotlivým user stories. Pre zlepšenie burndown grafov budeme zapisovať úlohy naraz pred začiatkom šprintu.

Príloha A Export evidencie úloh

Summary	Issue key	Issue Type	Status	Assignee	Description	Epic name	SP
Nasadenie JIRA	GD-15	Task	Done	Andrej Vitek	Nasadenie kolaboračného systému JIRA na pridelený server.		3
Požiadat' o navýšenie RAM na tímovom serveri	GD-16	Task	Done	Andrej Vitek	Požiadat' P. Lacka o zvýšenie parametrov servera.		0.5
Metodika - práca s GitLab	GD-32	Task	Done	Andrej Vitek	Vytvorit' metodiku ku práci s GitLabom * vetvy a ich názvy * komentáre ku commit-u * merge request a komentáre		2
Návrh grafických rozhraní	GD-3	Task	Done	Elena Stefancova	2 TP Vytvorit' prvotné mockupy pre rozhrania + vygenerovať css.		2
Frontend - vytvorenie hlasovacieho formulára	GD-9	Task	In progress	Elena Stefancova	Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre: * Názov * Možnosti hlasovania v textovom bloku pod sebou * API volanie po potvrdení * Vygenerovanie URL na admin a hlasovacie rozhrania	Vytvorenie jednoduchého hlasovania	3
Vytvorenie mockupov pre aspoň 5 ďalších stratégií	GD-14	Task	In progress	Elena Stefancova	3 SP Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.		3
Prieskum ohľadom stratégií hlasovania	GD-13	Task	In progress	Lukas Babula	Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.		3
Inštalácia a konfigurácia Jenkins	GD-35	Task	Done	Lukas Babula	Nainštalovanie Jenkins na server. Nastavenie a zabezpečenie Jenkins. Nastavenie Gitlab notifikácie pre Jenkins. Vytvorenie Jenkins projektu. Nastavenie SCM v projekte. Rozšírenie Slacku o Jenkins agenta. Integrácia Slack kanála do Jenkins.		8
Definícia POST/GET	GD-4	Task	Done	Michal Dolnak	Definovanie rozhrania a procesu tvorby rozhrania medzi frontendom a backendom.		2
Frontend - infraštruktúra	GD-8	Task	Done	Michal Dolnak	Vytvorenie nového projektu a nastavenie technológie. Pridanie prekladov, volania backendu. Nastavenie ciest v aplikácii.		3
Frontend - zaučenie, zdokumentovanie	GD-11	Task	Done	Michal Dolnak	Podpora ostatných členov tímu pri svojich úlohách. Predanie svojich vedomostí o technológií.		2
Metodika - dokumentovanie frontend kódu	GD-31	Task	Done	Michal Dolnak	2 SP		2
Implementácia backend v Django REST	GD-6	Task	Done	Michal Hucko	Implementácia základného vytvárania udalostí spolu s funkcionalitou hlasovania. Bude sa dať vytvorit' hlasovanie spolu s množinou možností. Následne je možné anonymne hlasovať za jednu z nich. Vytvorenie prihlášky na TP cup, ktorá bude zahŕňať všetky body zo stránky predmetu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.	Vytvorenie jednoduchého hlasovania	13
Prihlásenie sa do TP Cup	GD-24	Task	Done	Michal Hucko	Navrhnutie dátového modelu pre funkcionalitu a jeho vizualizácia pomocou UML diagramu.		5
Návrh DB schémy	GD-5	Task	Done	Miroslav Valcicak	Vytvorenie konfiguračných súborov pre docker-compose a jednotlivé moduly.		1
Docker setup	GD-7	Task	Done	Miroslav Valcicak	Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností.		8
Frontend - hlasovacia tabuľka a zobrazenie výsledkov	GD-10	Task	In progress	Pavel Balashov	Prvky grafického rozhrania: * tabuľka * staticky text pre možnosti * textové pole pre meno * hlasovacie tlačidlá pre možnosti * tlačidlo pre potvrdenie hlasovanie	Vytvorenie jednoduchého hlasovania	8
Migrovanie úloh z Asana do JIRA	GD-17	Task	Done	Pavel Balashov	Migrovanie úloh z Asana. Zahŕňa prepísanie úloh do JIRA a kontrolu ďalším členom tímu.		1
Vytvorenie tímového webu	GD-67	Task	Done	Lukas Babula	Vytvorenie tímovej stránky * rozloženie elementov * informácie o time a projekte * sprinty * dokumenty * kontakt		1
Vytvorenie tímového webu	GD-66	Task	Done	Andrej Vitek	Vytvorenie tímovej stránky * rozloženie elementov * informácie o time a projekte * sprinty * dokumenty * kontakt		1

Export pre šprint 1 - Ananás

Summary	Issue key	Issue Type	Status	Assignee	Description	Epic name	SP
Backend - Uprava voľby hlasovania	GD-53	Task	Done	Andrej Vítek	Backend pre zmenu hlasovania používateľa cez HTTP PUT metódu. Umožniť ukončenie hlasovania cez backend * uprava modelu hlasovania * vytvorenie url pre administráciu * zabezpečiť, že sa nedá hlasovať na uzavretom hlasovaní * API volania pre prerušenie hlasovania * Nová admin obrazovka (názov hlasovania, jeho prerušenie) * Testy 2 TP	Uprava voľby akéhokoľvek používateľa	3
Backend - Ukončenie hlasovania	GD-69	Task	Done	Andrej Vítek		Ukončenie hlasovania	3
Frontend - vytvorenie hlasovacieho formulára	GD-70	Task	In progress	Elena Stefancova		Vytvorenie jednoduchého hlasovania	3
Návrh grafických rozhraní	GD-3	Task	Done	Elena Stefancova	Vytvoríť prvotné mockupy pre rozhrania + vygenerovať css. 3 SP		2
Vytvorenie mockupov pre aspoň 5 ďalších stratégií	GD-14	Task	In progress	Elena Stefancova	Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.		3
Nasadzovanie projektu cez Jenkins	GD-48	Task	In progress	Lukas Babula	Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.		13
Prieskum ohľadom stratégií hlasovania	GD-13	Task	In progress	Lukas Babula	Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.		3
Metodika - testovanie frontendu	GD-29	Task	Done	Michal Dolnak	Vytvorenie metodiky pre testovanie frontendu. Vytvorenie prihlasovania cez Google Oath: - vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu - odhlásenie používateľa - udržiavanie stavu prihlásenia na strane používateľa - vytvorenie sekcie prístupnej len pre prihláseného používateľa		2
Frontend - Vytvorenie prihlasovania	GD-58	Task	In progress	Michal Dolnak	V tejto metodike pokryjeme všetky detaily spojené s testovaním backendu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.	Vytvorenie prihlasovania	8
Metodika - testovanie backendu	GD-26	Task	Done	Michal Hucko	V tejto metodike pokryjeme všetky detaily spojené s dokumentáciou backendu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.		2
Metodika - dokumentovanie backend kódu	GD-30	Task	Done	Michal Hucko	Nastavenie Apache na serveri aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.		8
Nastavenie Apache na serveri	GD-49	Task	In progress	Michal Hucko	Napísať Peťovi Lackovi o zvýšení RAMky na servery a zistiť či je možné získať SSL certifikát pre projekt.		1
Vyjadniť navýšenie RAM + certifikát	GD-50	Task	Done	Michal Hucko	Zodpovedná osoba dokonfiguruje docker na serveri. Tento krok predchádza samotnému spusteniu dockeru na serveri.		3
Dokonfigurovať Docker	GD-51	Task	Done	Michal Hucko	Úloha zahŕňa zodpovednosť nad tým aby sa stihli spojiť kódy frontendu a backendu. V prípade akýchkoľvek problémov zodpovedná osoba poskytne pomoc.		2
Merge do master	GD-47	Task	Done	Michal Hucko			
Metodika - vývoj a integrácia	GD-34	Task	In progress	Miroslav Valcicak	Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie. Vytvorenie prihlasovania cez Google * vytvorenie používateľa cez Google účet (OAuth2) * pridanie nášho hostname do overených hostnameov (Google) * vytvorenie tajných kľúčov (Google) * vygenerovanie SSL certifikátu * nakonfigurovanie Apache s certifikátom * nakonfigurovanie nginx * vytvorenie autentifikačného modulu na backende	Vytvorenie prihlasovania	13
Backend - Vytvorenie prihlasovania	GD-57	Task	In progress	Miroslav Valcicak	Implementácia frontend pre úpravu voľby akéhokoľvek používateľa v hlasovacej tabuľke. Zahŕňa možnosť prepnutie voľby do stavu upravovania a uloženie zmien.	Vytvorenie jednoduchého hlasovania	3
Frontend - hlasovacia tabuľka a zobrazenie výsledkov	GD-54	Task	In progress	Pavel Balashov			

Export pre šprint 2 - Banán

Summary	Issue key	Issue Type	Status	Assignee	Description	Epic name	SP
Backend - Skript testovacej databázy	GD-87	Task	Done	Andrej Vitek	Vytvorenie prikazov na vytvorenie testovacej databázy pre frontend s možným zálohovaním a obnovením databázy.		8
Vytvoriť šablónu pre metodiky	GD-76	Task	Done	Elena Stefancova	Vytvorenie latex šablóny pre metodiky * Titulná strana * Jazykové nastavenia * Rozloženia a formátovanie		3
Návrh UX	GD-88	Task	Done	Elena Stefancova	* Návrh farebnej schémy * Pozadie pre aplikáciu * Bez rozloženia elementov * Branding		13
Frontend - vytvorenie hlasovacieho formulára	GD-70	Task	Done	Elena Stefancova	* API volania pre prerušenie hlasovania * Nová admin obrazovka (názov hlasovania, jeho prerušenie) * Testy	Vytvorenie jednoduchého hlasovania	3
Nasadzovanie projektu cez Jenkins	GD-48	Task	In progress	Lukas Babula	Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.		13
Frontend - Voľba hlasovacej stratégie	GD-92	Task	Done	Michal Dolnak	Používateľ si po vybratí názvu hlasovania a zadania možnosti vyberie, ktorú z dostupných hlasovacích stratégií si chce vybrať. * Pridanie kroku zvolenia stratégie do procesu vytvárania hlasovania * Automatická zmena popisu hlasovacej stratégie.	Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.	8
Frontend - Vytvorenie prihlasovania	GD-58	Task	Done	Michal Dolnak	Vytvorenie prihlasovania cez Google Oath: - vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu - odhlásenie používateľa - udržiavanie stavu prihlásenia na strane používateľa - vytvorenie sekcie prístupnej len pre prihláseného používateľa	Vytvorenie prihlasovania	8
Backend - Voľba hlasovacej stratégie	GD-91	Task	Done	Michal Hucko	Doplní sa funkcionálna pri vytváraní hlasovania, ktorá používateľovi umožní vyber konkrétnej hlasovacej stratégie. V tomto prípade ide o nastavenie backendovej časti.	Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.	8
Nastavenie Apache na serveri	GD-49	Task	Done	Michal Hucko	Nastavenie Apache na serveri aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.		8
Metodika - vývoj a integrácia	GD-34	Task	Done	Miroslav Valcicak	Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.		3
Backend - Vytvorenie prihlasovania	GD-57	Task	Done	Miroslav Valcicak	Vytvorenie prihlasovania cez Google * vytvorenie používateľa cez Google účet (OAuth2) * pridanie nášho hostname do overených hostnameov (Google) * vytvorenie tajných kľúčov (Google) * vygenerovanie SSL certifikátu * nakonfigurovanie Apache s certifikátom * nakonfigurovanie nginx * vytvorenie autentifikačného modulu na backende	Vytvorenie prihlasovania	13
Metodika - agilný vývoj	GD-33	Task	Done	Pavel Balashov	Vytvorenie metodiky pre riadenie vývoja softveru založeného na metodológii Scrum. * roli a zodpovednosti * priebeh stretnutí * práca s JIRA		3
Vytvoriť metodiku pre metodiky	GD-77	Task	Done	Pavel Balashov	Vytvorenie metodiky pre zjednotenie vzhľadu a obsahu ostatných metodík. Časť: * Definícia a účel * Štýl * Obsah		2
Frontend - hlasovacia tabuľka a zobrazenie výsledkov	GD-54	Task	Done	Pavel Balashov	Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností. Prvky grafického rozhrania: * tabuľka * statický text pre možnosti * textové pole pre meno * hlasovacie tlačidlá pre možnosti * tlačidlo pre potvrdenie hlasovanie	Vytvorenie jednoduchého hlasovania	3
Frontend - úprava voľby hlasovania	GD-54	Task	Done	Pavel Balashov	Implementácia frontend pre úpravu voľby akéhokoľvek používateľa v hlasovacej tabuľke. Zahŕňa možnosť prepnutie voľby do stavu upravovania a uloženie zmien.	Úprava voľby akéhokoľvek používateľa	3

