

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 19 Bratislava 4

Projektová dokumentácia k riadeniu projektu

Tím č. 8

MOD-eli on Demand

Akademický rok:	2017/2018
Predmet:	Tímový projekt
Členovia tímu:	Bc. Mária Dlhá Bc. Štefan Grivalský Ing. Yurii Hubar Bc. Erik Chodorčuk Bc. Nadežda Juhásová Bc. Adam Strásky Bc. Roderik Williger Bc. Peter Žabka
Vedúci tímu:	Ing. Peter Pišteck, PhD.
Vlastník produktu:	Molpir, s.r.o.

Obsah

1	Úvod.....	4
2	Role členov tímu a podiel práce	5
3	Aplikácia manažmentov	7
3.1	Manažment vývoja metódou SCRUM	7
3.2	Manažment plánovania úloh.....	7
3.3	Manažment komunikácie.....	7
3.4	Manažovanie verziovania.....	8
3.5	Manažment kvality kódu	8
3.6	Manažment kvality dokumentácie.....	8
3.7	Manažment testovania	8
3.8	Manažment rizík.....	9
4	Sumarizácia šprintov	10
4.1	1. šprint – Mussomeli	10
4.1.1	Retrospektíva.....	11
4.1.2	Burndown chart	12
4.1.3	Zhodnotenie šprintu Mussomeli	12
4.2	2. šprint – Corleone	12
4.2.1	Retrospektíva.....	14
4.2.2	Burndown chart	14
4.2.3	Zhodnotenie šprintu Corleone	14
4.3	3. šprint – Trapani	15
4.3.1	Retrospektíva.....	17
4.3.2	Burndown chart	17
4.3.3	Zhodnotenie šprintu Trapani	17
4.4	4. šprint – Ribera	18
4.4.1	Retrospektíva.....	20
4.4.2	Burndown chart	21
4.4.3	Zhodnotenie šprintu Ribera	21
4.5	5. Šprint	22
4.6	Zhodnotenie šprintu.....	23
	V čase písania tohto dokumentu ešte nebol uzatvorený 5 šprint. Šprint má koniec 14. decembra a z tohto dôvodu nie je možné pridať burndown chart a retrospektívu šprintu.	23
5	Globálna retrospektíva ZS	24
7	Používané metodiky	25
7.1	Metodika písania zdrojového kódu	25
7.2	Metodika pre Code Review	25
7.3	Metodika pre Definition of Done	25

7.4	Metodika písania dokumentácie	25
8	Prílohy	26

1 Úvod

Predmetom tohto dokumentu je opísanie riadenia projektu. Dokumentácia k riadeniu projektu zachytáva celkový pohľad na priebeh a riadenie projektu v rámci tímu číslo 8 na predmete Tímový projekt. Náplňou projektu je vytvorenie novej funkcionality na systém Funtoro MOD, ktorý sa využíva v autobusoch a vlakoch. Produkt je vytváraný pre vlastníka produktu, ktorým je spoločnosť Molpir, s.r.o.. Funkcionalitu, ktorú ideme rozširovať, predstavujú dotazníky (vedomostné/zábavné/na spätnú väzbu pre personál), pomocou ktorých si môžu používatelia (cestujúci) skrátiť svoje cestovanie či už zábavnou alebo vzdelávacou formou. Dotazníky na spätnú väzbu slúžia pre vyhodnotenie personálu a prípadné zlepšenie služieb.

Obsahom dokumentu je opis jednotlivých manažérskych činností v rámci tímu, rozdelenie zodpovedností za jednotlivé časti projektu, komunikačné kanály, ktoré sú využívané členmi tímu, a taktiež spôsoby archivácie projektu. V rámci dokumentu sú taktiež vypracované metodiky, ktorými sa riadia členovia tímu číslo 8 pri vypracovávaní projektu.

V druhej kapitole sú podrobnejšie priblížené jednotlivé role členov tímu, pričom je taktiež priblížené, akým spôsobom rozkladáme zodpovednosť na jednotlivých členov tímu.

2 Role členov tímu a podiel práce

Molpir, s.r.o - Ing. Drahoš Maďar / Ing. Martin Kosa

- Vlastníkom produktu v rámci projektu tímu číslo 8 je spoločnosť Molpir, s.r.o.. V rámci konzultácií požiadaviek do styku s nami prichádzal Ing. Martin Kosa, ktorého neskôr nahradil Ing. Drahoš Maďar. Konzultanti spoločnosti Molpir, s.r.o. nám pomáhali priblížiť systém Funtoro MOD, na ktorom má byť vyvíjaná funkcionálna, doladzovali s nami biznisové požiadavky a hardwarové obmedzenia ich systému. Oba konzultanti majú bohaté skúsenosti s vývojom produktov v rámci SCRUM tímov na Slovensku aj v zahraničí.

Ing. Peter Pišteck, PhD.

- Pedagogickým vedúcim tímu číslo 8 je Ing. Peter Pišteck, PhD., ktorý má bohaté skúsenosti s vývojom softvéru v SCRUM tímoch, a taktiež skúsenosti s mentoringom tímov v rámci minuloročných tímových projektov. V rámci našich stretnutí sa snaží usmerňovať tím správnym smerom, dohliada na rozdeľovanie a plánovanie jednotlivých úloh. Priblížil nám ako funguje SCRUM, ako máme pracovať ako tím a snažil sa nás upozorniť na dôležité veci v rámci plánovania projektu a práce.

Bc. Mária Dlhá

- Mária bola v rámci tímu zodpovedná za vypracovanie jednotlivých metodík, ktorými sa náš tím riadi v rámci projektu. Má na starosti správu GitLabu a verzií kódu. Podieľa sa na tvorbe dizajnu, a taktiež na tvorbe tímových tričiek. Taktiež sa aktívne podieľa na písaní dokumentácie a jej kontrole.

Bc. Štefan Grivalský

- Štefan je jedným z hlavných vývojárov webovej aplikácie, ktorá bude slúžiť na spravovanie autobusových flotíl spoločnosťou Molpir, s.r.o.. Je hlavným zodpovedným za review Python kódov a pomoc pri problematikách, ktoré sa môžu vyskytnúť s Django. Podieľal sa na prvotnej tvorbe webovej aplikácie, multi-jazyčnosti stránky, a taktiež jej napojenie na databázu.

Ing. Yurii Hubar

- Yurii je členom tímu pôvodcom zo zahraničia, ktorý sa medzi nás výborne adaptoval. V rámci tímu je zodpovedný za štúdium SCRUM a usmerňovaním tímu pri jeho správnom používaní. Je hlavnou zodpovednou osobou za analýzu gamifikačných prvkov, ktoré budú zapracované do novej funkcionality.

Bc. Erik Chodorčuk

- Erik je zodpovedný za výber správnych vývojových technológií a nástrojov na vývoj. Aktívne sa podieľa na tvorbe webovej aplikácie v Pythone, a taktiež sa podieľal na tvorbe metodík pre programovanie v Javascripte a HTML. Podieľa sa na review zdrojových kódov a na tvorbe správy používateľských účtov vo webovej aplikácii.

Bc. Nadežda Juhášová

- Nadežda je zodpovedná osoba za JIRU a jej správu. Aktívne pomáha členom tímu pri práci s JIRA nástrojom a je zodpovedná za správne vkladanie a pridelovanie úloh. V rámci tímu sa taktiež podieľa na archivácii a kontrole dokumentácie.

Bc. Adam Strásky

- Adam je hlavným zodpovedným za vývoj aplikácie na systém Funtoro MOD. Aktívne sa podieľa na tvorbe aplikácie na prácu s dotazníkmi na systém Funtoro MOD, pričom pracoval na medzi-serverovej komunikácii, správe a spracovaní dát medzi servermi a databázou. Taktiež sa aktívne podieľa na review zdrojových kódov písaných v Javascripte, HTML a PHP.

Bc. Roderik Williger

- Roderik je hlavným zodpovedným za písanie a finálnu úpravu dokumentácie. Popri dokumentácií sa aktívne podieľa na vývoji funkcionality na systém Funtoro MOD, pričom pracoval na spracovaní textov z obrazovky do databázy a zároveň aj správou databázy. Taktiež sa podieľa na review zdrojových kódov v Javascript, HTML a PHP.

Bc. Peter Žabka

- Peter je SCRUM master tímu číslo 8. Je jedným z najskúsenejších členov tímu či už z pohľadu vedenia tímu, alebo analýzy požiadaviek. Aktívne sa podieľa na vedení a usmerňovaní členov tímu a pomáha pri všetkých problémoch. Je zodpovedným za kontrolu pri plnení všetkých úloh. Popri tejto práci sa aktívne podieľa na tvorbe webovej aplikácie na správu majiteľov flotíl, pričom sa podieľal na zmene obsahu webovej stránky podľa jednotlivých používateľov.

Podiel práce na dokumentácii:

Tabuľka 1. Podiel práce na dokumentácii

Meno	Dokument	Časť	Percentuálna časť v rámci celej dokumentácie
Bc. Mária Dlhá	Dokumentácia obsahu	Spísanie časti dokumentu	28%
	Dokumentácia riadenia	Metodika DoD, Metodika Code Review, Metodika písania dokumentácie	
Bc. Yurii Hubar	Dokumentácia obsahu	Analýza gamifikačných prvkov	6%
Bc. Erik Chodorčuk	Dokumentácia riadenia	Metodika písania zdroj. kódu	6%
Bc. Roderik Williger	Dokumentácia obsahu	Spísanie finálneho dokumentu	60%
	Dokumentácia riadenia	Spísanie finálneho dokumentu	

3 Aplikácia manažmentov

3.1 Manažment vývoja metódou SCRUM

Obsah predmetu tímový projekt predpisuje pre účely vývoja produktu vývojovú metódu SCRUM. Počas prvého stretnutia sme sa zaoberali jednotlivými úlohami v SCRUMe a vybrali sme nášho Scrum Master. Po dôkladnej diskusii a analýze sme dospeli k záveru, že táto rola ostane stacionárna a nebude sa meniť, keďže zvolený Scrum Master mal všetky predpoklady na to, aby túto úlohu plnohodnotne zvládal počas celej doby projektu.

Úlohy Scrum Master-a v rámci tímu číslo 8:

- Aktívna komunikácia s členmi tímu.
- Komunikácia a riešenie problémov.
- Zabezpečenie, aby sa všetci členovia tímu cítili spokojne v rámci svojej práce.
- Dohliadanie na plnenie úloh v časových intervaloch.

Pri úlohe Scrum Mastera je veľmi dôležitým prvkom komunikácia s členmi tímu. Jeho úlohou je zabezpečiť, aby sa žiaden člen necítil v rámci tímu nepríjemne, a aby nevznikali zbytočné problémy. Zároveň ak už nejaký problém vznikne, či už z dôvodu toho, že nejaký člen tímu nezvláda svoje úlohy alebo ich zanedbáva, je povinný pracovať na odstránení týchto problémov.

3.2 Manažment plánovania úloh

Pre spravovanie jednotlivých šprintov a úloh sme si zvolili nástroj JIRA. Nástroj JIRA je komerčný a platený avšak s možnosťou získania študentskej bezplatnej licencie, o ktorú sme požiadali na začiatku prvého šprintu. Licencia nám bola na školské účely schválená bezplatne. Nástroj JIRA priamo umožňuje vytváranie backlogu, user stories a šprintov.

Plánovanie úloh prebieha tak, že všetky požiadavky a úlohy, ktoré sme identifikovali si rozdelíme do user stories, ktoré môžeme vidieť v backlogu. User stories môžeme potom zaraďovať do jednotlivých šprintov. Každú user story podrobne rozoberieme, popíšeme a rozdelíme na podúlohy. Následne si ohodnocujeme jednotlivé podúlohy počtom bodov, ktoré znázorňujú ich náročnosť. V tomto kroku sa odlišujeme, pretože neohodnocujeme priamo celé user story, ale cena user story sa stanoví podľa súčtu cien jednotlivých podúloh.

Pri zaraďovaní user stories do šprintu prideliťujeme podúlohy jednotlivcom, ktorí sú za danú úlohu zodpovední. Po zaradení úlohy do šprintu vnímame v rámci podúlohy tri základne stavy, a to nerozpracovaná, rozpracovaná a dokončená. Šprint sa považuje za úspešný ak boli všetky podúlohy, a tým pádom aj user stories označené za dokončené.