Export pre šprint 3 - Citrón

Príloha B Metodiky

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Tímový projekt
Group de'Cider
Metodiky

Tím: číslo 1, sElectors

Vedúci projektu: Ing. Ondrej Kaššák

Členovia tímu: Bc. Lukáš Babula, Bc. Pavel Balashov, Bc. Michal Dolnák, Bc. Michal Hucko, Bc. Elena Štefancová, Bc. Miroslav Valčíčák, Bc. Andrej Vítek

Zodpovedná osoba: Bc. Elena Štefancová

Posledná aktualizácia: 14.11.2017

Obsah

Zoznam pojmov a skratiek	3
1 PÍSANIE METODÍK	5
1.1 Role	5
1.2 Definícia, účel a platnosť	5
1.3 Štýl	5
1.4 Štruktúra dokumentu	6
2 AGILNÉ RIADENIE SOFTVÉROVÉHO VÝVOJA	7
2.1 Role	7
2.2 Udalosti	8
2.2.1 Šprint	8
2.2.2 Hlásenie statusu	8
2.2.3 Tímové stretnutia	8
2.2.4 Plánovanie nového šprintu	8
2.2.5 Stretnutie v polovici šprintu	9
2.2.6 Vyhodnotenie šprintu	9
2.2.7 Retrospektíva šprintu	9
2.3 Artefakty	10
2.3.1 Produktový backlog	10
2.3.2 Backlog pre šprint	10
2.3.3 Položka backlogu	10
2.3.4 Cieľ šprintu	10
3 PRÁCA S VERZIAMI	11
3.1 Role	11
3.2 Vetvy	11
3.3 Commit-y	12
3.4 Merge request-y	12
4 VÝVOJ A INTEGRÁCIA	13
4.1 Role	13
4.2 Opis procesov	13
4.3 Záver	15
5 DOKUMENTÁCIA	16
5.1 Role	16
5.2 Predpoklady	16
5.2.1 Spustenie frontendu	16
5.2.2 Spustenie backendu	17
5.3 Opis procesu	18
5.3.1 Frontend	18
5.3.2 Backend	18

5.4	Záver	19
6	TESTOVANIE	20
6.1	Role	20
6.2	Definition of done z pohľadu testovania	20
6.3	Predpoklady	20
6.3.1	Frontend	20
6.3.2	Načítanie testovacích údajov cez docker	21
6.3.3	Backend	21
6.4	Opis procesu	21
6.4.1	Frontend	21
6.4.2	Backend	22
6.5	Záver	23

Zoznam pojmov a skratiek

backend serverová časť webovej aplikácie

backlog usporiadaný zoznam úloh, ktoré ešte nie sú vyriešené

camel case spôsob písania názvov bez medzier, kde prvý výraz začína malým písmenom každé nové slovo veľkým (napr. camelCaseWriting)

code review prehliadka kódu, statická analýza kódu - proces, kedy zodpovedná osoba skontroluje kód, tento proces je súčasťou definition of done

commit vytvorenie novej verzie s vybranými zmenami

cors celým názvom ang. *cross-origin resource sharing*, mechanizmus umožňujúci zdieľanie zdrojov webovej stránky pre aplikáciu na inej doméne

definition of done zoznam bodov, ktoré je potrebné splniť na dokončenie úlohy

docker nástroj, ktorý sa používa pri nasadzovaní riešenia do produkcie¹

e2e (end to end) test, ktorý pokrýva všetky časti aplikácie

frontend vizuálna časť webovej aplikácie

kontinuálna integrácia spôsob vývoja, pri ktorom všetci vývojári zdieľajú svoj kód vo vzdialenom repozitári, kód sa automaticky zostaví a otestuje

kontinuálne nasadzovanie rozšírenie kontinuálnej integrácie o automatické nasadenie do produkčného prostredia po úspešnom otestovaní

merge request žiadosť o spojenie vetiev

mockup používateľské rozhranie slúžiace len na demonštráciu bez implementovania funkčnej logiky

ng angular klient

npm celým názvom ang. *node package manager*, správca balíčkov uzla

osx operačný systém pre počítače spoločnosti Apple

používateľsky príbeh (angl. *user story*) Opis časti funkcionality aplikácie z pohľadu interakcie medzi používateľom a aplikáciou

REST celým názvom ang. *representational state transfer*, spôsob ako jednoducho vytvoriť, čítať, editovať alebo zmazať informácie zo serveru pomocou jednoduchých volaní

¹<https://www.docker.com/>

Scrum metodológia agilného vývoja softvéru

Selenium rámec na testovanie frontendu

snake case spôsob písania názvov bez medzier, kde prvý výraz začína malým písmenom každé nové slovo malým, slová sú rozdelené podtržníkmi (napr. snake_case_writing)

Swagger nástroj na dokumentovanie REST rozhraní²

tag tzv. značka, pomocné označenie, ktoré zlepšuje čitateľnosť

URL celým názvom ang. *uniform resource locator*, jednotný vyhľadávač prostriedku

view časť backendu, ktorá posiela dáta na frontend v určenom formáte

wireframe bezfarebný náčrt štruktúry, obsahu a rozmiestnenia prvkov grafického používateľského rozhrania

²<https://swagger.io/>

Kapitola 1

PÍSANIE METODÍK

Zodpovedná osoba: Bc. Pavel Balashov

Posledná úprava: 19.11.2017, verzia 1

Tato metodika je určená pre členov tímu sElectors, ktorý pracuje na projekte Group de'Cider v rámci predmetu Tímový projekt.

Tato metodika bola vytvorená pre upresnenie toho, čo je metodika a na čo sa používa. Taktiež v tejto metodike sa opisuje štýl písania a štruktúra, ktoré musia dodržiavať metodiky vytvorené a používané členmi tímu sElectors v rámci projektu Group de'Cider.

1.1 Role

Autor metodiky - člen tímu, ktorý napísal danú metodiku. Iba autor metodiky má možnosť meniť a dopĺňať metodiku.

1.2 Definícia, účel a platnosť

Metodika je dokument ktorý zachytáva skúsenosti a znalosti jedného alebo viacerých členov tímu v rámci konkrétnej oblasti vývoja, riadenia vývoja alebo technológií. Je potrebná pre interné štandardizovanie procesov, ktoré môžu alebo musia byť vykonané v rámci projektu. Metodika slúži pre zaistenie kvality vykonania procesov, bez ohľadu na to, kto ho vykonáva, a je nápomocná najmä v prípade, že existujú alternatívne možnosti a postupy¹.

V rámci tímu sElectors, ktorý pracuje na projekte Group de'Cider, metodika nadobúda účinnosť po ukončení šprintu, v rámci ktorého bola schválená. Po nadobudnutí účinnosti je metodika záväzná pre všetkých členov tímu.

Všetky návrhy pre zmenu alebo doplnenie metodiky sa píšú do komentárov. Ak sa nejedná o opravu gramatickej chyby, tieto zmeny a doplnenia musia byť následne schválené tímom. Po schválení ich autor metodiky doplní do metodiky a zvýrazní žltou farbou, čo znamená, že ešte nie sú platné. Tieto zmeny a doplnenia sa stávajú platnými po ukončení šprintu, v rámci ktorého boli schválené. Po nadobudnutí platnosti autor odstráni zvýraznenie.

1.3 Štýl

Metodika je formálny technický dokument, ktorý má obsahovať:

- jednotný strohý technický presný jazyk
- jednoznačné, zrozumiteľné a explicitné formulácie

¹<http://www2.fkit.stuba.sk/bielik/courses/msi-slov/metodika/metodika2012-13.pdf>

Metodika by mala mať odporúčací alebo normatívny charakter. Pri písaní metodiky by sa mal autor úplne odosobniť, čo znamená:

- nepoužívať vlastné hodnotenia
- nepoužívať osobné zámená (ja, my, vy atď.)
- používať neurčitky a neosobné zámená

Metodika musí byť zrozumiteľná aj čitateľom s nižšou úrovňou znalostí než akou disponuje jej autor. Podľa možnosti, autor by mal uviesť príklad vykonávania konkrétneho procesu a pridať aj snímky obrazoviek.

Nakoniec, metodika nemá obsahovať analýzy problémov, názory a hodnotenia, avšak môže sa odkazovať na dokumenty, ktoré ich obsahujú².

1.4 Štruktúra dokumentu

1. titulná strana
2. obsah
3. zoznam pojmov a skratiek
4. nadväznosť na iné metodiky
5. úvod
 - (a) cieľová skupina - stanovuje pre koho je metodika určená
 - (b) ciele - vymedzenie cieľov metodiky
6. role a ich zodpovednosti
7. opis procesov
 - (a) opis postupu – opis všetkých podstatných povinných a podmienených krokov, ktoré je alebo môže byť potrebné vykonať
 - (b) vyhodnotenie kvality – ako zistiť, že opísaný postup bol správne dodržaný
 - (c) doplňujúce zdroje informácií – napr. odkazy na detailnejší opis riešenia problému na internete alebo dokumentáciu konkrétnej časti technológie
 - (d) hľadanie a odstránenie závad (ang. *troubleshooting*) – detaily toho, kedy môže nastať konkrétna záhada a ako sa dá vyriešiť
8. zdroje

²<http://www2.fkit.stuba.sk/bielik/courses/msi-slov/metodika/metodika2012-13.pdf>

Kapitola 2

AGILNÉ RIADENIE SOFTVÉROVÉHO VÝVOJA

Zodpovedná osoba: Bc. Pavel Balashov

Posledná úprava: 19.11.2017, verzia 1

Táto metodika je určená pre členov tímu sElectors, ktorý pracuje na projekte Group de'Cider.

V rámci tímu bolo potrebné sa zhodnúť na prístupe k riadeniu projektu založenom na metodológii agilného vývoja softvéru Scrum. Najmä vznikla potreba jednoznačne definovať role, udalosti, artefakty a pravidlá podľa uvedenej metodológie, ako aj rozšíriť ich o ďalšie detaily, ktoré sú špecifické pre tím sElectors a projekt Group de'Cider.

2.1 Role

Produktový vlastník -

- vízia a definícia produktu¹
- manažment položiek produktového backlogu
- prioritizácia položiek produktového backlogu
- akceptácia implementácie

Scrum master - rola scrum master je pridelená jednému z členov tímu. Pred začiatkom nového šprintu sa táto rola prideľuje inému členovi.

- dodržiavanie Scrum a agilných princípov
- identifikácia a odstránenie problémov, ktoré ovplyvňujú výkon tímu
- sledovanie progresu vývoja
- organizácia a vedenie stretnutí

Tím -

- spôsob vývoja produktu
- tvorba plánu šprintu
- odhad náročnosti práce
- predvedie výsledkov šprintu
- zlepšenia procesu

¹<http://scrum.sk/o-agile-a-scrum/2-scrum-a-roly/>

2.2 Udalosti

Všetky udalosti sú časovo ohraničené.

2.2.1 Šprint

Šprint (angl. *sprint*) je udalosť, počas ktorej sa vyvíja ďalší plne funkčný a použiteľný inkrement funkcionality produktu. V rámci šprintu sa uskutočňujú ďalšie udalosti ako tímové stretnutia a hlásenia statusu.

Šprint má fixnú dĺžku 2 týždne. Pred šprintom prebehne plánovanie šprintu, po ukončení ktorého začne šprint. Počas šprintu:

- nemôže sa meniť cieľ šprintu
- nemôžu byť vykonané zmeny, ktoré ohrozia cieľ šprintu
- backlog pre šprint môže byť upravený po prediskutovaní s produktovým vlastníkom

Šprint končí vyhodnotením šprintu.

2.2.2 Hlásenie statusu

Hlásenie statusu je časový úsek, kedy každý člen tímu musí oznámiť pokrok na svojich úlohách a problémy, ktoré ho aktuálne brzdia. Je alternatívou pre denné scrumy (angl. *daily scrums*).

Hlásenie statusu sa koná minimálne 2 krát do týždňa - v utorok a sobotu. V stanovených dňoch je každý člen tímu povinný do 23:00 napísať do Slack kanálu #status:

- názvy úloh na ktorých pracuje
- vyjadrenie pokroku (0 % - 99 %, ready for review, done)
 - ak je pokrok uvedený v percentách a nie je 0 %, je potrebné do zátvorky krátko opísať čo ešte ostalo dorobiť
 - ak niektorá úloha nadväzuje na inú, je vhodné jej status doplniť o informáciu, na ukončenie ktorej úlohy pod koho vedením sa čaká

2.2.3 Tímové stretnutia

Tímové stretnutia vyžadujú prítomnosť všetkých členov tímu a produktového vlastníka. Neúčast je povolená iba vo výnimočných situáciách a musí byť vopred oznámená všetkým členom tímu.

Trestom za nedochvilnosť je povinnosť priniesť chutnú odmenu pre ostatných členov tímu na ďalšom stretnutí.

Scrum master vytvorí súpis tém, ktoré sa budú preberať na tímovom stretnutí a zverejní ho ostatným členom tímu najneskôr deň pred stretnutím. Toto neplatí pre retrospektívu šprintu.

2.2.4 Plánovanie nového šprintu

Počas plánovania nového šprintu (angl. *sprint planning*) tím spoločnými silami rozvrhne prácu pre ďalší šprint. Toto stretnutie zahŕňa 3 podstatné činnosti:

- stanovenie cieľa šprintu
- výber úloh z produktového backlogu, ktoré budú vykonané počas nového šprintu
- ako bude prebiehať práca na vybraných úlohách

Pri výbere úloh do backlogu nového šprintu berie tím ohľad na: prioritu úloh, rýchlosť tímu a predchádzajúcu skúsenosť členov tímu. O počte úloh, ktoré budú vybrané do nového šprintu, rozhoduje tím.

Po výbere úloh do backlogu nového šprintu vytvorí tím plán práce. Tento plán bude obsahovať nadväznosť úloh a približný odhad kedy sa bude pracovať na ktorej úlohe. Pre úlohy, na ktorých sa plánuje pracovať na začiatku šprintu, bude vytvorený detailný opis, ktorý musí obsahovať kroky vývoja a akceptačné kritériá. Pre ostatné úlohy môže byť spísaný krátky opis.

Po vytvorení opisov tím vyhodnotí náročnosť každej položky backlogu pre nový šprint. Pre tento účel prebehne plánovací poker (angl. *planning poker*). Nakoniec každej úlohe, na ktorej sa plánuje pracovať na začiatku šprintu, bude pridelený člen tímu, ktorý bude za ňu zodpovedný. Ostatné úlohy môžu, ale nemusia mať prideleného člena tímu.

Plánovanie nového šprintu prebieha každý druhý štvrtok o 12:00, spolu s vyhodnotením predchádzajúceho šprintu. Spolu tieto činnosti môžu trvať maximálne 3 hodiny.

2.2.5 Stretnutie v polovici šprintu

Stretnutie v polovici šprintu sa využíva pre:

- riešenie prekážok, ktorým čelia členovia tímu počas práce na položkách backlogu pre aktuálny šprint
- prediskutovanie úpravy backlogu pre aktuálny šprint s produktovým vlastníkom
- pridelenie zodpovedného člena tímu úlohám, ktoré ho nemali prideleného počas plánovania šprintu, ako aj úprava a doplnenie ich opisu
- ďalšie činnosti, ktoré vyžadujú vstup od všetkých členov tímu a ktoré nie je možné alebo sa nedajú efektívne vykonať mimo stretnutia

Toto tímové stretnutie prebieha každý druhý štvrtok o 12:00 a môže trvať maximálne 3 hodiny.

2.2.6 Vyhodnotenie šprintu

Na začiatku vyhodnotenia šprintu (angl. *sprint review*) sa posúdi práca, ktorá bola vykonaná počas šprintu a prezentovanie funkčného inkrementu, ktorý mal vzniknúť na konci šprintu. Ak na konci šprintu ostali nesplnené niektoré úlohy, počas tohto stretnutia sa posúdi ich stav a potrebnosť ich úpravy.

Potom tím spolu s produktovým vlastníkom vyhodnotí stav produktového backlogu a dohodne sa na ďalšej práci a smerovaní projektu. Výsledkom tohto stretnutia by mal byť revidovaný produktový backlog, ktorý je pripravený pre plánovanie ďalšieho šprintu.

Vyhodnotenie šprintu prebieha každý druhý štvrtok o 12:00, spolu s plánovaním nového šprintu. Spolu tieto činnosti môžu trvať maximálne 3 hodiny.

2.2.7 Retrospektíva šprintu

Retrospektíva šprintu (angl. *sprint retrospective*) slúži pre vyhodnotenie pozitívnych a negatívnych momentov šprintu, ktorý bol práve ukončený, s ohľadom na ľudí, vzťahy, činnosti a nástroje. Na základe nich tím spoločne sformuluje možné úpravy procesu riadenia a vývoja projektu, ktoré sa bude snažiť vykonať v ďalších šprintoch. Ako osnova slúžia tri otázky, na ktoré odpovie každý z členov tímu:

- Čo sa mi v tomto šprinte páčilo?
- Čo sa mi v tomto šprinte nepáčilo?
- Aké zmeny by som navrhol pre vylepšenie budúcich šprintov?

Retrospektíva šprintu prebieha každý druhý štvrtok o 18:40, v rovnaký deň ako prebehlo vyhodnotenie šprintu. Toto stretnutie trvá maximálne 2 hodiny.

2.3 Artefakty

2.3.1 Produktový backlog

Produktový backlog (angl. *product backlog*) je usporiadaný zoznam všetkých úloh, ktoré musia byť vykonané. Je to zdroj všetkých požiadaviek a zmien, ktoré musia byť zahrnuté do produktu. Keďže požiadavky na produkt sa stále menia, produktový backlog sa môže meniť. Podľa potreby, úlohy v produktovom backlogu môžu byť rozbité na menšie úlohy.

2.3.2 Backlog pre šprint

Backlog pre šprint (angl. *sprint backlog*) pozostáva z položiek produktového backlogu, ktoré boli vybrané tímom pre aktuálny šprint, a plánu práce pre dodanie ďalšieho inkrementu a dosiahnutie cieľu šprintu. Informácie pre položky tohto backlogu sa postupne dopĺňajú počas šprintu.

2.3.3 Položka backlogu

Každá položka backlogu definuje úlohu, ktorá musí byť vykonaná.

Pre všetky úlohy sú definované podmienky, ktoré musia byť splnené pred tým ako úloha môže byť považovaná za hotovú (tzv. *definition of done*), ktorý je tvorený nasledovnými činnosťami:

1. implementácia
2. testovanie (jednotkové, integračné)
3. dokumentácia kódu
4. (v prípade používateľského príbehu) doplnenie používateľskej dokumentácie
5. posudok od aspoň jedného ďalšieho člena tímu (napr. code review)
6. oznámenie všetkým členom tímu
7. (v prípade používateľského príbehu) prezentácia produktovému vlastníkovi
8. (v prípade používateľského príbehu) schválenie produktovým vlastníkom
9. nasadenie do produkčného prostredia (merge do master vetvy a riešenie konfliktov)

Taktiež každá úloha môže mať špecifické akceptačné kritéria, ktoré musia byť splnené pred tým, ako sa úloha bude považovať za hotovú.

2.3.4 Cieľ šprintu

Cieľ šprintu (angl. *sprint goal*) je stanovený na to, aby členovia tímu nepracovali na vlastných úlohách izolovane, ale spolupracovali pre splnenie spoločného cieľa. Počas šprintu je tento cieľ realizovaný úlohami, ktoré boli vybrané do backlogu pre aktuálny šprint a ďalšími súvislosťami, ktoré motivujú tím pracovať spolu.

Kapitola 3

PRÁCA S VERZIAMI

Zodpovedná osoba: Bc. Andrej Vitek

Posledná úprava: 19.11.2017, verzia 1

Cieľom tejto metodiky je definovať spôsob práce s git repozitárom v rámci tímového projektu a tým dosiahnuť prehľadné verziovanie kódu nezávisle od člena tímu, ktorý zmeny vykonáva. Metodika sa zaoberá spôsobom práce s vetvami, tvarom commit správ a spôsobom ako spájať už dokončenú funkcionality vo vetve s hlavnou vetvou.

3.1 Role

Zodpovedný člen tímu - osoba ktorá je zodpovedná za vykonané zmeny.

Reviewer - osoba zodpovedná za code review.

3.2 Vetvy

Pre zjednodušenie vývoja sa budú pre nové funkcionality a riešenia v projekte vytvárať vetvy. Úlohy, pre ktoré sa budú vytvárať nové vetvy, by mali byť na úrovni „story“ alebo náročnejších úloh. Názvy vetiev sa budú skladať z 3 častí:

- typ funkcionality - podľa riešenej úlohy obsahuje buď „feature“ alebo „bug_fix“,
- časť projektu - ktorá časť projektu je riešená (backend, frontend, team_web...),
- názov - napísane malým písmenom v angličtine a v prípade viacerých slov, budú slová oddelené podčiarkovníkom (znak "_").

Názov vetvy sa bude skladať spojením týchto 3 častí pričom časti budú oddelené lomítkom (znak "/"). Vzor názvu vetvy: typ_funkcionality/časť_projektu/názov

Vetvy sa prehadzujú pomocou príkazu „git checkout nazov_vetvy“ a vytvárajú pomocou príkazu "git checkout -b nazov_vetvy". Nová vetva sa vytvorí na základe aktuálne vybranej vetvy.

Správne názvy:

- bug_fix/backend/<bug_xy>
- feature/frontend/<feature_xy>

Nesprávne názvy:

- feature-BackEnd-Feature-xy

Po ukončení práce sa pre danú vetvu vytvorí merge request.

3.3 Commit-y

Po ukončení práce na časti funkcionality sa vytvorí commit obsahujúci krátku správu popisujúcu, čo bolo zmenené. Správa bude obsahovať do 10 slov v anglickom jazyku v imperatívne. Prvé slovo bude začínať veľkým písmenom.

Commity sa vytvárajú pomocou príkazu "git commit -m "Správa"".

Správne názvy:

- Add table
- Fix bug xy

Nesprávne názvy:

- fixed bug xy
- table_for_database

3.4 Merge request-y

Pred spojením vetvy s dokončenou funkcionalitou s hlavnou vetvou je ešte potrebné natiahnuť si do vetvy poslednú verziu hlavnej vetvy a vyriešiť konflikty. Jeden spôsob ako toto spraviť je použiť príkazy:

- "git checkout nazov_vetvy"
- "git pull origin master"

Po vyriešení konfliktov je potrebné vytvoriť merge request, ktorý musí prejsť prehliadkou. Človek vykonávajúci prehliadku (reviewer) doplní svoje pripomienky pomocou komentárov do merge requestu. V prípade, ak po dokončení prehliadky sú v kóde nejaké pripomienky, tak sa pošlú zodpovednému členovi tímu na prepracovanie. Ak po prehliadke nie sú žiadne pripomienky, tak sa merge request akceptuje.