3.3 Manažment komunikácie

Ako nástroj pre tímovú komunikáciu sme chceli zvoliť niečo, kde budeme môcť prehľadne kontrolovať tok správ a úloh. Ako vhodný nástroj sme vybrali nástroj Slack. Nástroj umožňuje rozdeliť komunikáciu do viacerých kanálov. Jednotlivé kanály sme pomenovali podľa okruhu problematiky, ktorou sa zaoberajú, a ktorá sa v nich rieši. Táto možnosť nám výrazne sprehľadnila komunikáciu a rýchlu nápravu vzniknutých problémov. Okrem kanálov sa dá komunikovať aj jednotlivito medzi členmi tímu. Ak sú problémy, ktoré netreba riešiť v rámci celého tímu, je možné vytvoriť komunikáciu len medzi pár členmi, poprípade len medzi dvojicou členov tímu.

Ako druhý oficiálny komunikačný kanál sme si vytvorili spoločný tímový email tim8fiit@gmail.com, prostredníctvom ktorého riešime dôležitú komunikáciu s vlastníkom produktu alebo vedúcim tímu. Zdieľanie emailu funguje tak, že každý kto odošle správu sa musí pod správu podpísať svojim menom. Druhá podmienka je, že ak niekto prečíta správu, je potrebné o tejto správe informovať ostatných členov tímu na Slacku.

3.4 Manažovanie verziovania

Na správu jednotlivých verzií kódu a dokumentácie sme sa po prvotnej analýze a rozhovoroch rozhodli pre nástroj GitLab. Prostredníctvom nástroja GitLab môžeme prehľadne uchovávať jednotlivé verzie zdrojových kódov a aj dokumentácií, pričom k starším verziám sa vieme jednoducho vrátiť späť. Nástroj nám poskytuje vo webovom prostredí prehľadné grafické rozhranie, kde môžeme prísť k zdrojovým kódom a dokumentácii a pozorovať ich zmeny a aj to, kým boli tieto zmeny vykonané.

Dôležitým faktorom pri voľbe úschovy verzií zdrojových kódov bola bezpečnosť. Keďže sme podpísali zmluvu o mlčanlivosti a nechceli sme ani zbytočne vypúšťať na web zdrojové kódy, hľadali sme riešenie, ktoré umožní uschovávať verzie len pre pridaných členov. Možnosť súkromného repozitára ponúka viacero nástrojov, avšak pri nástroji GitLab bola táto možnosť bezplatná, a preto sme sa rozhodli pre tento nástroj.

3.5 Manažment kvality kódu

Kvalita kódu je kontrolovaná prostredníctvom nástroja GitLab. Kontrola kódu prebieha po tom, ako autor kódu odovzdá kód do GitLabu a vykoná požiadavku Merge Request. V Merge Requeste je označená osoba, ktorá má kód skontrolovať. Tejto osobe príde upozornenie a následne skontroluje kód. Jednotlivé pripomienky ku kódu môže opoznámkovať priamo vo webovom rozhraní, čo kontrolórovi kódu výrazne uľahčuje prácu. Na kontrolovanie kódu bola spísaná metodika pre Code Review, podľa ktorej sa musí riadiť aj autor kódu a aj kontrolór.

Kvalita kódu sa posudzuje aj testami, ktoré prebiehajú v jednotlivých šprintoch, pričom sa kontroluje funkčnosť a výkonnosť zdrojového kódu vždy v rámci členov tímu, ktorí sa podieľajú na riešení rovnakých častí projektu. Nikdy však nemôže byť kontrolórom kódu len osoba, ktorá kód vytvárala.

3.6 Manažment kvality dokumentácie

Dokumentácie sú tvorené pomocou nástroja Microsoft Word, ktorý umožňuje písanie poznámok k danej dokumentácii. Autor kódu zozdieľal dokumentáciu s kontrolórom, ktorý následne skontroluje dokument a komentármi navrhne zmeny a úpravy, ktoré požaduje od autora dokumentu. Autor následne prevezme okomentovaný dokument späť a opravuje dokument podľa požiadaviek. Ako nástroj bol zvolený Microsoft Word preto, že možnosti revízie dokumentov sú na vysokej úrovni a nástroj si dokáže ukladať aj viaceré verzie dokumentov.

3.7 Manažment testovania

Testovanie v projekte je vykonané vždy medzi dvojicou členov tímu. Ak sa test týka väčšej časti funkcionality môže sa testovania zúčastniť aj viacero členov tímu. Testy sú potrebné, aby nedochádzala k chybnému splneniu úloh v user stories, a aby všetka pridaná funkcionality zachovávala dostatočnú kvalitu. Vykonávateľ úlohy musí pred tým, ako úlohu označí za dokončenú, kontaktovať iného člena tímu, s ktorým sa dohodne na testovaní svojej úlohy a následne úlohu otestujú.

V prípade, že testovanie prebehne bez problémov a úloha spĺňa parametre Definition of Done, môže byť táto úloha označená za dokončenú.

V prípade, že sa pri testovaní vyskytli nejaké problémy, je potrebná analýza problémov testerom a vykonávateľom úlohy. Po analýze je potrebné navrhnúť zmeny na opravu, ktoré je povinný vykonávateľ úlohy vykonať. V prípade, že zmenu nestíha vykonať sám požiada o pomoc testera poprípade iného člena tímu.

3.8 Manažment rizík

Za riziko sa pokladá stav, ktorý môže nejakým spôsobom ovplyvniť neúspech vykonanie úlohy poprípade výraznejšie poškodiť tímovú prácu.

Medzi rizika, ktorými sme sa zaoberali patria vedomé absencie členov tímu na stretnutiach, choroby a nečakané udalosti, ktoré človek nie je schopný ovplyvniť.

Za vedomú absenciu na stretnutí sa považuje stav, kedy člen tímu na stretnutie mohol prísť, ale uprednostnil iné aktivity. Tento druh absencie je nežiadúci avšak môže sa vyskytnúť a spôsobiť problémy. Pre minimalizáciu týchto absencií sme v rámci tímu zaviedli systém pokút, ktorý je člen tímu v prípade absencie alebo meškania na stretnutie povinný zaplatiť. Tento systém sa nám osvedčil a minimalizovali sme v rámci tímu absencie.

Ďalším rizikom sú situácie, ktoré človek nedokáže ovplyvniť ako sú autonehoda, kolóna alebo choroba a v tom prípade nemôžeme člena tímu nijakým spôsobom trestať. V prípade choroby sa však musí počet úloh šprintu zredukovať poprípade prerozdeliť medzi ostatných členov tímu.

4 Sumarizácia šprintov

V tejto kapitole sú zosumarizované jednotlivé šprinty. Podkapitoly popisujú jednotlivé user stories, retrospektívu a tabuľky znázorňujúce priebeh šprintu. Názvy šprintov sme volili podľa talianskych mafiánskych rodín.

4.1 1. šprint – Mussomeli

Prvý šprint sa zaoberal analýzou servera Funtoro MOD, rozbeháním FTP a inštaláciou servera, tvorbou šablóny dokumentácií, analýzou SCRUM a tvorbou plagátu. Prvotné úlohy mali tvoriť základ pre ďalšiu prácu a na týchto úlohách sme sa snažili počas prvého šprintu vyskúšať ako dokážeme pracovať v tíme.

Tabuľka 2. Tabuľka s úlohami šprintu a ich pridelenie k členom tímu

User story	Úloha	Zodpovedná osoba
Vyhľadať nástroj na SCRUM projekt	Analyzovať nástroje na SCRUM	Nadežda Juhásová
	Vytvoriť SCRUM projekt na prvý šprint	Štefan Grivalský
Rozbehať FTP a nainštalovať server	Rozbehať FTP	Adam Strásky
	Nainštalovať server	Peter Žabka
Aktualizovať labák	Spojiť sa s ľuďmi z firmy Molpir	Erik Chodorčuk
	Vymeniť staré HD obrazovky za nové SHD	Yurii Hubar / Erik Chodorčuk
Analyzovať SCRUM	Vykonať a spísať analýzu SCRUM	Yurii Hubar
Splniť úlohy na prednášku TP	Vytvoriť stránku TP	Peter Žabka
	Vytvoriť plagát	Erik Chodorčuk / Štefan Grivalský
Spravovať dokumentáciu	Vytvoriť formu dokumentácie	Roderik Williger
	Spísať poznámky a vytvoriť záznamy	Mária Dlhá

Analyzovať nástroje na SCRUM

- Pre lepšiu organizáciu tímu a šprintov bolo pre tím potrebné zanalyzovať nástroje, ktoré nám umožnia pracovať na projekte ako SCRUM tím. Na základe analýzy mal byť vybraný najlepší možný nástroj. Výsledkom tejto analýzy bol výber nástroja JIRA, na ktorý sme však potrebovali licenciu, a preto sme dočasne zvolili nástroj Trello.

Vytvoriť SCRUM projekt na prvý šprint

- Po analýze nástrojov na SCRUM projekt, bolo potrebné vytvoriť v nástroji projekt a popridávať členov tímu.

Nainštalovať server

- Na prácu na tímovom projekte sme dostali školský server, ktorý bolo treba nainštalovať, aby bola na ňom umožnená ďalšia práca. Tento krok bol základným pre uskutočnenie ďalšej práce.

Rozbehať FTP

- Na prácu so serverom bolo potrebné rozbehať protokol prenosu súborov FTP. Tento krok bol nevyhnutný na to, aby sa dalo nad serverom manipulovať s dátami.

Spojiť sa s ľuďmi z firmy Molpir

- Pred začatím všetkej práce na projekte, bolo potrebné aktualizovať školské laboratórium firmy Molpir, pretože softvér má byť vyvíjaný na nové typy obrazoviek, ktoré museli byť dodané. Dohodnutie stretnutia na výmenu bolo predmetom tejto úlohy.

Vymeniť staré HD obrazovky za nové SHD

- Predmetom tejto úlohy bolo zbaliť celú stenu obrazoviek a nahradiť ju novými obrazovkami za pomoci technikov z firmy Molpir.

Vykonať a spísať analýzu SCRUM

- Aby sme mohli pracovať ako SCRUM tím, museli sme vedieť ako SCRUM funguje. Z tohto dôvodu sme zadali úlohu na analýzu SCRUM-u, aby nám jeden z členov tímu sprostredkoval informácie a usmernil nás pri našej práci.

Vytvoriť stránku TP

- Predmetom tejto úlohy bolo vytvorenie stránky tímu, ktorá bola potrebná na ďalšie pokračovanie na predmete, aby sme na nej mohli dokumentovať naše pokroky a zviditeľniť tím.

Vytvoriť plagát

- Predmetom tejto úlohy bolo vytvorenie plagátu na predmet Tímový projekt.

Vytvoriť formu dokumentácie

- Pre lepšiu prácu a manipuláciu s dokumentami bolo potrebné vytvoriť jednotnú formu dokumentácie a zápisov zo stretnutí.

Spísať poznámky a vytvoriť záznamy

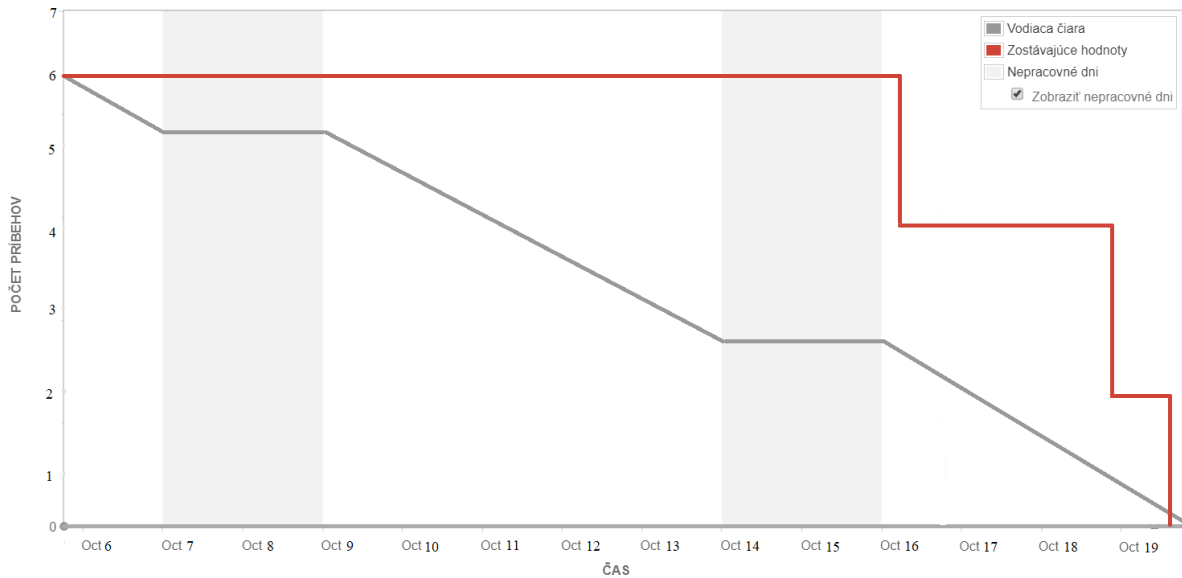
- Poznámky zo všetkých stretnutí, ktoré boli napísané ručne bolo potrebné prepísať do šablón, ktoré boli na to určené a bolo treba spísať zápisnice stretnutí.

4.1.1 Retrospektíva

Tabuľka 3. Retrospektíva šprintu

START	Začať sa viacej stretávať – Stand up
KEEP	Rovnomerné rozdelenie práce
STOP	Viacero komunikačných kanálov

4.1.2 Burndown chart



Obrázok 1 znázorňuje Burndown chart prvého šprintu

4.1.3 Zhodnotenie šprintu Mussomeli

Prvý šprint sa niesol v znamení základných úloh a prvotnej tímovej práce. Na začiatku sme si stanovili základne úlohy, ktoré sme sa snažili rovnomerne rozdeliť medzi členov tímu. Snažili sme sa do šprintu zaradiť úlohy, ktoré by sme reálne mohli stihnúť, a ktoré boli potrebné pre ďalšiu prácu na projekte. Všetky úlohy sa nám podarilo splniť, a tým pádom sa považoval prvý šprint za úspešný.

4.2 2. šprint – Corleone

Druhý šprint sa zaoberal základnými prvkami webovej aplikácie, zmeny na serveri Funtoro MOD, vytvorením prvých metodík a vytvorení SCRUM projektu v nástroji JIRA, na ktorú sme vtedy dostali bezplatnú študentskú licenciu. V tomto šprinte sme sa snažili vytvoriť základy a naučiť sa pracovať s nástrojmi, ktoré sú nutne potrebné na ďalšie pokračovanie projektu, avšak doteraz sme sa s nimi ešte moc nestretli.

Tabuľka 4. Tabuľka s úlohami šprintu a ich pridelenie k členom tímu

User story	Úloha	Zodpovedná osoba
Vytvorenie SCRUM projektu v nástroji JIRA	Premigrovať úlohy z trella	Nadežda Juhásová
	Vytvorenie SCRUM projektu	Nadežda Juhásová
Vytvoriť 2 metodiky	Metodika na kód	Erik Chodorčuk
	Metodika na pull requesty	Erik Chodorčuk
Viacjazyčná stránka s prihlásením	Vybrať rozumnú technológiu + DB	Roderik Williger
	Naprogramovať základnú stránku	Roderik Williger
	Podporovať zmenu jazyka	Štefan Grivalský
	Vytvorenie prihlásenia	Erik Chodorčuk
Jednoduchá zmena na obrazovke	Zohnať čo najviac informácií o servere Funtoro MOD	Adam Strásky
	Urobiť zmenu na obrazovke	Adam Strásky
Vytvoriť repozitár na úschovu zdrojových kódov a review	Analyzovať online nástroj GIT	Mária Dlhá
	Vytvoriť repozitár vo vhodnom nástroji	Mária Dlhá

Premigrovať úlohy z trela

- Po získaní licencie na JIRU a po jej nainštalovaní bolo potrebné premigrovať úlohy z nástroja Trello

Vytvorenie SCRUM projektu

- Po nainštalovaní nástroja JIRA bolo potrebné vytvoriť SCRUM projekt a vytvoriť v ňom prvé šprinty. Bolo potrebné naštudovať ako funkcie nástroja JIRA prebiehajú.

Metodika na kód

- Na písanie typovo jednotného zdrojového kódu bolo potrebné spísať metodiku, podľa ktorej sa budú programátori riadiť. Vďaka tomuto postupu by malo byť zabezpečené, aby bol kód pre každého z tímu prehľadný a mohol sa k nemu vyjadriť.

Vybrať rozumnú technológiu + DB

- Na prácu na webovej aplikácii bude potrebná vhodná technológia, ktorú sa budú môcť naučiť viacerí členovia tímu a bude pomocou nej možné vytvoriť kvalitnú webovú aplikáciu. Zároveň bolo potrebné vybrať voľne dostupnú databázu, ktorá bude spolupracovať s webovou aplikáciou. Po analýze sme zvolili framework Django jazyku Python a databázu PostgreSQL.

Naprogramovať základnú stránku

- Vo vybranej technológii bolo potrebné položiť základ webovej aplikácie vytvorením prvej statickej stránky. Vďaka tomuto kroku sme vedeli v tíme rozšíriť vedomosť o používaní technológie.

Podporovať zmenu jazyka

- Keďže webová aplikácia je určená pre medzinárodných prepravcov je nutná multi-jazyčnosť stránky. Z tohto dôvodu bolo treba naprogramovať základnú zmenu jazyka, ktorá bude v ďalších etapách projektu rozširovaná.

Vytvorenie prihlásenia

- Vo webovej aplikácii budú vystupovať dva typy používateľov. Jeden typ používateľa bude Molpir a druhý typ používateľa bude majiteľ autobusovej flotily. Majiteľov autobusovej flotily môže byť však viac, a preto treba zabezpečiť prihlasovanie. Predmetom tejto úlohy bolo vytvoriť základné prihlásenie.

Zohnať čo najviac informácií o serveri Funtoro MOD

- O serveroch Funtoro MOD existuje len málo dostupných informácií a z tohto dôvodu bolo treba vykonať kvalitnú analýzu a kontaktovať ľudí, ktorí sa už do styku so serverom dostali.

Urobiť zmenu na obrazovke

- Po získaní informácií sme sa mali napojiť na server a pokúsiť sa vykonať nejaké viditeľné zmeny na obrazovke.

Analyzovať online nástroj GIT

- Na archiváciu zdrojových kódov a dokumentáciu sme potrebovali vybrať nástroj, kde sa budú dať vhodne ukladať súbory. Bolo potrebné analyzovať nástroj GIT, ktorý je všeobecne známy a určený na tento typ práce. Pre náš projekt bolo potrebné vybrať nástroj, kde bude možné vytvoriť súkromný repozitár.

Vytvoriť repozitár vo vhodnom nástroji

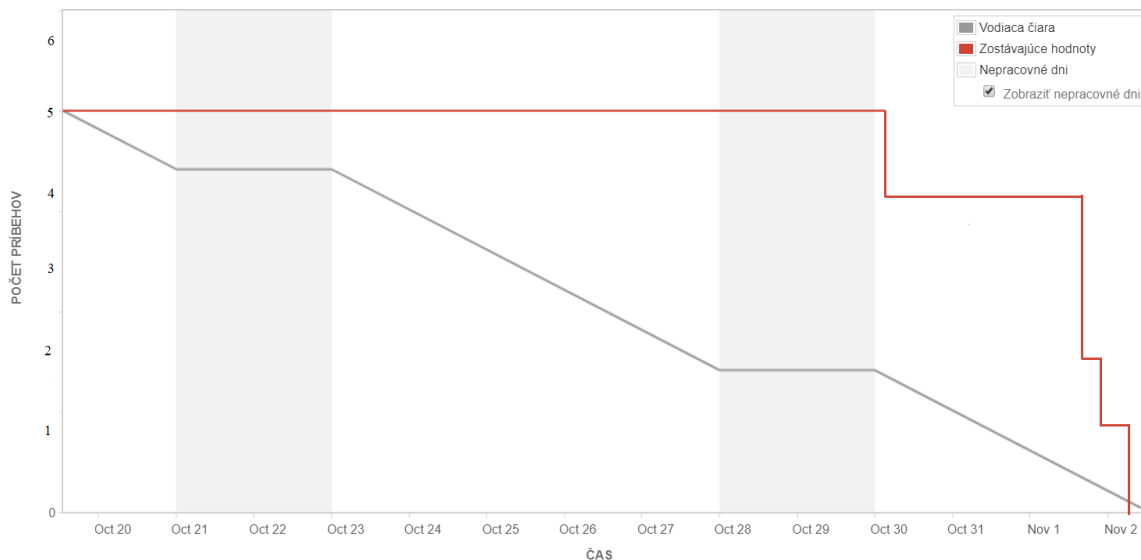
- Po analýze sme vybrali nástroj GitLab, ktorý umožňoval vytvoriť bezplatne súkromný repozitár. V tejto úlohe bolo potrebné založiť účet a zaregistrovať všetkých členov tímu.

4.2.1 Retrospektíva

Tabuľka 5. Retrospektíva šprintu

START	Zlepšiť iniciatívu Zlepšiť komunikáciu na stretnutiach
KEEP	Vytváranie zápisníc Pravidelnosť teambuildingov Vykonávanie pokrokov
STOP	Minimalizovať FB komunikáciu

4.2.2 Burndown chart



Obrázok 2 znázorňuje Burndown chart druhého šprintu

4.2.3 Zhodnotenie šprintu Corleone

Šprint Corleone sa niesol v znamení prvého programovania. Začali sme tvoriť webovú aplikáciu a prvé zmeny na serveri Funtoro MOD. Ďalším cieľom bolo využiť licenciu, ktorá nám bola schválená na nástroji JIRA a vytvoriť repozitár na archiváciu zdrojových kódov. Na vykonávanie činností bolo potrebné taktiež vytvoriť prvé metriky. Vytýčené ciele sme si v riadnom čase všetci splnili, čiže šprint sa považoval za úspešný, avšak keďže sa jedná o jeden z prvých šprintov je možné, že všetci mali ešte časové rezervy, ktoré môžu byť do ďalších šprintov obsadené ďalšou prácou.

4.3 3. šprint – Trapani

Tretí šprint mal byť prvým výraznejším pokrokom na projekte. Po prvých dvoch šprintoch, na ktorých sme si mohli odskúšať prácu v tíme, sme chceli v treťom šprinte využiť všetok dostupný čas a nenechávať zbytočne voľné kapacity. Medzi úlohy zaradené do tohto šprintu patrila prvá komunikácia medzi obrazovkami a serverom mimo autobusu, pokrok na webovej aplikácii pre majiteľov flotíl autobusov, spísanie ďalších metodík a analyzovať gamifikačné prvky, ktoré budú neskôr užitočné pri tvorbe funkcionality.

Tabuľka 6. Tabuľka s úlohami šprintu a ich pridelenie k členom tímu

User story	Úloha	Zodpovedná osoba
Správa zákazníkov vo webovej aplikácii	Vytvoriť a napojiť PostgreDB	Štefan Grivalský
	Vytváranie, zmena a zmazanie používateľských účtov	Erik Chodorčuk
	Zobrazenie špecifického obsahu pre prihláseného používateľa	Peter Žabka
Odoslanie textu z obrazoviek cez server ďalej	Vytvorenie editačného poľa + zobrazenie klávesnice	Nadežda Juhásová
	Vytvorenie databázy na Funtoro MOD	Roderik Williger
	Ukladanie textov z obrazoviek do DB	
	Vytvorenie aplikácie na odosielanie textu z DB na server mimo bus	Adam Strásky
Zistiť vhodné gamifikačné prvky	Spracovanie textu, ktorý príde z busu	
	Analýza prvkov	Yurii Hubar
	Konzultácia s odborníkom	
Spracovanie výsledkov		
Spraviť metodiku CR, DoD a dokumentácie	Metodika Code Review	Mária Dlhá
	Metodika Definition of done	
	Metodika na dokumentáciu	

Vytvoriť a napojiť PostgreDB

- V rámci webovej aplikácie je potrebné, aby bola webová aplikácia prepojená s databázou. Ako databázu sme zvolili PostgreSQL a predmetom tejto úlohy bolo prepojenie webu s databázou.