Kapitola 4

VÝVOJ A INTEGRÁCIA

Zodpovedná osoba: Bc. Miroslav Valčičák

Posledná úprava: 19.11.2017, verzia 1

Metodika je určená všetkým, ktorí sa akýmkoľvek spôsobom podieľajú na tvorbe produktu a jeho nasadení do produkčného prostredia. Pritom nadväzuje na:

- Kapitola 3. Práce s verziami
- Kapitola 5. Dokumentácia
- Kapitola 6. Testovanie

Snahou nášho tímu je, aby proces vývoja a integrácie bol čo najjednoduchší, najefektívnejší, nadmieru nezaťažoval tím a zároveň zaručoval zachovanie vysokej kvality vytváraného produktu. Taktiež sa proces pokúša presunúť čo najviac rutínnej práce na nástroje a systémy podporujúce ich automatizované vykonávanie.

4.1 Role

Databázový návrhár - človek, ktorý navrhuje dátový model a definuje obmedzenia.

Návrhár grafického používateľského rozhrania - jeho úlohou je navrhnuť rozhranie pre používateľa z vizuálneho hľadiska. Zaoberá sa farbami, ale aj rozmiestnením jednotlivých prvkov.

API návrhár - je zodpovedný za dobre definované rozhranie pre komunikáciu medzi klientom a serverom.

Vývojár - implementuje návrhy a odstraňuje implementačné chyby.

Recenzent zdrojového kódu - vykonáva statickú analýzu kódu, za účelom odhalenia chýb, odstraňovania nekonzistencie a nedodržiavania dohodnutých zásad pri programovaní.

4.2 Opis procesov

Návrh procesu vývoja a integrácie bol výrazne ovplyvnený praktikami kontinuálnej integrácie (ang. *continuous integration*, Obrázok 4.1)¹ a kontinuálneho nasadzovania (ang. *continuous deployment*)². Ich aplikovanie v procese pomáhajú dosiahnuť nasledovné nástroje a služby:

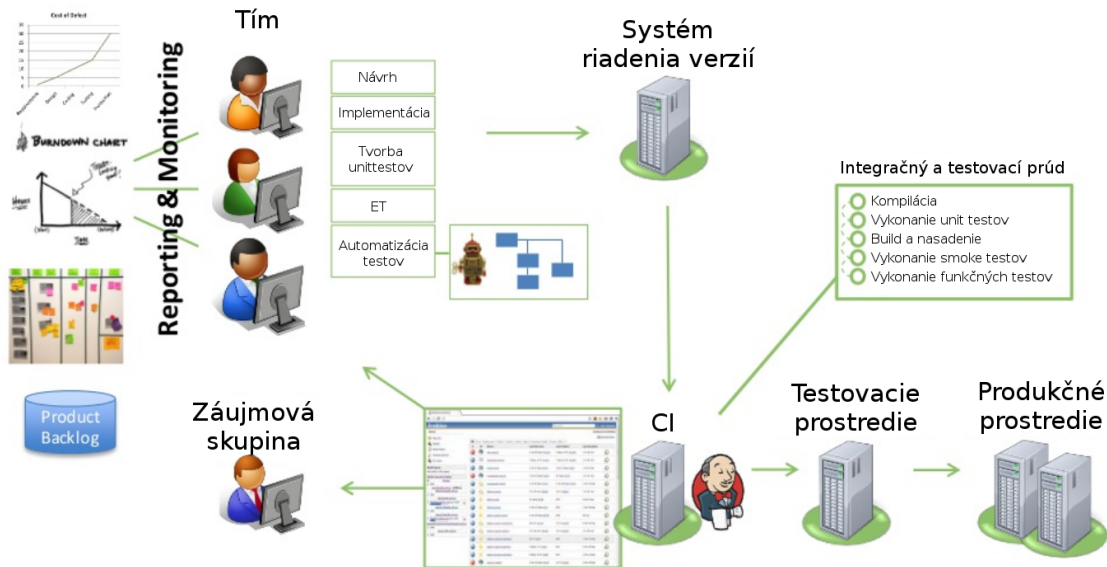
- Git/GitLab (manažment verzií zdrojového kódu)
- Docker (integrácia)
- Jenkins (testovanie, nasadzovanie)

Pri vývoji každej samostatnej funkcionality je potrebné dodržiavať nasledujúcu postupnosť krokov:

¹<https://codeship.com/continuous-integration-essentials>

²<https://codeship.com/continuous-integration-essentials>

Kontinuálne nasadzovanie



Obr. 4.1: Proces kontinuálneho nasadzovania.

1. vyberie sa prípad použitia
2. databázový návrhár vytvorí návrh dátového modelu a jeho vizualizácia UML diagramom na základe definovaných požiadaviek (ak to charakter úlohy umožňuje)
3. návrhár grafického používateľského rozhrania vytvorí wireframe-y a mockup-y pre používateľa systému (ak to charakter úlohy umožňuje)
4. API návrhár definuje rozhrania komunikácie medzi klientskou časťou systému a serverovou časťou - potrebná konzultácia s vývojárom klientskej aj serverovej časti systému
5. vývojári vytvoria osobitné vetvy v systéme na manažment verzií zdrojového kódu
6. vývojári implementujú časti funkcionality
7. vývojári vytvoria testy pre príslušné časti zdrojového kódu
8. vývojári vytvoria dokumentáciu k príslušnej časti zdrojového kódu
9. vývojári okomentujú a odošlú zmeny do vzdialeného repozitára vetvy
10. v prípade, že funkcia nie je úplne implementovaná, pokračovať bodom 6
11. vývojári si k sebe prevezmú zmeny z hlavnej vývojovej vetvy
12. vývojári vyriešia prípadné konflikty
13. vývojári pridajú komentáre a odošlú riešenia konfliktov do vzdialeného repozitára vetvy
14. vývojári vytvoria požiadavky na zlúčenie s hlavnou vývojovou vetvou (ang. *merge request*) v prostredí GitLab-u (iba ak ešte neexistuje)
15. vývojári požiadajú recenzentov zdrojového kódu (sú pridelení pre každú úlohu) o vykonanie code review

16. recenzenti zdrojového kódu si prezrú novovytvorený kód a v prípade potreby pridajú k nemu pripomienky
17. vývojári zapracujú pripomienky vo vlastnej vetve a pokračujú bodom 11 (iba v prípade, že nejaké pripomienky existujú)
18. vývojári zlúčia vývojové vetvy

Po každej zmene v hlavnej vývojovej vetve Jenkins spustí automatizované testy a v prípade ich úspešného ukončenia nasadí novú verziu produktu do produkčného prostredia. Vývojári sú povinný sledovať stav testovania a v prípade ich neúspešného dokončenia musia chyby odstrániť vo vlastných vývojových vetvách. Bod číslo 2 a 3 sú od seba navzájom nezávislé a môžu sa vykonávať paralelne.

4.3 Záver

Aby mala metodika procesu vývoja a integrácie reálny pozitívny prínos, je nevyhnutné, aby ju každý člen vývojového tímu rešpektoval a dodržiaval. Jednotlivé kroky vývoja môžu presnejšie upravovať ďalšie metodiky, s ktorými sa je každý člen tímu povinný oboznámiť ešte pred začatím práce na vývoji produktu.

Kapitola 5

DOKUMENTÁCIA

Zodpovedná osoba: Bc. Michal Dolnák, Bc. Michal Hucko

Posledná úprava: 19.11.2017, verzia 2

Cieľom dokumentu je objasniť postup pri dokumentovaní. Metodika je záväzná a platí pre všetkých členov tímu 1, ktorí sa podieľajú na tvorbe kódu. Všetky postupy opísané v tomto dokumente sú platné. Táto metodika nadväzuje na metodiku:

- Kapitola 3. Práca s verziami

5.1 Role

Programátor - osoba ktorá je zodpovedná za napísaný kód (či na frontende, alebo backende).

Reviewer - osoba zodpovedná za code review.

5.2 Predpoklady

5.2.1 Spustenie frontendu

Kľúčové závislosti Angularu sú (názov -> odporúčaná verzia):

- npm -> 5.3.0
- nodeJs¹ -> 8.6.0
- ng -> 1.4.4

Odporúča sa pracovať na operačných systémoch založených na UNIX technológii, ako napr. Linux či OSX. Pre používateľov Windows doporučujeme použiť VirtualBox na vytvorenie UNIX virtuálneho stroja.

- npm a nodejs stiahnete pomocou príkazu "\$ apt-get install npm"(Linux) alebo príkazom "\$ brew install npm"(OSX).
- ng nainštalujete pomocou príkazu (Linux, OSX): "\$ npm install -g @angular/cli"

Lokálne spustenie

Postup:

- Lokálne spustíme aplikáciu pomocou príkazu: "\$ ng serve"
- V environment.local.ts si je potrebné nastaviť premennú prostredia apiBase, aby smerovala na backend.

¹<https://nodejs.org/en/>

- Backend musí mať povolené CORS pre prístup z portu, kde je lokálne spustený frontend. V opačnom prípade odmietne prístup a nie je možné lokálne vyvíjať².

Lokálne spustenie (docker)

V súbore dockerfile v priečinku frontendu je nutné spúšťať aplikáciu pomocou príkazu: "CMD ["npm", "run", "devel"]"

5.2.2 Spustenie backendu

Na to, aby bolo možné písať dokumentáciu je potrebné aby mal programátor nasledujúce knižnice:

- django ("\$ pip install django")
- django-rest-framework ("\$ pip install djangorestframework")
- django swagger ("\$ pip install django-rest-swagger")

Swagger je nástroj na dokumentovanie. V našom projekte ho používame na dokumentovanie REST rozhrania. Aby správne bežal, musí byť okrem inštalácie pridaný k nainštalovaným aplikáciám³:

```
1 INSTALLED_APPS = [
2     ...
3     'rest_framework_swagger',
4     ...
5 ]
```

Ukážka zdrojového kódu 5.1: Vyhodenie dokumentácie Python

A taktiež musí byť pridaný medzi URL v projekte:

```
1 from django.conf.urls import url
2 from rest_framework_swagger.views import get_swagger_view
3
4 schema_view = get_swagger_view(title='Some cool name')
5
6 urlpatterns = [
7     url(r'^$', schema_view)
8 ]
```

Ukážka zdrojového kódu 5.2: Vyhodenie dokumentácie Python

Lokálne spustenie

Pre lokálne spustenie projektu prejdite do adresára /backend/group/ v koreňovom adresári projektu a spustite príkaz: "\$ python manage.py runserver"

Následne beží na localhost:8000/.

Lokálne spustenie (docker)

Pre lokálne spustenie pomocou dockeru je potrebné z koreňového adresára projektu spustiť príkaz: "\$ docker-compose up"

Následne beží na localhost:8000/.

²https://en.wikipedia.org/wiki/Cross-origin_resource_sharing

³<https://django-rest-swagger.readthedocs.io/en/latest/>

5.3 Opis procesu

5.3.1 Frontend

Pravidlá:

- Dokumentáciu sa píše k triedam, k funkciám a k zložitejším premenným.
- Pri dokumentácii funkcií, ktoré majú parametre a návratové hodnoty používajte tagy @param a @return s opisom a použitými dátovými typmi.
- Dokumentáciu píšete po anglicky a vo vetách.
- Ak majú premenné deklarované typy, tak ich píšeme v zátvorkách.

```
1 /**
2 * Function creates http request to the server in a proper format, which is decided by it's
   type.
3 * @param options object with type(string), query(boolean) and body(object) fields
4 * @param callback to call after response is received
5 *
6 * @return callback(err(object)), data(object))
7 */
```

Ukážka zdrojového kódu 5.3: Príklad dokumentácie frontend

5.3.2 Backend

Pre tento účel sa používa nástroj swagger. Aby swagger vedel, čo má dať do dokumentácie, špecifikú sa detaily už počas vytvárania modelu⁴. Je potrebné uviesť, ktoré polia sú povinné, aký majú typ, správne napísať ich názov, prípadne uviesť základné hodnoty. Príklad:

```
1 class Vote(models.Model):
2     created = models.DateTimeField(auto_now_add=True)
3     username = models.CharField(max_length=100, default="User")
```

Ukážka zdrojového kódu 5.4: Príklad dokumentácie backend

Ďalej je potrebné okomentovať každú metódu vo view, ktorá bude použitá v URL. Swagger používa tieto komentáre ako pole Implementation notes. Je nutné, aby bolo v komentári spomenuté, na čo bude metóda slúžiť. Príklad:

```
1 class EventCreate(generics.GenericAPIView):
2     """
3     Create new Event with given Options.
4     """
5     def get_serializer_class(self):
6         return NewEventSerializer
7
8     def post(self, request, format=None):
9         """
10        Create new Event with given Options.
11        """
```

Ukážka zdrojového kódu 5.5: Príklad dokumentácie backend

Komentáre

V projekte je nutné dodržať štruktúru komentárov. Komentáre dodržiať nasledujúce pravidlá:

- Komentár je písaný po anglicky.
- Komentár začína veľkým písmenom.

⁴<http://www.django-rest-framework.org/topics/documenting-your-api/>

- Komentár končí bodkou.
- Komentár je písaný rozkazovacím spôsobom.

Vhodné príklady:

```
1 "Create new action."
2 "Update action depending on the date."
```

Ukážka zdrojového kódu 5.6: Príklad komentárov backend

Nevhodné príklady:

```
1 "It will create a record"
2 "Some method"
```

Ukážka zdrojového kódu 5.7: Príklad komentárov backend

5.4 Záver

Dokumentácia je jedna z častí definition of done. Programátor je teda zodpovedný za jej obsah. Reviewer je povinný okrem funkcionality kontrolovať aj dokumentáciu, či je v súlade s týmto dokumentom. Ak sa počas vývoja produktu vyskytnú nedostatky tohto dokumentu, je možné ho rozšíriť.

Kapitola 6

TESTOVANIE

Zodpovedná osoba: Bc. Michal Dolnák, Bc. Michal Hucko
Posledná úprava: 19.11.2017, verzia 2

Cieľom dokumentu je objasniť postup pri testovaní. Metodika je záväzná a platí pre všetkých členov tímu 1, ktorí sa podieľajú na tvorbe kódu. Všetky postupy opísané v tomto dokumente sú platné. Táto metodika priamo nadväzuje na metodiku:

- Kapitola 5. Dokumentácia

6.1 Role

Programátor - osoba, ktorá je zodpovedná za napísaný kód (či na frontende, alebo backende).

6.2 Definition of done z pohľadu testovania

Každý prípad použitia musí byť napísaný test, ktorý zachytáva korektné správanie aplikácie pri vykonávaní prípadu použitia. Test nepíšeme len v prípade, že ho z objektívnych príčin nie je možné vykonať alebo by jeho výsledok nebol deterministický. Aplikácia sa odovzdáva na posúdenie (ang. *review*), ak sú funkčné všetky existujúce automatické testy.

6.3 Predpoklady

6.3.1 Frontend

- Predpoklad je fungujúca frontendová aplikácia. Návod na spustenie aplikácie nájdete v Kapitole 5, v časti dokumentovanie frontendu, v sekcii Návod na lokálne spustenie projektu.
- Nainštalovanú najnovšiu verziu Google Chrome prehliadača (v dobe písania kódu aspoň v61) pre Selenium.
- Testy sa spúšťajú pomocou príkazu: "\$ npm run e2e"
- Testy sa spúšťajú na porte 49152 a využíva sa lokálna konfigurácia frontendu.
- Pre korektné fungovanie testov je nutné mať povolené CORS na strane backendu pre port 49152.

6.3.2 Načítanie testovacích údajov cez docker

Na načítanie testovacích údajov do databázy, zálohu aktuálnej databázy a jej obnovenie slúžia 3 príkazy:

- „docker-compose run backend python manage.py setupdbfrontend“ - resetuje databázu a vloží testovacie údaje.
- „docker-compose run backend python manage.py setupdbfrontendwithbackup“ - zálohuje aktuálne údaje v databáze, resetuje ju a vloží testovacie údaje.
- „docker-compose run backend python manage.py restorebackup“ - resetuje databázu a načíta zálohované údaje.

Tieto príkazy sa spúšťajú z hlavného adresára projektu, tj. adresár v ktorom je súbor docker-compose.yml.

6.3.3 Backend

Zdrojový kód na backende je napísaný nad frameworkom django-rest. Verzia frameworku django je aspoň 1.11.5 a frameworku django-rest je 3.6.4. Pri implementácii testovania používame štandardné testovanie nad frameworkom Django, ktoré sa dá spustiť z koreňového adresára backendu cez príkaz: "\$ python manage.py test"

6.4 Opis procesu

6.4.1 Frontend

Štruktúra testov

Príklad testu:

```
1 describe('Home page is correct.', () => {
2   let page: HomePage;
3
4   beforeEach(() => {
5     page = new HomePage();
6   });
7
8   it('should display project name in header', () => {
9     page.navigateTo();
10    expect(page.getProjectName()).toEqual('Group de Cider');
11  });
12 });
```

Ukážka zdrojového kódu 6.1: Príklad testu frontend

Testy sa skladajú z 3 častí. Prvou je časť describe, kde je nutné uviesť aká časť systému sa testuje. Druhou sú deklarované komponenty alebo stránky, ktoré sa budú testovať. Tretou sú jednotlivé časti testu zoskupené v podmienkach¹.

Formálne zásady:

- Popisy testu sa píše po anglicky.
- Všetky súbory s testami musia byť nazvané *.e2e.ts a musia byť umiestnené v priečinku e2e.
- Na testovanie sa používa framework protractor².

¹<http://www.protractortest.org/#/style-guide>

²<http://www.protractortest.org>

Štruktúra stránok

```
1
2 export class HomePage {
3   navigateTo() {
4     return browser.get('/');
5   }
6
7   getProjectName() {
8     return element(by.id('project-name')).getText();
9   }
10
11   getAboutUsButton() {
12     return element(by.id('about-us'));
13   }
14
15   getLoginButton() {
16     return element(by.id('login-button'));
17   }
18 }
```

Ukážka zdrojového kódu 6.2: Príklad testu frontend

Stránky opisujú komponenty aplikácie a vyberajú z nich jednotlivé elementy, ktoré sa následne používajú v testoch. Elementy sa vyberajú na základe atribútov unikátnych identifikátorov (id) a ak to nie je možné, tak na základe atribútov tried.

6.4.2 Backend

Štruktúra testov

Testy organizujeme do logických celkov, ktorými sú triedy. Každá trieda môže dediť z jednej z možností:

- Štandardnej triedy djanga `TestCase` v prípade, že sa jedná o funkcionality mimo frameworku `django-rest`: `"from django.test import TestCase"`
- Triedy `APITestCase` v prípade testovania funkcionality `django-rest` frameworku. Viac v kapitole 6.4.2. `"from rest_framework.test import APITestCase"`

Testovacie scenáre sú ďalej organizované do metód.

Štábná kultúra

Konvencie štruktúry písania názvov V prípade „test case-ov“ použijeme camel-case formát názvu (prvé písmeno veľké a každé nové slovo s veľkým, všetky slová spolu). V prípade špecifických názvov metód využijeme snake-case (každé slovo s malým oddelené podtržníkom). Príklad:

```
1 class MyNewTestCase(APITestCase):
2     def new_test_1_test(self):
```

Ukážka zdrojového kódu 6.3: Príklad testu backend

Konvencie štruktúry obsahu názvov Celý „test case“ je potrebné robiť v kontexte view, na ktorý sa viaže. To znamená, že ak testujem funkcionality view, kde vytvárame Auto, názov bude vytvorený presne podľa toho. Názvy test-casov končia reťazcom „TestCase“. Konkrétne metódy v test case (testy) sú potom pomenúvané podľa špecifickej udalosti, ktorá je testovaná. Názvy testov končia reťazcom „_test“. Nasledujúci príklad ukazuje ako pomenovať test case a testy na vytváranie triedy auta:

```
1 class CreateCarTestCase(APITestCase):
2     def test_create_car_test(self):
3         ...
4     def test_create_car_without_wheels_test(self):
```

Ukážka zdrojového kódu 6.4: Príklad testu backend

```

1 class New_testcase(APITestCase):
2     def testcreatecartest(self):
3         ...
4     def testCreateCarWithoutWheelsTest(self):

```

Ukážka zdrojového kódu 6.5: Nevhodný príklad testu backend

Django-rest APITestCase

Výhodou použitia tejto triedy je fakt, že nám poskytuje parameter “client”, cez ktorý vieme jednoducho posielat’ dopyty na REST API. Príklad³:

```

1 class CreateCarTestCase(APITestCase):
2     def test_create_car_test(self):
3         response = self.client.post(url, data, format='json')
4         ...

```

Ukážka zdrojového kódu 6.6: Príklad testu backend

6.5 Záver

Testovanie je jedna z častí definition of done. Programátor je teda zodpovedný za jej obsah. Posudzovateľ je povinný okrem funkcionality kontrolovať aj testy, či sú v súlade s týmto dokumentom. Okrem toho spúšťa testy. Ak počas sa počas vývoja produktu vyskytnú nedostatky tohto dokumentu, je možné ho rozšíriť.

³<http://www.django-rest-framework.org/api-guide/testing/>

Fakulta informatiky a informačných technológií
Slovenská technická univerzita

Group de'Cider

Dokumentácia k tímovému projektu časť 1
(Inžinierske dielo)

Tím: číslo 1, sElectors

Pedagogický vedúci: Ing. Ondrej Kaššák

Členovia tímu: Bc. Lukáš Babula, Bc. Pavel Balashov, Bc. Michal Dolnák, Bc. Michal Hucko, Bc. Elena Štefancová, Bc. Miroslav Valčíčák, Bc. Andrej Vítek

Akademický rok: 2017/2018

Úvod	2
1 Globálne ciele	3
1.1 Zimný semester	3
2 Celkový pohľad na systém	4
2.1 Backend systému	5
2.2 Frontend systému	5
3 Moduly systému	6
3.1 Modul hlasovacích stratégií	6
3.1.1 Analýza	6
3.1.2 Návrh	6
3.2 Modul hlasovania	6
3.2.1 Analýza	7
3.2.2 Návrh	7
3.2.3 Implementácia	7
3.2.4 Testovanie	7
3.3 Modul autentifikácie	7
3.3.1 Analýza	7
3.3.2 Návrh	8
3.3.3 Implementácia	8
3.3.4 Testovanie	8
Záver	9
Prílohy	10
Príloha A: Dokumentácia volaní z nástroja swagger	10
Volania patriace do modulu hlasovania	10
Volania patriace do modulu autentifikácie	13
Príloha B: Návrhy grafických rozhraní	14
Prvotný návrh grafických rozhraní	14
Druhá iterácia návrhu grafických rozhraní	15
Tretia iterácia návrhu grafických rozhraní	17

Úvod

Dokument obsahuje informácie o projekte Group de'Cider po prvých 3 šprintoch v rámci predmetu Tímový projekt. Cieľom dokumentu je opísať technickú časť nášho projektu.

Dokument sa skladá z globálnych cieľov nášho projektu, cieľov v rámci zimného semestra a celkového pohľadu na vytváraný systém. Taktiež obsahuje informácie o moduloch systému a ich návrhu, implementácie a spôsobe testovania. V prípade, keď modul vyžadoval aj detailnejšiu analýzu nejakého problému, tak obsahuje aj túto časť.

1 Globálne ciele

Hlasovacích aplikácií existuje veľa. Väčšinou však zvládnu len jednoduché jednorazové úlohy. Rovnako spôsob hlasovania v nich je obmedzený na spočítanie hlasov, ktoré sa nehodí vždy.