Vytváranie, zmena a zmazanie používateľských účtov

- Na správu webovej aplikácie je potrebná správa používateľských účtov. Je potrebné, aby administrátor mal možnosť vytvárať, meniť a mazať používateľské účty.

Zobrazenie špecifického obsahu pre prihláseného používateľa

- Pre rôzne používateľské roly sa musí zobrazovať iný obsah. Každý používateľ môže pracovať len s právami, ktoré sú mu pridelené.

Vytvorenie editačného poľa + zobrazenie klávesnice

- Predmetom tejto úlohy bolo vytvorenie editačného poľa a zobrazenie klávesnice na písanie textu. Po analýze sme zistili, že možnosť zobrazenia klávesnice je súčasťou displeja.

Vytvorenie databázy na Funtoro MOD

- Na prácu s dátami na obrazovkách treba mať databázu na serveri, ktorá bude slúžiť na ukladanie dát. Priamo na serveri sa nachádza MySQL databáza, avšak server ma obmedzené práva na použiteľnosť databázy.

Ukladanie textov z obrazoviek do DB

- Po vytvorení databázy, bolo potrebné prepojiť obrazovky s databázou, aby sa mohla uchovávať spätná väzba.

Vytvorenie aplikácie na odosielanie textu z DB na server mimo bus

- Texty z databázy, je potrebné sprostredkovať mimo autobus z dôvodu, že bude viacero autobusových serverov a dáta chceme mať na jednom spoločnom serveri.

Spracovanie textu, ktorý príde z autobusu

- Text, ktorý príde z autobusu je potrebné spracovať priamo na serveri. V tejto úlohe sme spracovali dáta a zobrazovali ich v tabuľke.

Analýza prvkov

- Pri vytváraní funkcionality je potrebné zapracovať gamifikačné prvky. V rámci tejto úlohy bolo potrebné vykonať kvalitnú analýzu.

Konzultácia s odborníkom

- Na základe analýzy bola potrebná konzultácia s odborníkom, ktorý nás usmernil.

Spracovanie výsledkov

- Výsledky analýzy a konzultácie s odborníkom bolo potrebné spracovať do prehľadnej podoby.

Metodika Code Review

- Na kontrolu kódov bolo potrebné vytvoriť jednotný postup, ktorý je predmetom tejto metodiky.

Metodika Definition of Done

- Každá úloha musí mať svoju Definition of Done. Z tohto dôvodu vznikla metodika Definition of Done, ktorá predpisuje, kedy sa základné typy úloh považujú za splnené.

Metodika na dokumentáciu

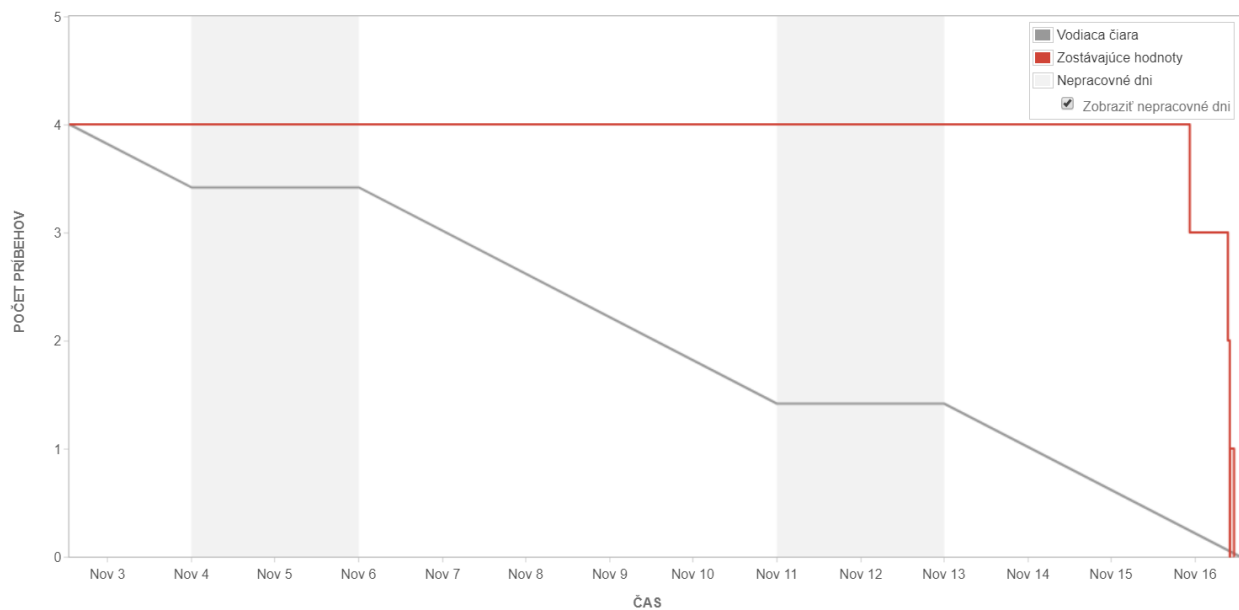
- Z dôvodu písania jednotnej dokumentácie vznikla metodika na písanie dokumentácie, ktorá predpisuje štýly, písmo a mnohé ďalšie pravidlá na písanie dokumentácie.

4.3.1 Retrospektíva

Tabuľka 7. Retrospektíva šprintu

START	Odvzdávať úlohy skôr ako posledné dni
KEEP	Dobrá komunikácie Riešenie problémov
STOP	Vymeškávanie stretnutí

4.3.2 Burndown chart



Obrázok 3 zobrazuje Burndown chart tretieho šprintu.

4.3.3 Zhodnotenie šprintu Trapani

Tretí šprint priniesol prvé výraznejšie pokroky, vďaka ktorým môžeme v ďalších šprintoch napredovať. Objavili sa však aj prvé ťažkosti a jedna úloha nemohla byť úplne dokončená, a preto bola presunutá mimo tento šprint a zaradená do ďalšieho. Táto skutočnosť však nebola moc podstatná, lebo na tejto úlohe nestáli žiadne ďalšie a podarilo sa nám splniť všetky ostatné úlohy.

4.4 4. šprint – Ribera

Štvrtý šprint mal byť ďalším výraznejším pokrokom na projekte. Medzi úlohy zaradené do tohto šprintu patrilo vytvorenie systému logovania v systéme Funtoro MOD, ale taktiež aj vo webovej aplikácii. Ďalšiu úlohu vo webovej aplikácii predstavovalo vytvorenie účtov u zákazníka a taktiež aj vyriešenie prenesenej úlohy z 3. šprintu. Ďalšími úlohami bol prenos súboru zo servera na autobus. Ďalšie úlohy predstavovali spísanie dokumentácie riadenia a obsahu a vytvorenie tímových tričiek.

Tabuľka 8. Tabuľka s úlohami šprintu a ich pridelenie k členom tímu

User story	Úloha	Zodpovedná osoba
Vyrobiť tímové trička	Vytvoriť návrh trička	Mária Dlhá
	Poslať trička na výrobu	
Vytvoriť systém logovania	Návrh formátu	Adam Strásky
	Vytvoriť logger	
	Zpracovanie logov do existujúceho kódu	
Vytvoriť systém logovania vo webovej aplikácii	Návrh formátu	Peter Žabka
	Navrhnuť model prístupových práv	
	Vytvorenie funkcie na zaznamenávanie logu	
	Zpracovanie logovania do existujúceho systému	
Vytváranie účtov vo webovej aplikácii u zákazníka	Zobrazenie používateľov skupiny	Erik Chodorčuk
Small issues+ sprint fixies	Rozbehať viacjazyčnosť na celej stránke	Štefan Grivalský
	Zmena formátu dokumentu o gamifikačných prvkoch	Yurii Hubar
	Zriadenie jednotného miesta pre ukladanie dokumentácie	Nadežda Juhásová
	Vytvoriť ukázkovú šablónu na dokumentáciu	Mária Dlhá
Spracovanie dokumentácie pre prvé odovzdanie	Dokumentácia riadenia	Roderik Williger
	Dokumentácia obsahu	Mária Dlhá
	Úpravy a doplnenie textov	
Návrh použitia gamifikačných prvkov	Vybrať najlepšie gamifikačné prvky a mechaniky	Yurii Hubar
	Navrhnuť scenáre	
Prenos súboru zo servera na autobus	Vytvorenie rozhrania na vloženie súboru	Adam Strásky
	Vytvorenie aplikácie na odoslanie súboru zo servera na autobus	

Vytvoriť návrh trička

- V rámci tejto úlohy bolo potrebné vytvoriť návrh tímových tričiek, ktoré mali byť následne schválené tímom.

Poslať trička na výrobu

- Po schválení tričiek bolo potrebné nájsť vhodného výrobcu a nechať trička vyrobiť.

Návrh formátu

- V rámci tejto úlohy bolo potrebné vytvoriť formát logu.

Vytvoriť logger

- V rámci tejto úlohy bolo potrebné vytvoriť logger na zaznamenávanie logov.

Zpracovanie logov do existujúceho kódu

- Zpracovanie navrhnutého formátu do kódu.

Návrh formátu vo webovej aplikácii

- V rámci tejto úlohy bolo potrebné vytvoriť vhodný formát logu vo webovej aplikácii.

Navrhnuť model prístupových práv

- Bolo potrebné navrhnuť model prístupových práv na základe zmeny používateľa – zmena obsahu, priradenie obsahu. Vytvorenie prístupových práv na logy.

Vytvorenie funkcie na zaznamenávanie logu

- V rámci tejto úlohy bolo potrebné naprogramovať funkciu, pomocou ktorej sa budú zaznamenávať logy.

Zpracovanie logovania do existujúceho systému

- Naprogramovanú funkciu bolo potrebné zapracovať do súčasného rozbehnutého projektu.

Zobrazenie používateľov skupiny

- V rámci tejto úlohy bolo potrebné zariadiť, aby admin zákazníka mohol vytvárať, meniť a deaktivovať nové používateľské kontá pre pracovníkov.

Rozbehať viacjazyčnosť na celej stránke

- V rámci tejto úlohy bolo potrebné rozbehať viacjazyčnosť v rámci celej webovej stránky.

Zmena formátu dokumentu o gamifikačných prvkoch

- V rámci tejto úlohy bolo potrebné upraviť dokument.

Zriadenie jednotného miesta pre ukladanie dokumentácie

- V rámci tejto úlohy bolo potrebné zriadiť jednotný repozitár na archiváciu dokumentov.

Vytvoriť ukázkovú šablónu na dokumentáciu

- V rámci tejto úlohy bolo potrebné vytvoriť šablónu so všetkými typmi fontov a nadpisov z používanej metodiky.

Dokumentácia riadenia

- V rámci tejto úlohy bolo potrebné vytvoriť dokumentáciu riadenia.

Dokumentácia obsahu

- V rámci tejto úlohy bolo potrebné vytvoriť dokumentáciu obsahu.

Úpravy a doplnenie textov

- V rámci tejto úlohy bolo potrebné upraviť a doplniť texty dokumentácie riadenia a dokumentácie obsahu.

Vybrať najlepšie gamifikačné prvky a mechaniky

- V rámci tejto úlohy bolo potrebné vybrať najlepšie gamifikačné prvky a mechaniky pre náš projekt.

Navrhnuť scenáre

- V rámci tejto úlohy bolo potrebné navrhnuť scenáre, v ktorých sa dajú použiť gamifikačné prvky.

Vytvorenie rozhrania na vloženie súboru

- V rámci tejto úlohy bolo potrebné vytvoriť rozhranie na vkladanie súborov kvôli posielaniu.