V rámci tímového projektu sa snažíme vytvoriť univerzálne riešenie umožňujúce efektívne rozhodovanie v skupinách. Group de'Cider je webová aplikácia poskytujúca:

- tvorbu hlasovaní pre skupiny používateľov
- intuitívne používateľské rozhranie
- výber rozličných hlasovacích stratégií (average without misery, most pleasure, cumulative voting, dictatorship)
- opakované hlasovania (napríklad pre každodenný výber miesta na obed)
- komplexné hlasovania (súčasný výber napríklad času A miesta alebo miesta A aktivity)
- možnosť skrytého hlasovania
- chat pre rýchle vyjasnenie situácií

Vytváraná aplikácia je určená pre aktívnych ľudí komunikujúcich s priateľmi, kolegami či spolužiakmi, ktorí sa chcú vedieť dohodnúť dobre a rýchlo. Používať ju zvládne každý, kto vie ovládať počítač. Vytvoriť hlasovanie a pozvať doň priateľov je úplne jednoduché.

Biznis potenciál nášho riešenia bude môcť spočívať v platenej funkcionalite, napríklad v obmedzení maximálnej veľkosti skupiny pre free verziu, zamknutí zaujímavých hlasovacích stratégií, či obmedzení maximálneho počtu opakovaní hlasovania.

1.1 Zimný semester

Počas zimného semestra sa náš tím sústreďí na navrhnutie a implementáciu základnej funkcionality vytváraného systému, jeho nasadenie a použiteľnosť.

V rámci základnej funkcionality je naším cieľom poskytnúť možnosť vytvorenia hlasovania a opakovaného hlasovania, umožniť používateľovi zahlasovať a taktiež vytvoriť jednoduchú administráciu, pomocou ktorej bude možné upraviť vytvorené hlasovanie. Chceme taktiež umožniť používateľom sa prihlásiť a mať prehľad o všetkých hlasovaniach, ktoré buď vytvorili, alebo do nich patria.

Nasadením je myslené vytvorenie kontinuálnej integrácie pomocou nástroja Jenkins. Taktiež sem patrí vytvorenie Docker obrazu pre náš projekt a samotné nasadenie na server so zabezpečením možného prístupu cez internet.

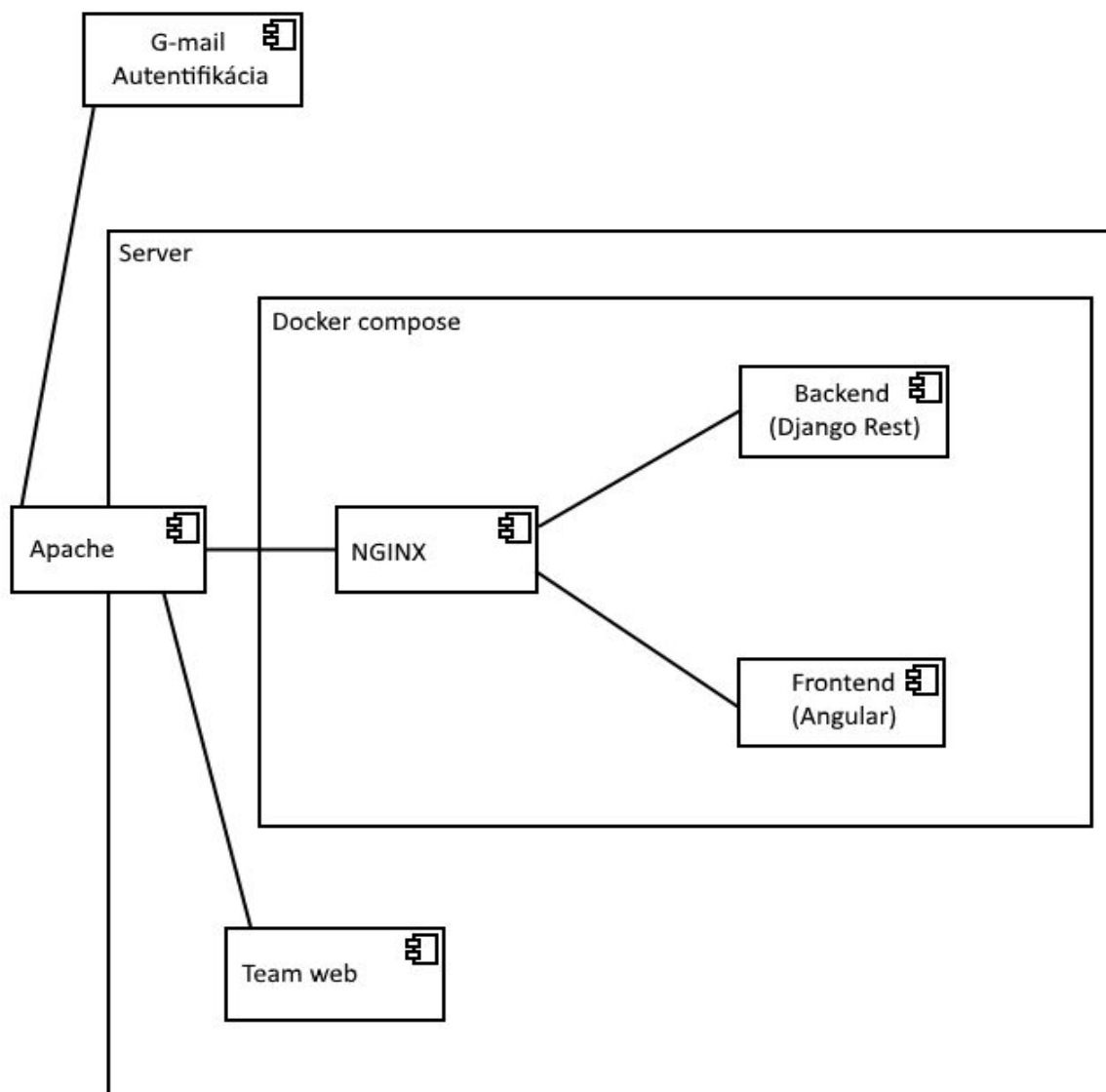
Na zabezpečenie použiteľnosti sa zameriavame na navrhnutie vhodného rozhrania pomocou mockupov.

2 Celkový pohľad na systém

Celá aplikácia pozostáva z docker riešenia, ktoré združuje tri kontainery. Prvým z nich je backend kontajner, ktorý v sebe uchováva django REST API a postgresql databázu. Druhým je frontend s angulárom. Aby tieto 2 kontainery vedeli medzi sebou komunikovať používame tretí s NGINXom, ktorý slúži ako proxy HTTP server. Využitie technológie dockeru značne podporuje integráciu a znovu-nasadenie aplikácie .

Compose spolu s hlavnou web stránkou sme nasadili na pridelený server. Ich komunikáciu s okolím zabezpečujeme pomocou apache HTTP serveru, ktorý funguje ako reverse proxy pre NGINX kontajner. Celkový pohľad na architektúru nášho systému v rámci servera je vidieť na obrázku 1.

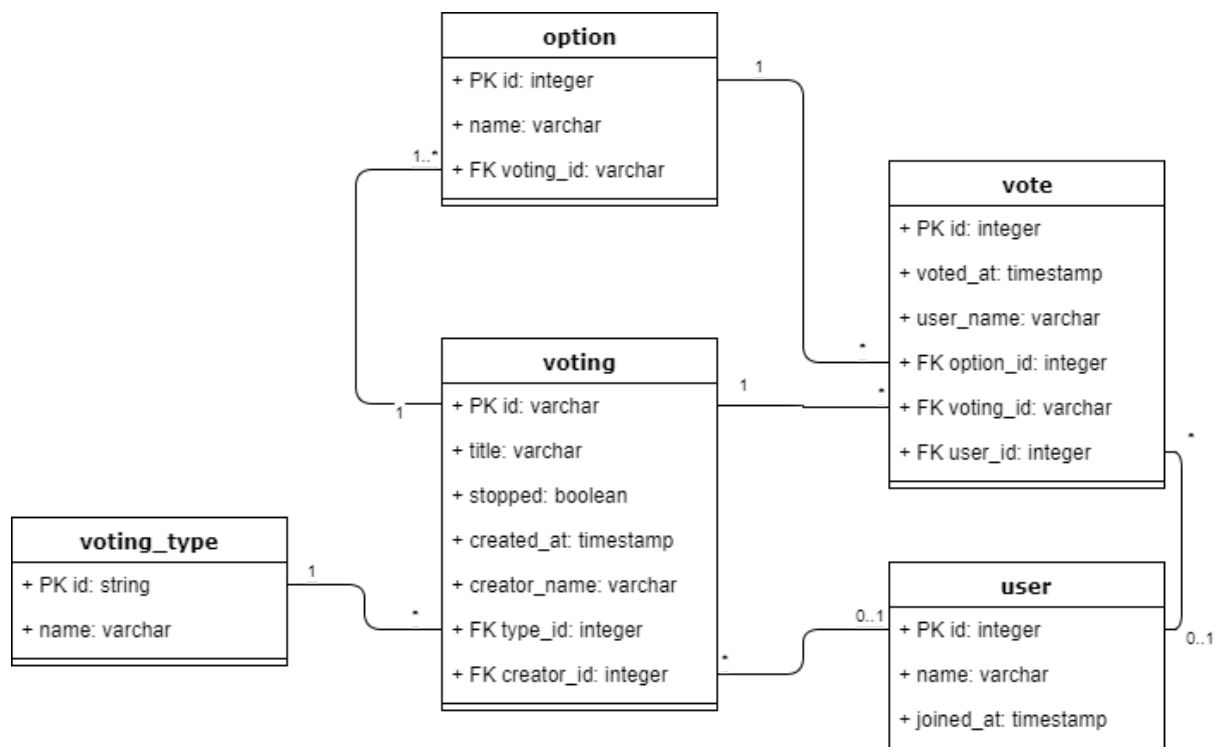
Kód projektu ako aj jednotlivé verzie vyvíjame pomocou systému na kontrolu verzií Git uchováваме vo vytvorenom úložisku od spoločnosti GitLab.



Obr. 1: Ukážka architektúry systému v rámci servera. Apache je použitý na zabezpečenie prístupu do vytvoreného Docker kontajnera s projektom a na tímovú stránku.

2.1 Backend systému

Do časti backend patrí databáza PostgreSQL a rámec (framework) Django REST, ktorý slúži na vývoj webových aplikácií. Testovanie, správu databázy a vytváranie modelu ma tiež na starosti rámec Django REST a jeho knižnice. Dokumentácia je v rámci tejto časti zabezpečená pomocou nástroja Swagger. Návrh schémy databázy je vidieť na obrázku 2. Hlavnou úlohou tejto časti je komunikácia pomocou API volaní s frontend-om a poskytovanie aplikačnej logiky.



Obr. 2: Návrh schémy pre databázu.

Aby sme umožnili aj vytvorenie anonymných hlasovaní, tak v prípade vytvorenia nového hlasovania (voting) alebo zahlasovania za možnosť (vote), nie je nutné prepojenie s tabuľkou používateľa (user) a v tom prípade sa zaznamená len meno používateľa.

2.2 Frontend systému

Frontend je v rámci nášho projektu oddelený od backend-u a je vytvorený pomocou aplikačného rámca Angular v programovacom jazyku typescript. Na automatické testovanie celej aplikácie sa využíva aplikačný rámec Protractor, ktoré je postavený na nástroji Selenium. Automatické testy sa spúšťajú prostredníctvom nástroja Headless Chromium. Dokumentácia zdrojového kódu sa riadi odporúčaniami projektu TypeDoc.

Do aplikácie sa postupne pridávajú grafické návrhy používateľských rozhraní prostredníctvom jazyka SASS, ktorý sa kompiluje na štandardné CSS.

3 Moduly systému

3.1 Modul hlasovacích stratégií

3.1.1 Analýza

Jedným z cieľov nášho systému je možnosť použiť rôzne zaujímavé hlasovacie stratégie pri vytváraní hlasovania. Kvôli tomu sme v tejto časti robili prieskum v oblasti možností hlasovania a systémov volieb. Mnohé tieto systémy a hlasovania sú v súčasnosti používané v politickej scéne, kde sa pomocou nich vyberajú politickí hodnostári. Sú však aj hlasovacie systémy a stratégie, ktoré majú svoj pôvod pri manažovaní kolaboratívneho výberu a rozhodovania sa. Medzi typické príklady takýchto stratégií patrí napríklad hlasovacia stratégia *Dot voting*.

Počas hľadania a zhromažďovania jednotlivých stratégií bol kladený dôraz aj na vedecké zdroje a publikácie z oblasti modelovania skupín používateľov, rozhodovania sa jednotlivcov v rámci takýchto skupín a možností rozhodovania sa skupiny ako celku. Na základe toho je výsledkom tejto analýzy výber trinástich rôznorodých, pre náš problém použiteľných, hlasovacích stratégií a ich modifikácií, z ktorých sa po spoločnej konzultácii v kruhu členov nášho tímu a produktového vlastníka niektoré vyberú a následne použijú v našom systéme.

Všetky hlasovacie stratégie uvažujú asynchrónny model rozhodovania sa, kedy používateľ hlasuje nezávisle na čase hlasovania ostatných používateľov. Do úvahy však pripadá aj možnosť, kedy by hlasovanie prebiehalo synchronne. V takom prípade by bolo možné dosiahnuť výsledné rozhodnutie hlasovaním v reálnom čase, na čo by bola napríklad vhodná stratégia viackolových volieb. Tento model hlasovania sme však bližšie nerozvíjali a v prípade potreby a záujmu sa mu budeme venovať v neskorších fázach projektu.

3.1.2 Návrh

Návrh tohto modulu je v súčasnosti vo veľmi ranom štádiu. Uvažujeme robustné riešenie na backend-e, ktoré by generalizovalo všetky hlasovacie stratégie pod spoločné rozhranie, kde by pre každú stratégiu existovali iné ohraničenia na základe možností voľby.

Keďže tento modul sa zatiaľ nachádza vo fáze návrhu, tak implementácia ani testovanie nie sú realizované.

3.2 Modul hlasovania

Tento modul sa zaoberá vytvorením hlasovania na základe vybraných možností, hlasovacej stratégie a titulu. Tiež sem patrí aj schopnosť používateľa zahlasovať v už vytvorenom hlasovaní a správa hlasovania cez administráciu, v ktorej je možné upravovať jeho vlastnosti. Po zahlasovaní má používateľ umožnené zmeniť možnosti, za ktoré hlasoval.

3.2.1 Analýza

Uskutočnili sme prieskum iných webových stránok, ktoré poskytujú hlasovania a zhodli sme sa, že by mal byť proces vytvorenia hlasovania z pohľadu používateľa čo možno najjednoduchší. Vytvorenie hlasovania a pridanie hlasu do hlasovania by malo byť umožnené prihláseným aj neprihláseným používateľom. Po vytvorení hlasovania dostane používateľ dve unikátne URL adresy. Pomocou prvej z nich môže administrovať hlasovanie. Druhá slúži na pridávanie hlasov do hlasovania inými používateľmi.

3.2.2 Návrh

Frontend a backend časti sú v našom systéme oddelené a vyžadujú definovať žiadosti (request) a odpovede (response) v API volaniach, ktoré sa používajú na komunikáciu medzi nimi. Pre účel komunikácie medzi backend-om a frontend-om sme zdefinovali API, ktoré umožňuje prenos dát vo formáte JSON.

Vytvoril sa návrh dátového modelu databázy, ktorý je súčasťou backend-u. Ďalej sa navrhli grafické rozhrania vytvorenia hlasovania a pridania nového hlasu do hlasovania, ktoré je možné vidieť v prílohe B.

3.2.3 Implementácia

Na strane backend-u boli implementované volania, ktoré poskytovali funkcionality vytvorenia hlasovania, pridania nového hlasu do hlasovania a získania aktuálneho stavu hlasovania. Dokumentáciu jednotlivých volaní z nástroja Swagger možno vidieť v prílohe A.

Na základe grafického návrhu boli implementované obrazovky vytvorenia hlasovania a pridania nového hlasu do hlasovania. Vzhľadom na to, že išlo o implementáciu prvých rozhraní medzi časťami systému, tak sa vyriešila integrácia všetkých používaných technológií.

3.2.4 Testovanie

Pre tento modul sú vytvorené automatizované testy. Na strane backend-u testy overujú konzistentnosť ukladaných dát. Na strane frontend-u sa testuje úspešné vykonanie všetkých používateľských scenárov automatickou simuláciou prostredníctvom webového prehliadača.

3.3 Modul autentifikácie

Primárnou úlohou modulu je overovať oprávnenosť používateľov vykonávať určité akcie. Slúži na prvotnú registráciu používateľa do nášho systému a overovanie jeho totožnosti pri prihlasovaní sa. Taktiež poskytuje ostatným modulom informáciu o identite aktuálne (ne)prihláseného používateľa.

3.3.1 Analýza

Je v našom záujme, aby sa používateľ cítil pri používaní našej webovej aplikácie čo najpríjemnejšie a nebol zbytočne zaťažovaný nepotrebnými úkonmi. Preto sme sa rozhodli proces registrácie - vkladanie osobných a prihlasovacích údajov a overovanie hesla -

nahradiť autentifikačnou službou tretej strany. Takéto riešenie by malo výrazne skrátiť dobu prvého prihlásenia sa a odbremeniť používateľa od pamätania si ďalších prihlasovacích údajov.

Na základe prieskumu popularity sociálnych sietí sme ako sprostredkovateľa autentifikácie zvolili *Google+ API* autentifikačné služby. Vybraný sprostredkovateľ je v súčasnosti jediný, s ktorým naša aplikácia spolupracuje. Je teda nevyhnutné, aby používateľ, ktorý sa rozhodne registrovať do našej služby, vlastnil nejaký účet (stačí e-mailový) od spoločnosti Google.

3.3.2 Návrh

Registrácia a prihlásenie zo strany používateľa pozostáva len z kliknutia na tlačidlo prihlásenia sa a potvrdenia povolenia poskytnúť osobné údaje používateľa našej službe zo stránky spoločnosti Google. Naša aplikácia následne prevezme potrebné informácie o používateľovi a v prípade, že sa jedná o nového používateľa, vytvorí nové záznamy v databáze.

3.3.3 Implementácia

Registrácia a autentifikácia prebieha v backend časti systému, ktorá komunikuje so službami Google cez poskytnuté rozhranie. Aby bolo možné vyžívať autentifikačné služby Google, bolo potrebné vytvoriť *credentials*. Takisto bol získaný SSL certifikát od certifikačnej authority *Lets's Encrypt* pre náš hostname, pretože ďalšou podmienkou vyžívania Google služieb bolo zabezpečené šifrované spojenie, ktoré implementáciu mierne skomplikovalo a oddialilo jej dokončenie.

Proces komunikácie s Google službami, vytváranie nových používateľov na základe získaných informácií a obnovovanie platnosti autorizačných tokenov prebral na seba modul od externých poskytovateľov *python-social-auth*. Informácie o prihlásení a oprávnenosti používateľov je typu *"session-based"* a ich kontrola prebieha výlučne na backend-e, s čím boli spojené prvotné komunikačné problémy s frontend-om, ktoré zapríčinili značné oneskorenie dokončenia a nasadenia modulu.

3.3.4 Testovanie

Testovanie registrácie a prihlasovania je vykonávané manuálne kvôli náročnej tvorbe testovacieho skriptu, ktorého výsledok je závislý od tretej strany. Avšak, testovanie kontroly oprávnenosti vykonávať akcie vyžadujúce prihlásenie, ako aj prístupnosť verejných funkcií pre anonymných používateľov, sa už vykonáva automaticky pomocou testovacieho skriptu.

Záver

Počas prvých troch šprintov sme sa sústredili na vytvorenie základného hlasovania, návrh používateľských rozhraní a na prípravu prostredia. Aby sme zjednodušili ďalší vývoj na projekte, tak sme vytvorili Docker obraz pre jednotlivé časti projektu a zabezpečili automatické nasadzovanie pomocou aplikácie Jenkins. Návrh používateľských rozhraní prešiel cez viacero iterácií, ako je možné vidieť v prílohe B.

Naším cieľom do budúcnosti je pridať opakované hlasovania a tiež pripraviť univerzálne robustné riešenie na backend-e, ktoré by umožnilo pridať ďalšie hlasovacie stratégie a navrhnuť pre ne vhodné používateľské rozhrania. Ďalšou úlohou je pridanie funkcionality pre prihláseného používateľa, ako je poskytnutie prehľadu o hlasovaniach, do ktorých je zapojený. Taktiež sem patrí vytvorenie miestností, ktoré budú slúžiť na zjednotenie používateľov do skupín.

Prílohy

Príloha A: Dokumentácia volaní z nástroja swagger

Volania patriace do modulu hlasovania

POST

/event_create/

Create new Event with given Options.

Implementation Notes

Create new Event with given Options.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	- title Hlasovanie strategy Dot voting options item 1 Možnosť 1 Delete item Move down item 2 Možnosť 2 Delete item Move up Add item Delete Last item Delete All body Model Example Value <pre>{ "title": "string", "strategy": "string", "options": ["string"] }</pre>			

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

Obr. 3: Dokumentácia pre volanie POST nad stránkou event_create, ktorého úlohou je vytvorenie nového hlasovania.

GET /event/{id}/ Retrieve details about certain voting, create new vote.

Implementation Notes
Retrieve details about certain voting, create new vote.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

[Try it out!](#)

Obr. 4: Dokumentácia pre volanie GET nad stránkou event/<id>, ktorého úlohou je získanie informácií o hlasovaní.

POST /event/{id}/ For certain Event create new Vote and assign its Option.

Implementation Notes
For certain Event create new Vote and assign its Option.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string
data	-		body	Model Example Value

name
Jozef

voted_item_id
2

Parameter content type: application/json

```
{
  "name": "string",
  "voted_item_id": 0
}
```

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

[Try it out!](#)

Obr. 5: Dokumentácia pre volanie POST nad stránkou event/<id>, ktorého cieľom je umožniť zahlasovať za možnosť.

PUT /event/{id}/ For certain Event change selected Option of given Vote

Implementation Notes
For certain Event change selected Option of given Vote

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string
data	- name Jozef voted_item_id 1		body	Model Example Value

name

Jozef

voted_item_id

1

Parameter content type: application/json

{

"name": "string",

"voted_item_id": 0

}

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

Obr. 6: Dokumentácia pre volanie PUT nad stránkou event/<id>, ktoré umožňuje zmenu zahlasovaných možností používateľa.

GET /event/admin/{id}/ Administration of given Event, changing Event attributes.

Implementation Notes
Administration of given Event, changing Event attributes.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

Obr. 7: Dokumentácia pre volanie GET nad stránkou event/admin/<id>, ktorý vracia informácie o hlasovaní.

POST

/event/admin/{id}/

Change attributes of given Event.

Implementation Notes

Change attributes of given Event.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string
data	- stopped true		body	Model

Example Value

```
{
  "stopped": true
}
```

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

Obr. 8: Dokumentácia pre volanie POST nad stránkou event/admin/<id>, ktorý umožňuje zmenu vlastností hlasovania.

Volania patriace do modulu autentifikácie

GET

/auth/logout/

Log out currently logged in user.

Implementation Notes

Log out currently logged in user.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page	1	A page number within the paginated result set.	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

Obr. 9: Dokumentácia pre volanie GET nad stránkou auth/logout, ktoré slúži na odhlásenie používateľa.