Vytvorenie aplikácie na odoslanie súboru zo servera na autobus

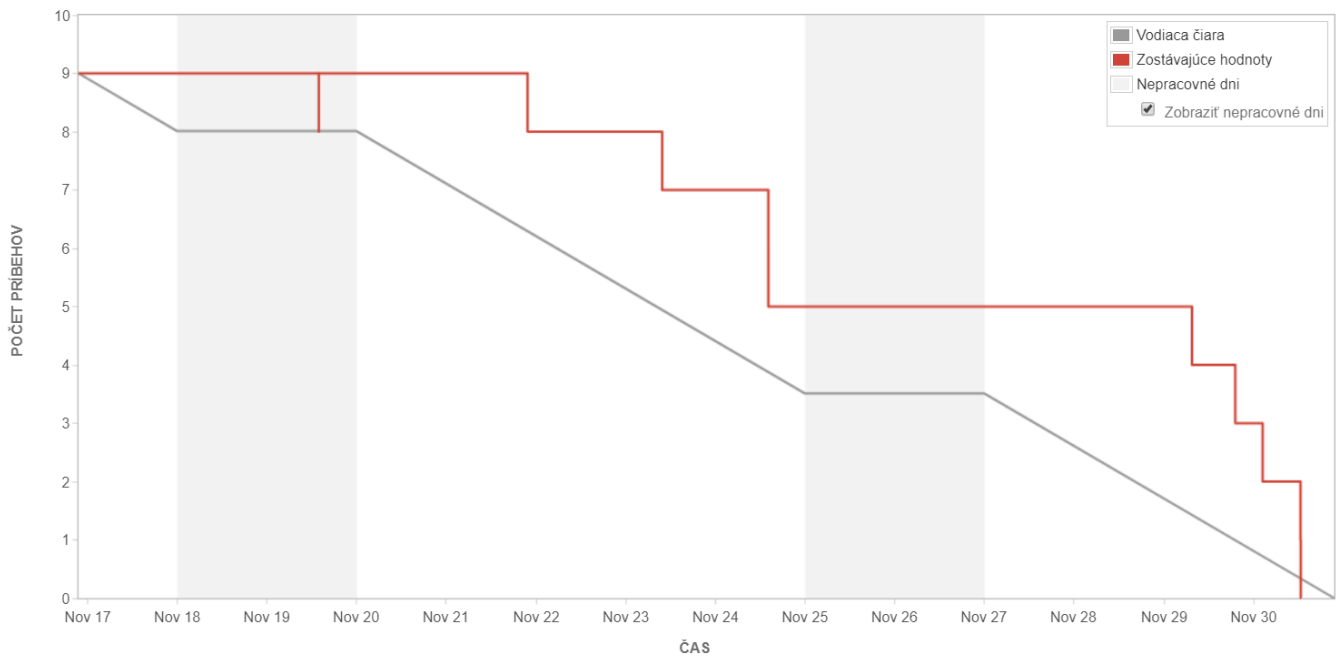
- V rámci tejto úlohy bolo potrebné vytvoriť funkciu, pomocou ktorej sa budú dať odosielať súbory medzi servermi.

4.4.1 Retrospektíva

Tabuľka 9. Retrospektíva šprintu

START	Pravidelnejšie stand up
KEEP	Kvalita práce na obrazovkách
STOP	Vynechávanie stretnutí

4.4.2 Burndown chart



Obrázok 4 zobrazuje Burndown chart štvrtého šprintu.

4.4.3 Zhodnotenie šprintu Ribera

Štvrtý šprint priniesol so sebou prvé odovzdávanie dokumentácie a výraznejšie pokroky vo webovej aplikácii aj na serveri Funtoro MOD. Z dôvodu veľkého množstva úloh a menších komplikácií sme sa rozhodli jednu úlohu zo šprintu vyradiť. Všetky ostatné úlohy boli úspešne vykonané.

4.5 5. Šprint

Zadanými úlohami v tomto šprinte bolo identifikovať sedadlo cestujúceho potrebné na vyhodnocovanie dotazníkov. Ďalšou úlohou bol zber údajov o klientovi vytvorením formulára, cez ktorý sa budú prenášať kontaktné údaje o klientovi medzi servermi. Ďalšími úlohami týkajúcich sa webovej aplikácie bolo vytvorenie správy firiem, vytvorenie formátu dotazníka, ktorý bude možné vytvoriť cez web. Ďalšou úlohou bolo spísanie finálnej verzie dokumentácie riadenia.

Tabuľka 10. Tabuľka s úlohami šprintu a ich pridelenie k členom tímu

User story	Úloha	Zodpovedná osoba
Odovzdať a dopracovať finálnu dokumentáciu	Spísať sumár zvyšných šprintov	Roderik Williger
	Napísať globálnu retrospektívu	Mária Dlhá
Vytvorenie správy firiem	Vytvoriť grupy a priradiť adminov	Erik Chodorčuk
	Určenie role userovi+ mazanie	
Identifikovať sedadlo cestujúceho	Zistiť spôsoby identifikácie sedadla od firmy Molpir	Adam Strásky
	Vytvoriť funkciu na identifikáciu sedadla	
Vytvorenie formátu dotazníka	Navrhnuť formát súboru	Yurii Hubar
	Navrhnuť štruktúru dotazníka	
	Vedieť cez web vytvoriť ukázkový formulár	Nadežda Juhásová
Zber údajov o klientovi	Vytvoriť formulár, pomocou ktorého bude možné sťahovať informácie len v prípade zadaných kontakt. údajov	Roderik Williger
	Spraviť prenos údajov medzi servermi	Adam Strásky
Small issues + sprint fixies	Vytvorenie role u pracovníka a zákazníka	Štefan Grivalský
	Pridanie možnosti vytvorenia a zmeny konta pracovníka	

Spísať sumár zvyšných šprintov

- V rámci tejto úlohy bolo potrebné doplniť dokumentáciu riadenia o zázname zo 4. a 5. šprintu.

Napísať globálnu retrospektívu

- V rámci tejto úlohy bolo potrebné napísať globálnu retrospektívu za zimný semester.

Vytvoriť grupy a priradiť adminov

- V rámci tejto úlohy bolo potrebné vytvoriť firmy v podobe grupy a dokázať vytvoriť adminov každej z firiem a priradiť ich k daným grupám. Každá grupa je jedna firma.

Určenie role userovi+ mazanie

- V rámci tejto úlohy bolo potrebné umožniť vytvoriť usera, ktorému vieme priradiť rolu admin a priradiť ho do našej grupy. Potrebné bolo taktiež vytvoriť lepšie mazanie firmy (grupy).

Zistiť spôsoby identifikácie sedadla od firmy Molpir

- V rámci tejto úlohy bolo potrebné zistiť informácie o možnostiach identifikácie sedadla v autobuse.

Vytvoriť funkciu na identifikáciu sedadla

- V rámci tejto úlohy bolo potrebné vytvoriť funkciu, pomocou ktorej vieme identifikovať sedadlo používateľa na základe predom zistených informácií z inej úlohy.

Navrhnuť formát súboru

- V rámci tejto úlohy bolo potrebné navrhnúť formát súboru, pomocou ktorého budeme prenášať dotazníky.

Navrhnuť štruktúru dotazníka

- V rámci tejto úlohy bolo potrebné navrhnúť štruktúru dotazníka, ktorý sa bude odosielať na server do autobusov.

Vedieť cez web vytvoriť ukázkový formulár

- V rámci tejto úlohy bolo potrebné umožniť cez web vytvárať ukázkový formulár.

Vytvoriť formulár, pomocou ktorého bude možné sťahovať informácie len v prípade zadaných kontaktných údajov

- V rámci tejto úlohy bolo potrebné vytvoriť formulár na obrazovkách, pomocou ktorého sa budú sťahovať kontaktné informácie o používateľovi. Bolo potrebné ošetriť prázdne a chybové vstupy, keďže chceme prenášať len vhodné údaje.

Spraviť prenos údajov medzi servermi

- V rámci tejto úlohy bolo potrebné zabezpečiť preposielanie kontaktných údajov stiahnutých pomocou formulára na server mimo autobus.

Vytvorenie role u pracovníka a zákazníka

- V rámci tejto úlohy bolo potrebné umožniť vytvárať role pracovníkov v autobusových firmách.

Pridanie možnosti vytvorenia a zmeny konta pracovníka

- V rámci tejto úlohy bolo potrebné umožniť autobusovým firmám meniť a vytvárať kontá pracovníkov.

4.6 Zhodnotenie šprintu

V čase písania tohto dokumentu ešte nebol uzatvorený 5 šprint. Šprint má koniec 14. decembra a z tohto dôvodu nie je možné pridať burndown chart a retrospektívu šprintu.

5 Globálna retrospektíva ZS

Počas zimného semestra členovia tímu postupovali podľa postupov, ktoré sa vykonávajú pri SCRUM vývoji produktu. Na začiatku každého šprintu boli ohodnotené príbehy a úlohy členmi tímu, a na základe náročnosti úloh boli úlohy rozdelené pomedzi členov tímu. Počas zimného semestra sme sa sústredili na vytvorenie prototypu aplikácie na prenos informácií medzi webovým serverom majiteľov flotíl a servermi v autobusoch.

Počas prvých troch šprintov sme sa sústredili na vytvorenie prototypu webovej stránky, ktorá bude neskôr slúžiť na správu zákazníkov, ako aj spravovanie dotazníkov, ktoré sa budú odosielať na server v autobusoch. Na webovej stránke je zavedené prihlasovanie pre admina firmy Molpir, s.r.o. a admina autobusovej flotily, ktorým aplikácia ponúka rozdielnu funkcionálnosť. Na webovej stránke je zavedená aj viacjazyčnosť stránky, ktorá umožní prepínanie stránok do ľubovoľného jazyka. Počas prvých troch šprintov sa nám podarilo umožniť interakciu používateľov obrazoviek v autobusoch so serverom v autobusoch. Interakcia bude slúžiť na získavanie odpovedí dotazníkov, na základe ktorých budú vyhodnocované dotazníky.

Počas ďalších dvoch šprintov sa nám podarilo rozšíriť funkcionálnosť webových stránok, avšak kvôli slabšej komunikácii medzi členmi tímu sa nám nepodarilo splniť všetky úlohy jedného šprintu, čím tento šprint bol považovaný za neúspešný. Po tejto vzniknutej situácii sme si uvedomili, že komunikácia medzi členmi tímu je dôležitá a snažili sme sa to napraviť. V týchto dvoch šprintoch sa nám tiež podarilo zaznamenávať informácie od používateľa, ako aj informácie o obrazovke, z ktorej dané informácie sú, ktoré sú potrebné pre budúce vyhodnocovanie dotazníkov.

Po každom ukončení šprintu bola vykonaná retrospektíva šprintu, v ktorej sa každý člen tímu vyjadril k pozitívom a negatívom daného šprintu. Na základe negatívnych pripomienok členov tímu o šprinte sme sa snažili tieto pripomienky v budúcom šprinte vylepšiť, čím by sme sa vyhli možným problémom počas šprintu. Pozitívne pripomienky sme sa snažili udržať aj v nasledujúcich šprintoch pre správny chod šprintov.

V budúcich šprintoch sa budeme snažiť splniť všetky požadované úlohy a šprinty ukončiť úspechom. Pre lepšiu komunikáciu medzi členmi tímu sa budeme stretávať aj pomimo školy, čím zabezpečíme aj lepšiu súdržnosť tímu.

7 Používané metodiky

Metodiky sú nevyhnutnou súčasťou pri práci na tímovom projekte. Pri kolaboratívnej práci treba zaviesť spoločné postupy a pravidlá, aby bola práca jednotná, a aby si všetci členovia mohli ľahko vzájomne kontrolovať úlohy. Metodiky sa nachádzajú v prílohe.

7.1 Metodika písania zdrojového kódu

Metodika sa zaoberá písaním zdrojového kódu v jazykoch Javascript a HTML. Má za úlohu navigovať programátora pri písaní zdrojového kódu v týchto jazykoch. Usmerňuje ho v syntaxe a určuje základné pravidlá, ktoré udržiavajú čitateľnosť zdrojového kódu pre všetkých členov tímu.

7.2 Metodika pre Code Review

Metodika pre Code Review sa zaoberá postupmi a pravidlami ako kontrolovať zdrojový kód. Kontrolór, ktorému je pridelená úloha sa musí podľa týchto pravidiel riadiť, aby bolo overovanie kódu jednotné a aby sa kvalita kódu udržiavala na približne rovnakej úrovni.

7.3 Metodika pre Definition of Done

Metodika pre Definition of Done predpisuje základným typom úloh, kedy sa považujú za splnené. Je v nej zovšeobecnená schéma troch typov úloh, ktorá sa môže namapovať následne na konkrétne pridelené úlohy. Daná úloha sa následne pokladá za hotovú ak spĺňa podmienky spísané v tejto metodike.

7.4 Metodika písania dokumentácie

Metodika písania dokumentácie predpisuje pravidlá pri písaní všetkých dokumentov. V metodike sa popisuje aký štýl písma, veľkosti odsekov a veľkosti písma majú byť použité v dokumente. V metodike je taktiež popísaný štýl tabuliek.

8 Prílohy

- Príloha A: Metodiky
- Príloha B: Export evidencie úloh

Príloha A

Táto príloha obsahuje metodiky pre :

- Code Review.
- Definition of Done.
- Písanie dokumentácie.
- Písanie zdrojového kódu v Javascripte a HTML.

A.1 Metodika pre Code Review

Hlavným cieľom metodiky pre Code Review je zosumarizovanie pravidiel a vytvorenie základného postupu potrebného pre posudzovanie kódu (angl. code review). Pre získanie čo najlepšej podoby kódu, ktorá nie je problematická a vyhovuje podmienkam projektu, je vhodné aj po otestovaní kódu vykonať jeho posudzovanie/odkontrolovanie.

Roly vývojárov

Na vytváraní zdrojového kódu sa podieľajú viacerí vývojári, preto je potrebné rozdeliť ich do nasledujúcich rolí:

- Autor kódu
 - autorom kódu je osoba, ktorá sa aktívne podieľala na napísaní/vytváraní zdrojového kódu
- Posudzovateľ kódu
 - posudzovateľom kódu je osoba, ktorá je zodpovedná za skontrolovanie zdrojového kódu napísaním poznámok k danému kódu, ktoré sú následne odoslané autorovi kódu

Pravidlá pri Code Review

- Posudzovateľom by nemala byť osoba, ktorá sa aktívne podieľala na vytváraní zdrojového kódu.
- Posudzovateľom by nemala byť osoba, ktorá nemá žiadne vedomosti o danej problematike.
- Posudzovateľ priamo neopravuje chyby v kóde, ale ich detekuje, prípadne navrhne riešenie problému.
- Posudzovateľ kontroluje zdrojový kód až po jeho vytvorení.

Odporúčané:

- Autor kódu s posudzovateľom by sa spoločne mali podieľať na vykonávaní code review pre lepšie skonzultovanie jednotlivých zmien.

Priebeh Code Review

Pre správny postup posudzovania kódu a zachovanie jeho konzistentnosti je potrebné zadefinovať základný postup jeho vykonávania.

Postup vykonávania Code Review

- **Odobzdanie kódu**
Autor kódu odovzdá kód s riešením úlohy do príslušnej vetvy repozitára v GitLab-e.
- **Vytvorenie *merge request***
Po odovzdaní kódu do repozitára autor vytvorí *merge request*, pričom sa označia vetvy (z ktorej a do ktorej) má byť *merge request* vykonaný. Autor kódu označí aj osobu (posudzovateľ'a), ktorá nad kódom bude vykonávať code review. Posudzovateľ'ovi môže byť odoslaná aj súkromná správa v Slack-u na pripomenutie.
- **Skontrolovanie zdrojového kódu**
Posudzovateľ skontroluje dané riešenie a vyhodnotí kvalitu riešenia, pričom sa zameria na splnenie základných úloh, kvalitu kódu atď.

- **Vyhodnotenie code review**
Posudzovateľ vyhodnotí posudzovanie kódu. V prípade objavenia chýb alebo navrhnutia lepšieho riešenia danej úlohy poskytne autorovi spätnú väzbu.
- **Poskytnutie spätnej väzby**
V prípade, že má posudzovateľ nejaké pripomienky k danému riešeniu, pridá komentáre na dané miesta, ku ktorým mal pripomienky, alebo oznámi autorovi kódu tieto pripomienky súkromnými správami v Slack-u.
- **Úprava kódu**
Autor na základe pripomienok upraví riešenie danej úlohy, aby vyhovovali podmienkam posudzovateľa.
- **Akceptovanie *merge requestu***
V prípade, že posudzovateľ nemá žiadne pripomienky a kód s riešením úlohy považuje za vyhovujúci, akceptuje daný *merge request*.

A.2 Metodika pre Definition od Done

Cieľom metodiky pre Definition of Done (skr. DoD) je zosumarizovanie podmienok, kedy je špecifická úloha ukončená. Definition of Done predstavuje jednoduchý zoznam aktivít (písanie kódu, celkové testovanie, integračné testovanie, atď.), ktoré pridávajú overiteľnú hodnotu produktu. DoD určuje, aké aspekty práce je potrebné vykonať, aby bolo možné user story/šprint/release považovať za ukončený. DoD charakterizuje a smeruje prácu v tíme.

Metodika DoD

Metodika je určená všetkým členom tímu, ktorí pracujú na pridelených úlohách:

- **Správca JIRY**
Zodpovedá za vytvorenie DoD na základe dohodnutého obsahu úlohy, typu úlohy a všeobecných pravidiel DoD. DoD sa vytvára pri vytváraní samotnej úlohy.
- **Člen tímu**
Zodpovedá za splnenie dohodnutého a prideleného DoD (k úlohe, za ktorú je zodpovedný). Člen tímu musí byť oboznámený s DoD.
- **Product owner**
Kontroluje progres plnenia úloh porovnávaním dohodnutého DoD a výstupu od člena tímu, ktorý je zodpovedný za konkrétnu úlohu.

Navrhnuté rozdelenia DoD

Rozdelenie podľa typu úlohy:

1. Analytické úlohy
2. Návrhové úlohy
3. Vývojové úlohy

Rozdelenie DoD podľa úrovni progresu plnenia:

1. User story
2. Sprint
3. Release

Rozdelenie DoD podľa kategórií:

1. Vývoj
2. Testovanie
3. Dokumentácia
4. Manažment
5. Ostatné

Analytické úlohy

- Je naštudovaná problematika problému a spísané kľúčové body analýzy.
- V prípade porovnávania viacerých technológií alebo postupov je vypracovaná tabuľka a záver.
- Je vytvorený dokument danej problematiky s dodržanými postupmi pre tvorbu dokumentov.
- Dokument je zdieľaný medzi ostatných členov tímu.

Návrhové úlohy

- Je navrhnutý a odsúhlasený model riešeného systému alebo podsystému (minimálne tými členmi tímu, ktorí sa podieľajú na problematike).
- Je zdokumentovaný model systému s dodržanými postupmi pre tvorbu dokumentov.
- Dokument je zdieľaný medzi ostatnými členmi tímu.

Vývojové úlohy

1. User story

a. Vývoj

- Vývoj softvéru bol z hľadiska funkcionality ukončený.
- Doplnená časť systému je uložená v súkromnom repozitári na GitLab-e.
- Doplnený kód/skript bol skontrolovaný ďalším vývojárom (code review).
- Vytvorený kód/skript dodržiava metodiku pre zdrojový kód.

b. Testovanie

- Funkcionalita softvérovej súčiastky je otestovaná pomocou jednotkových testov alebo manuálnym testovaním.

c. Dokumentácia

- Napísaný kód/skript je dostatočne zdokumentovaný na porozumenie funkčnosti vyvinutej softvérovej súčiastky.
- Dokumentácia kódu dodržiava metodiku pre písanie dokumentácie.

d. Manažment

- Prípadné chybové situácie a neriešiteľné prekážky sú zdokumentované a predložené vlastníkovi produktu.

2. Sprint

a. Vývoj

- Vývojové vetvy sú zlúčené do hlavnej vetvy v GitLab-e.

b. Testovanie

- Pridaná funkcionality spĺňa akceptačné kritériá (definované vlastníkom produktu).

c. Dokumentácia

- Projektová dokumentácia je doplnená o časti, ktoré pokrývajú vyvinuté softvérové súčiastky (vrátane modelov, diagramov a ich opis).
- Je doplnená dokumentácia k riadeniu projektu.

d. Manažment

- Vyvinutá časť systému je úspešne skompilovaná a nasadená na serveri.
- Bola vykonaná retrospektíva šprintu.
- Bola doplnená tímová webová stránka o zápisky zo stretnutí.

3. Release

a. Vývoj

- Nasadenie systému.

b. Testovanie

- Verzia produktu spĺňa požiadavky zákazníka.

c. Dokumentácia

- Odovzdaná dokumentácia (riadenia aj projektu).
- Doplnený web o dokumentácie.

A.3 Metodika pre tvorbu dokumentácie

Cieľom metodiky pre písanie dokumentácie projektu je zosumarizovanie základných pravidiel pre písanie dokumentov. Vďaka vopred daných pravidiel bude písanie dokumentov všetkými členmi tímu jednotné, čo umožní ľahšie spracovanie týchto dokumentov do jedného celkového dokumentu.

Účel dokumentu

Tento dokument popisuje spôsoby tvorby dokumentácie a uvádza štruktúru a štýly, ktoré majú byť použité pri tvorbe dokumentácie projektu. Dokument sa využije pri tvorbe všetkých dokumentov týkajúcich sa dokumentácií projektu v rámci tímového projektu.

Členovia tímu v rámci tvorby dokumentácie

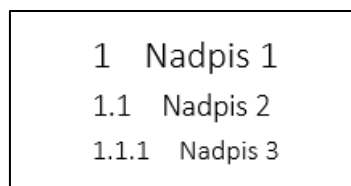
Daná metodika je určená všetkým členom tímu, ktorí sa aktívne zúčastňujú na tvorbe dokumentácie, pre udržiavanie jednotných a konzistentných štruktúr dokumentov. Členovia tímu v rámci tvorby dokumentácie:

- Správca dokumentácie
 - Zpracováva dodané dokumenty a dohliada na obsah celkovej dokumentácie.
- Člen tímu
 - Zabezpečuje tvorbu textov do dokumentácie na základe danej metodiky.

Metodika pre tvorbu dokumentácie projektu

Formát všetkých dokumentov počas ich editácie je DOCX, ktorý sa pri odovzdaní môže meniť na formát PDF. V dokumentácii bude použité písmo *Times New Roman* s veľkosťou 11. Text v dokumentácii bude zarovnaný podľa okrajov.

V dokumente na písanie textu sa používa štýl „Normálny“, v ktorom sú definované potrebné parametre. V dokumentácii sa používajú nadpisy až do štvrtej úrovne nadpisov. Pre nadpisy sa používajú štýly „Nadpis1“, „Nadpis2“, „Nadpis3“, ktoré umožňujú jednoduchú synchronizáciu medzi viacerými tvorcami dokumentácie. Pre ďalšie menšie nadpisy v texte možno použiť štýl „Nadpis4“.



Obrázok 4. Názorný obrázok zobrazujúci úrovne nadpisov

Pri tvorbe tabuliek sa bude používať štýl písma „Normálny“ s **tučným písmom** pre popisné bunky tabuľky. Pre bunky tabuľky, ktoré tvoria jej obsah, sa bude používať štýl „Normálny“. Popisné bunky zároveň budú mať nastavené tieňovanie na svetlosivú farbu, ktorá zabezpečí ich odlíšenie od ostatných buniek tabuľky. Na popis tabuľky bude využitý štýl „Popis“ a popis bude umiestnený nad danou tabuľkou, viď **Chyba! Odkaz na záložku nie je platný.**

Tabuľka 11. Názorný príklad tabuľky v dokumente

Bunka pre popis	Bunka pre popis
Bunka pre obsah	Bunka pre obsah

Odrážky v texte budú mať nasledovný tvar:

- prvá úroveň odrážok
 - druhá úroveň odrážok

Číslovanie v texte bude mať nasledovný tvar:

1. prvá úroveň číslovania
 - a. druhá úroveň číslovania

Na popis k vloženým obrázkom bude využitý štýl „*Popis*“ a popis sa bude nachádzať pod daným obrázkom, vid' Obrázok 4.

Pri tvorbe dokumentácie je potrebné zaznamenávať použité skratky v dokumentácii. Tieto skratky budú zaznamenávané v časti „Použité skratky“.

V prípade citácií a pridávania zdrojov sa vo všetkých formálnych dokumentoch používa norma ISO 690.

A.4 Metodika pre písanie zdrojového kódu

Zámer dokumentu

Zámerom tejto časti dokumentu je určiť základné konvencie pre tvorbu kódu v programovacích jazykoch použitých v našom projekte.