Príloha B: Návrhy grafických rozhraní

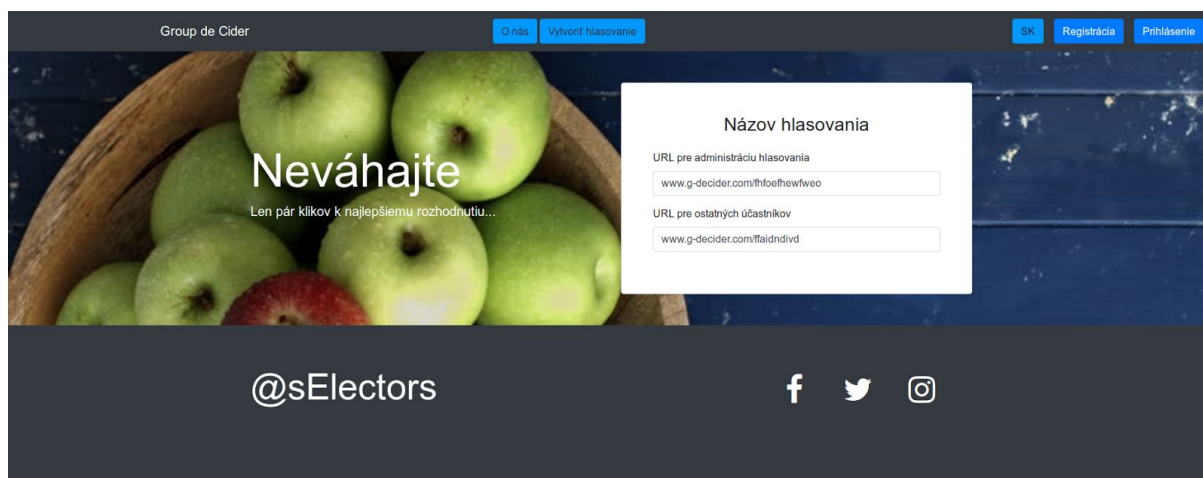
Prvotný návrh grafických rozhraní

The screenshot shows the 'Vytvor hlasovanie' (Create Poll) form in the @sElectors app. The background features a bowl of green and red apples. The form is a white modal with the title 'Vytvor hlasovanie'. It contains two input fields: 'Názov' (Name) with the placeholder 'Názov podujatia' and 'Možnosti' (Options) with the placeholder 'Možnosť 1; Možnosť 2'. A 'Dalej >>' button is at the bottom of the form. The app's header includes 'Group de Cider', 'O nás', 'Vytvor hlasovanie', 'SK', 'Registrácia', and 'Prihlásenie'. The footer shows '@sElectors' and social media icons for Facebook, Twitter, and Instagram.

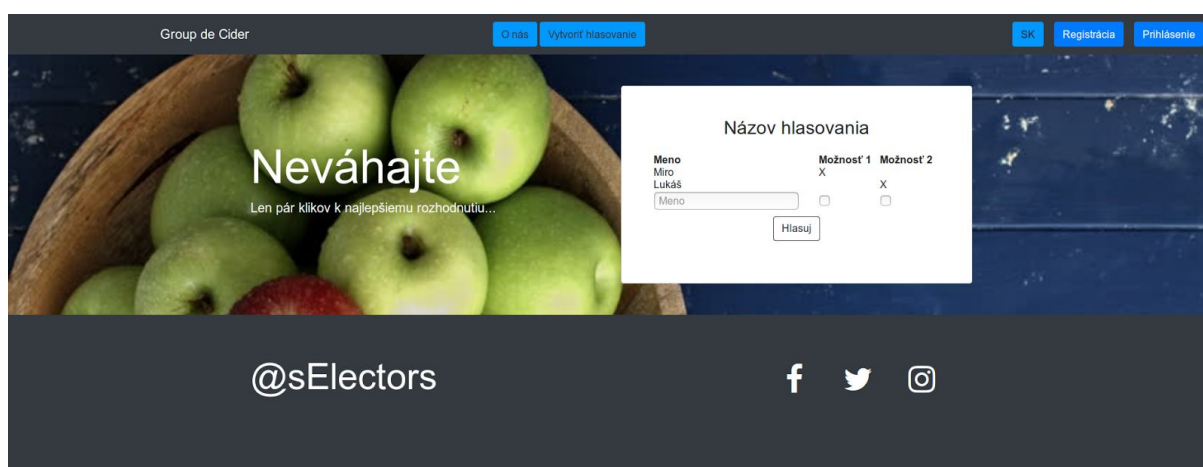
Obr. 10: Vytvorenie hlasovania - 1. obrazovka.

The screenshot shows the 'Vyber spôsob hlasovania' (Choose voting method) screen in the @sElectors app. The background features a bowl of green and red apples. The screen has a white modal with the title 'Vyber spôsob hlasovania'. It contains three radio button options: 'Možnosť č.1', 'Možnosť č.2', and 'Možnosť č.3'. A 'Vytvor hlasovanie' button is at the bottom of the modal. The app's header includes 'Group de Cider', 'O nás', 'Vytvor hlasovanie', 'SK', 'Registrácia', and 'Prihlásenie'. The footer shows '@sElectors' and social media icons for Facebook, Twitter, and Instagram.

Obr. 11: Vytvorenie hlasovania - 2. obrazovka.

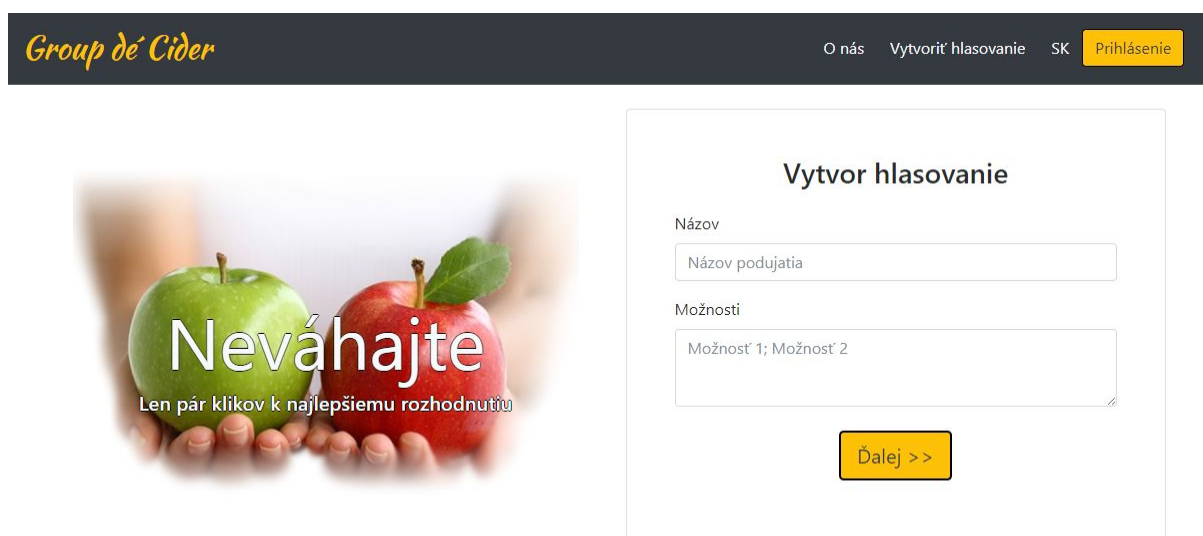


Obr. 12: Vytvorenie hlasovania - 3. obrazovka.

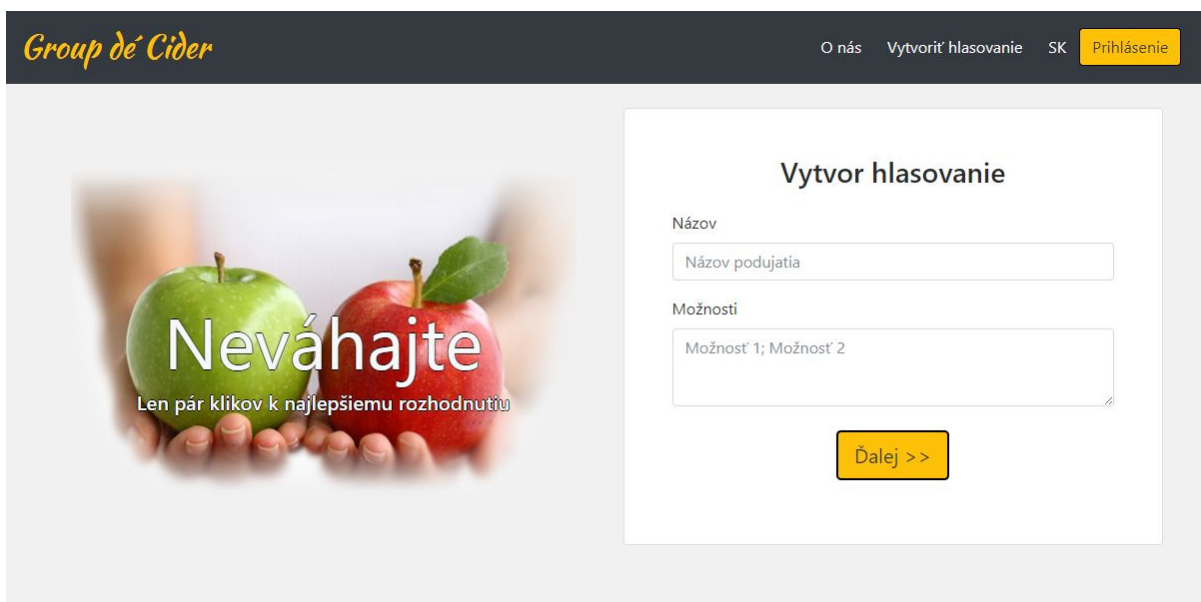


Obr. 13: Pridanie hlasu do hlasovania.

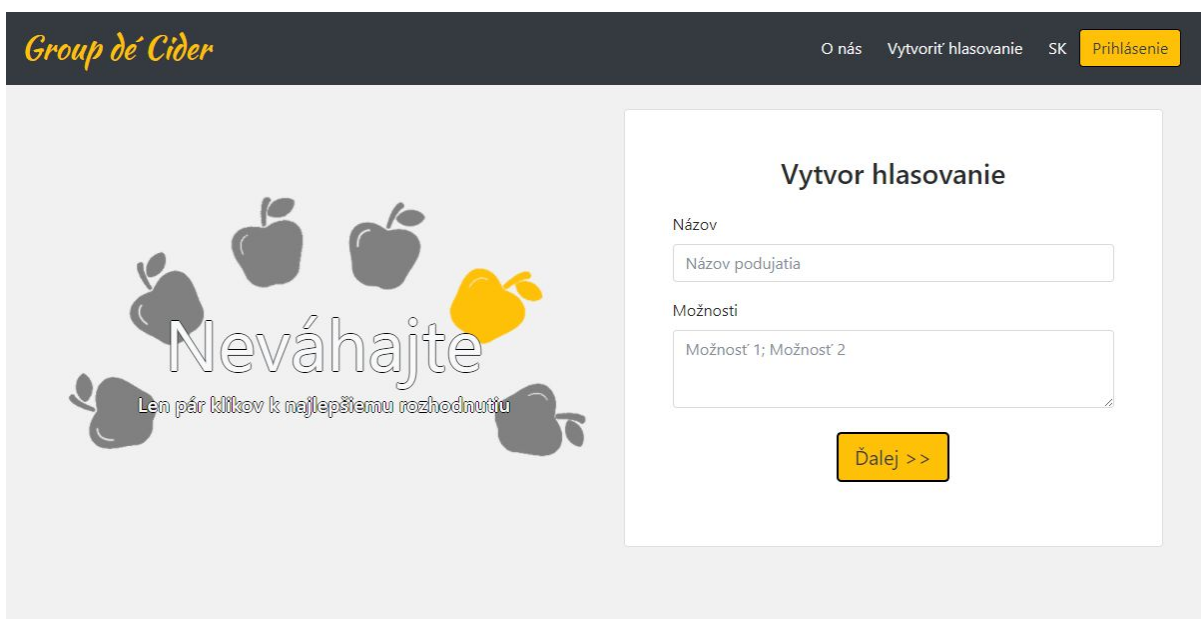
Druhá iterácia návrhu grafických rozhraní



Obr. 14: Mockup pre hlavnú obrazovku - verzia 1.



Obr. 15: Mockup pre hlavnú obrazovku - verzia 2.



Obr. 16: Mockup pre hlavnú obrazovku - verzia 3.

Tretia iterácia návrhu grafických rozhraní



Obr. 17: Mockup pre hlavnú obrazovku s vyberateľnou stratégiou.