Konvencie tvorby kódu v programovacom jazyku HTML

V tejto časti dokumentu sú definované konvencie pre tvorbu kódu v programovacom jazyku HTML5. Tieto konvencie sú prebraté z dokumentácie o konvenciách kódu pre programovací jazyk HTML5 zo stránky w3school.com.

Názov zdrojového súboru

Používame len malé písmená. Snažiť sa o čo najkratší a zároveň pochopiteľný názov. Názov začína vždy písmenom. Názvy v HTML5 sú „case sensitive“.

Typ dokumentu

Každý dokument HTML musí obsahovať na prvom riadku typ dokumentu veľkými písmenami.

Príklad:

```
<!DOCTYPE html>
```

Názvy elementov

Názvy jednotlivých elementov sú zložené len z malých písmen.

Príklad:

```
<section>
  <p>This is a paragraph.</p>
</section>
```

Zatváranie elementov

V HTML5 nie je potrebné zatvárať všetky elementy, no našim odporúčaním je uzatvárať každý element. Prázdny element uzatvárame taktiež.

Príklad:

```
<section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>

<meta charset="utf-8" /> //prázdny element
```

Názvy atribútov

Názvy jednotlivých atribútov sú zložené len z malých písmen. Každý názov atribútov zaquótujeme úvodzovkami.

Príklad:

```
<div class="menu">
```

Medzery

HTML5 umožňuje medzery okolo znakov „=“ . Napriek tomu je lepšie čitateľná konvencia, kde pri tomto znaku medzery nie sú.

Príklad:

```
<link rel="stylesheet" href="styles.css" > //dobre
```

```
<link rel = "stylesheet" href = "styles.css" > //zle
```

Prázdné riadky

Pre lepšiu čitateľnosť pridávame prázdne riadky na oddelenie veľkých alebo logických blokov kódu. Nepoužívajte nepotrebné prázdne riadky a odsadenie (tabulátor).

Príklad:

```
<body>
  <h1>Famous Cities</h1>
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan, the center of the
  Greater Tokyo Area, and the most populous metropolitan
  area in the world. It is the seat of the Japanese
  government and the Imperial Palace, and the home of the
  Japanese Imperial Family.</p>
</body>
```

Komentáre

Je potrebné písať komentáre pri všetkých väčších blokoch pre lepšiu orientáciu v kóde. Napríklad pri začiatku a konci elementov.

Syntax:

```
<!-- This is a comment -->
```

```
<!--
```

```
This is a long comment example. This is a long
comment example.
```

```
This is a long comment example. This is a long
comment example.
```

```
-->
```

Validácia

Možnosť validácie kódu na <https://validator.w3.org/>

Konvencie tvorby kódu v programovacom jazyku JavaScript

V tejto časti dokumentu sú definované konvencie pre tvorbu kódu v programovacom jazyku JavaScript. Tieto konvencie sú prebraté z dokumentácie o konvenciách kódu pre programovací jazyk JavaScript zo stránky w3school.com.

Názvy premenných

Každý názov premenných a funkcií začína písmenom. Názov je zložený z malých písmen. V prípade v viac slovných názvov premenných, jednotlivé slová neoddeľovať, ale použiť veľké začiatkové písmeno druhého a nasledujúceho slova názvu („camelCase“).

Globálne premenné využiť len veľké písmená (UPPERCASE). Konštanty ako napr. PI využívať UPPERCASE.

Príklad:

```
firstName = "John";
lastName = "Doe";

price = 19.90;
tax = 0.20;

fullPrice = price + (price * tax);

nameOfFunction = function() { }
```

Medzery

Vždy používať medzery okolo operátorov (= + - * /) a za čiarkami.

```
var x = y + z;
var values = ["Volvo", "Saab", "Fiat"];
```

Pravidlá

- Zložené zátvorky: Otvárajúca zátvorka vždy na konci prvého riadku.
- Použite medzeru pred otvárajúcou zátvorkou.
- Uzatvárajúcu zátvorku umiestniť na nový riadok bez medzier pred ňou (automatické dopĺňanie).

Komentáre

Písať komentáre nad každú funkciu. V prípade zložitejších operácií komentovať daný krok hneď v danom riadku (oddeliť tabulátorom). Na začiatku súboru komplexný komentár o funkcionalite, update-ov, zmien.

Syntax:

```
//toto je komentár

/*
  Toto je dlhy
  komentar
*/
```

Konvencie tvorby kódu v programovacom jazyku CSS

V tejto časti dokumentu sú definované konvencie pre tvorbu kódu v programovacom jazyku CSS. Tieto konvencie sú prebraté z dokumentácie o konvenciách kódu pre programovací jazyk CSS zo stránky <https://cssguidelin.es/>.

- dodržiavať organizáciu kódu, držať definície patriace k jednému prvku (príp. skupine prvkov) pri sebe
- používať čo možno najpresnejšie selektory
- oddeľovať definície patriace k rozdielnym skupinám prvkov (príp. k iným stránkam) prostredníctvom komentárov alebo prázdneho riadku
- začiatočnú zloženú zátvorku oddeliť od názvu medzerou, zvyšok kódu písať na nový riadok
- končiacu zloženú zátvorku písať na nový riadok
- názvy tried písať malými písmenami
- dvojslovné názvy oddeľovať pomlčkou
- komentáre: využívať syntax `/* comment here */`

Konvencie tvorby kódu v programovacom jazyku Python

V tejto časti dokumentu sú definované konvencie pre tvorbu kódu v programovacom jazyku Python. Tieto základné konvencie sú prebraté z dokumentácie o konvenciách kódu pre programovací jazyk Python zo stránky <https://www.python.org/>.

Konvencia názvov

- názov súboru (modulu) má byť krátky, výstižný, len z malých písmen a bez podtržníkov (myfile.py)
- názov tried využíva konvenciu „CapWords“ (MyClass)
- názvy metód a premenných využívať lowercase s podtržníkom (my_variable)

Organizovanie Importov

- vždy na vrchu súboru pred globálnymi premennými a modulmi

- každý import na novom riadku
 - nie: import sys, os
 - áno: import sys
import os
- môžu byť triedené do skupín (štandardné knižničné importy, import hlavných balíkov, import špecifických balíkov)

Zarážky a dĺžka riadku

- v prípade jednotného IDE používať tabulátor, inak 2 medzery
- nepoužívať viac než 5 levelov zarážiek (tabulátorov)
- maximálny počet znakov na riadok je 72
- na prerušenie dlhého riadka možno použiť „\“

Prázdne riadky

- vynechať jeden riadok medzi funkciami v triede
- skupinu súvisiacich funkcií oddeliť riadkom navyše alebo komentárom
- využívať prázdny riadok v rámci funkcii pri oddelení logických sekcií

Medzery

- okolo operátorov používať medzery, vždy len jednu
- nepoužívať medzery v zátvorkách a pred nimi (napr. polia)
- medzera vždy za čiarkou

Komentáre

- odsadené na rovnakej úrovni ako kód
- každý riadok blokového komentára začína znakom # a jednou medzerou
- jednotlivé odseky v blokovom komentári sú oddelené jedným znakom # na novom riadku
- jednoduchý komentár je oddelený od kódu 2 medzerami

Zdroje

HTML dostupné z: https://www.w3schools.com/html/html5_syntax.asp

JavaScript dostupné z: https://www.w3schools.com/js/js_conventions.asp

CSS dostupné z: <https://cssguidelin.es/>

Python dostupné z: <https://www.python.org/dev/peps/pep-0008/>

Príloha B

Export evidencie úloh:

T	Kľúč	Sumár	Priradený	Zadávateľ	P	Stav	Rozhodnutie	Vytvorený	Aktualizovaný	Termin
🔗	MOD-94	MOD-88 / Napísať globálnu retrospektívu	Maria Dlha	Roderik Williger	↑	HOTOVO	Hotovo	11/dec/17	12/dec/17	
🔗	MOD-93	MOD-88 / Spísať súmar zvyšných šprintov	Roderik Williger	Roderik Williger	↑	HOTOVO	Hotovo	11/dec/17	12/dec/17	
🔗	MOD-92	MOD-89 / vedieť cez web vytvoriť ukazkovy formular	Nada Juhasova	Peter Zabka	↑	ROZPRACOVANE	Nevyriešené	30/nov/17	11/dec/17	
🔗	MOD-91	MOD-89 / navrhnuť strukturu dotazníka	Yurii Hubar	Peter Zabka	↑	HOTOVO	Hotovo	30/nov/17	11/dec/17	
🔗	MOD-90	MOD-89 / navrhnuť format suboru	Yurii Hubar	Peter Zabka	↑	HOTOVO	Hotovo	30/nov/17	07/dec/17	
🟢	MOD-89	Vytvorenie formátu dokazníka	Yurii Hubar	Peter Zabka	↑	ÚLOHY	Nevyriešené	30/nov/17	30/nov/17	
🟢	MOD-88	Odozvať finalnu dokumentáciu	Roderik Williger	Peter Zabka	↑	HOTOVO	Hotovo	30/nov/17	12/dec/17	
🟢	MOD-87	Small issues + sprint fixes	Stefan Grivalsky	Peter Zabka	↑	ÚLOHY	Nevyriešené	30/nov/17	30/nov/17	
🔗	MOD-86	MOD-59 / Zobrazenie pouzivatelov skupiny	Erik Chodorouk	Peter Zabka	↑	HOTOVO	Hotovo	23/nov/17	29/nov/17	
🔗	MOD-85	MOD-87 / Pridanie možnosti vytvorenia a zmeny konta pracovníka [8]	Stefan Grivalsky	Peter Zabka	↑	ROZPRACOVANE	Nevyriešené	21/nov/17	30/nov/17	
🔗	MOD-84	MOD-87 / Vytvorenie role "pracovník u zákazníka" [3]	Stefan Grivalsky	Peter Zabka	↑	ROZPRACOVANE	Nevyriešené	21/nov/17	07/dec/17	
🔗	MOD-83	MOD-64 / Navrhnuť scenáre	Yurii Hubar	Yurii Hubar	↑	HOTOVO	Hotovo	21/nov/17	29/nov/17	
🔗	MOD-82	MOD-64 / Vybrať najlepšie gamifikačné prvky a mechaniky	Yurii Hubar	Yurii Hubar	↑	HOTOVO	Hotovo	21/nov/17	27/nov/17	
🔗	MOD-81	MOD-55 / Vytvoriť logger [2]	Adam Strasky	Adam Strasky	↑	HOTOVO	Hotovo	20/nov/17	24/nov/17	
🔗	MOD-80	MOD-56 / Zapracovanie logovania do existujuceho systemu [2]	Peter Zabka	Peter Zabka	↑	HOTOVO	Hotovo	20/nov/17	30/nov/17	
🔗	MOD-79	MOD-56 / Vytvorenie funkcie na zaznamenanie do logu [1]	Peter Zabka	Peter Zabka	↑	HOTOVO	Hotovo	20/nov/17	30/nov/17	
🔗	MOD-78	MOD-56 / Navrhnuť model pristupovych prav [2]	Peter Zabka	Peter Zabka	↑	HOTOVO	Hotovo	20/nov/17	30/nov/17	
🔗	MOD-77	MOD-56 / Navrh formatu [3]	Peter Zabka	Peter Zabka	↑	HOTOVO	Hotovo	20/nov/17	30/nov/17	
🔗	MOD-76	MOD-63 / Úpravy a doplnenie textov	Maria Dlha	Roderik Williger	↑	HOTOVO	Hotovo	20/nov/17	21/nov/17	
🔗	MOD-75	MOD-63 / Dokumentácia obsahu	Roderik Williger	Roderik Williger	↑	HOTOVO	Hotovo	20/nov/17	21/nov/17	
🔗	MOD-74	MOD-63 / Dokumentácia riadenia	Roderik Williger	Roderik Williger	↑	HOTOVO	Hotovo	20/nov/17	21/nov/17	
🔗	MOD-73	MOD-55 / Zapracovanie logov do existujuceho kodu [1]	Adam Strasky	Adam Strasky	↑	HOTOVO	Hotovo	19/nov/17	24/nov/17	
🔗	MOD-72	MOD-55 / Návrh formatu [1]	Adam Strasky	Adam Strasky	↑	HOTOVO	Hotovo	19/nov/17	24/nov/17	
🔗	MOD-71	MOD-65 / Vytvorenie aplikácie na odoslanie suboru zo servera na autobus [3]	Adam Strasky	Adam Strasky	↑	HOTOVO	Hotovo	19/nov/17	24/nov/17	
🔗	MOD-70	MOD-65 / Vytvorenie rozhrania na vloženie súboru [1]	Adam Strasky	Adam Strasky	↑	HOTOVO	Hotovo	19/nov/17	24/nov/17	
🔗	MOD-69	MOD-62 / Vytvoriť ukazkovu sablonu na dokumentáciu [1]	Maria Dlha	Peter Zabka	↑	HOTOVO	Hotovo	16/nov/17	21/nov/17	

T	Kľúč	Sumár	Priradený	Zadávateľ	P	Stav	Rozhodnutie	Vytvorený	Aktualizované	Termín
	MOD-68	MOD-62 / Zriadenie jednotneho miesta pre ukladanie dokumentácie[1]	Nada Juhasova	Peter Zabka	↑	HOTOVO	Hotovo	16/nov/17	20/nov/17	
	MOD-33	Sprava zakaznikov vo webovej aplikacii	Peter Zabka	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	16/nov/17	
	MOD-34	Odoslanie textu z obrazovky na server a zo servra posunut dalej	Adam Strasky	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	16/nov/17	
	MOD-36	Spravit metodiku na CR a DoD a Dokumentaciu	Maria Dlha	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	16/nov/17	
	MOD-39	MOD-33 / Vytvaranie, zmena a zmazanie pouzivatel'skych uctov [5]	Erik Chodorcuk	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-35	Zistit vhodne gamifikacne prvky	Yurii Hubar	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-48	MOD-35 / Spracovanie vysledkov [3]	Yurii Hubar	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-40	MOD-33 / Zobrazenie specifickoho obsahu pre prihlaseneho usera [3]	Peter Zabka	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-51	MOD-36 / Metodika na dokumentaciu [2]	Maria Dlha	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-50	MOD-36 / Metodika na DoD [2]	Maria Dlha	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-49	MOD-36 / Metodika na CR [2]	Maria Dlha	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-47	MOD-35 / Spracovanie analyzy (konzultacia) [3]	Yurii Hubar	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-44	MOD-34 / Vytvorenie aplikacie, ktora odosle text z DB na server mimo autobus [8]	Adam Strasky	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	15/nov/17	
	MOD-37	MOD-33 / Vytvorit a napojit Postgre DB [3]	Stefan Grivalsky	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	14/nov/17	
	MOD-41	MOD-34 / Vytvorenie editacneho pola + zobrazenie klavesnice [2]	Nada Juhasova	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	13/nov/17	
	MOD-43	MOD-34 / Ulozenie textu z editacneho pola do DB [3]	Roderik Williger	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	13/nov/17	
	MOD-46	MOD-35 / Analyza [2]	Yurii Hubar	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	12/nov/17	
	MOD-45	MOD-34 / Spracovanie (zorazenie) textu, ktory pride z autobusu [3]	Adam Strasky	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	09/nov/17	
	MOD-18	Vybrať pokuty od Števa	Adam Strasky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	09/nov/17	
	MOD-17	Vyhľadať vývojové prostredia - technológie	Erik Chodorcuk	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	09/nov/17	
<input checked="" type="checkbox"/>	MOD-4	Spraviť metodiku	<i>Nepripravené</i>	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	09/nov/17	
	MOD-42	MOD-34 / Vytvorenie databazy v autobuse [2]	Roderik Williger	Peter Zabka	↑	HOTOVO	Hotovo	02/nov/17	08/nov/17	
	MOD-5	Viacjazyčná stránka s prihlásením	<i>Nepripravené</i>	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	02/nov/17	
	MOD-9	MOD-5 / Vytvorenie prihlásenia	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	02/nov/17	
	MOD-8	MOD-5 / Podporovať zmenu jazyka	Stefan Grivalsky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	01/nov/17	
	MOD-12	Spraviť metodiku	<i>Nepripravené</i>	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	01/nov/17	
	MOD-14	MOD-12 / Metodika na pull requesty	Erik Chodorcuk	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	01/nov/17	
	MOD-13	MOD-12 / Metodika na kód	Erik Chodorcuk	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	01/nov/17	
	MOD-6	MOD-5 / Vybrať rozumnú technológiu na web + DB	Roderik Williger	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	31/okt/17	

T	Kľúč	Sumár	Priradený	Zadávatel	P	Stav	Rozhodnutie	Vytvorený	Aktualizované	Termín
	MOD-7	MOD-5 / Nakódiť Hello World statickú stránku	Roderik Williger	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	31/okt/17	
	MOD-1	Jednoduchá zmena na orazovke	<i>Nepriradené</i>	Peter Zabka	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-11	MOD-1 / Urobiť nejakú zmenu	Adam Strasky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-10	MOD-1 / Zohnať čo najviac informácií	Adam Strasky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-2	Vytvoríť návrh tričiek	Maria Dlha	Nada Juhasova	↓	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-15	Vytvorenie scrum projektu v Jire	<i>Nepriradené</i>	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-20	MOD-15 / Vytvorenie projektu	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-19	MOD-15 / Premigrovať úlohy z trela	Nada Juhasova	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-30	Nainštalovať server	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-31	MOD-30 / Nainštalovať PHP a MySQL	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-26	Získať bezplatné konto na JIRU	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-27	Analyzovať Scrum	<i>Nepriradené</i>	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-28	Vytvoríť trello	Stefan Grivalsky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-29	Nainštalovať wordpress a vytvoríť stránku	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-22	Dopyt na server s odpoveďou	Adam Strasky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-23	Vytvoríť súkromný repozitár gitLab	Maria Dlha	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-24	Vytvoríť pokuty	Adam Strasky	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-25	Vytvoríť plagát	Erik Chodorcuk	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	
	MOD-16	Rozbehať Jiru	Peter Zabka	Nada Juhasova	↑	HOTOVO	Hotovo	30/okt/17	30/okt/17	

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 19 Bratislava 4

Projektová dokumentácia produktu

Tím č. 8

MOD-eli on Demand

Akademický rok: 2017/2018
Predmet: Tímový projekt

Členovia tímu: Bc. Mária Dlhá
Bc. Štefan Grivalský
Ing. Yurii Hubar
Bc. Erik Chodorčuk
Bc. Nadežda Juhásová
Bc. Adam Strásky
Bc. Roderik Williger
Bc. Peter Žabka

Vedúci tímu: Ing. Peter Pišteck, PhD.
Vlastník produktu: Molpir, s.r.o.

Obsah

1	Úvod	3
2	Globálne ciele pre ZS	4
2.1	Brainstorming ku projektu Funtoro MOD.....	4
3	Celkový pohľad na systém	6
4	Moduly systému	8
4.1	SHD obrazovky a server v autobusoch	8
4.1.1	Spracovanie dát z obrazovky	8
4.1.2	Spracovanie textu z databázy na pozadí	8
4.1.3	Komunikácia servera v autobuse so serverom mimo autobusu.....	9
4.2	Webová aplikácia	9
4.2.1	Autentifikácia používateľa	9
4.2.2	Viacjazyčnosť	10
5	Prílohy	11

1 Úvod

Predmetom tohto dokumentu je technická dokumentácia k projektu po prvých troch šprintoch. Cieľom dokumentu je priblíženie celkovej architektúry a technickej časti projektu. V dokumente sú opísané globálne ciele, celkový pohľad na systém a jeho jednotlivé moduly.

Moduly systému sa delia na dve časti. Prvou časťou je webová aplikácia vyvíjaná pre majiteľov autobusových flotíl, ktorá má slúžiť aj pre Molpir, s.r.o ako majiteľa produktu. Molpir bude môcť pomocou webovej aplikácie spravovať zákazníkov, ktorých predstavujú majitelia autobusových flotíl. Tým bude zase aplikácia slúžiť na spravovanie obsahu na ich serveroch v autobusoch. Pomocou webovej aplikácie budú môcť zasielať dáta na autobusový server, a taktiež spracovávať dáta, ktoré im z autobusov budú zaslané.

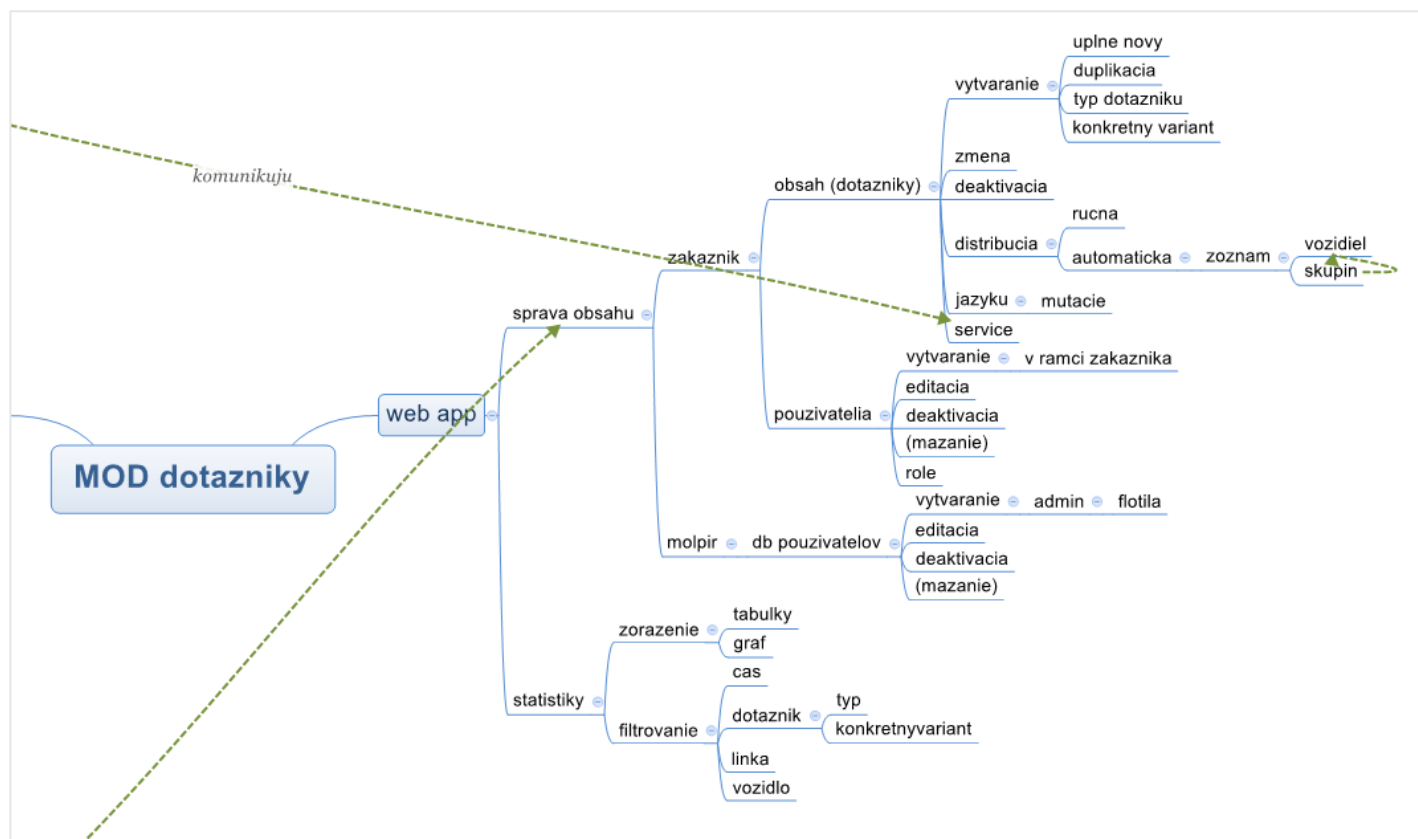
Druhou časťou je vyvíjaná funkcionálna na Funtoro MOD server, ktorý napája nové SHD obrazovky do autobusov. Keďže SHD hardvér je novinka a nie je k nemu žiadna dôstojná dokumentácia a ani funkcionálna, vyvíjame funkcionálnu od začiatku a aj architektúra je analyzovaná hlavne naším tímom.

2 Globálne ciele pre ZS

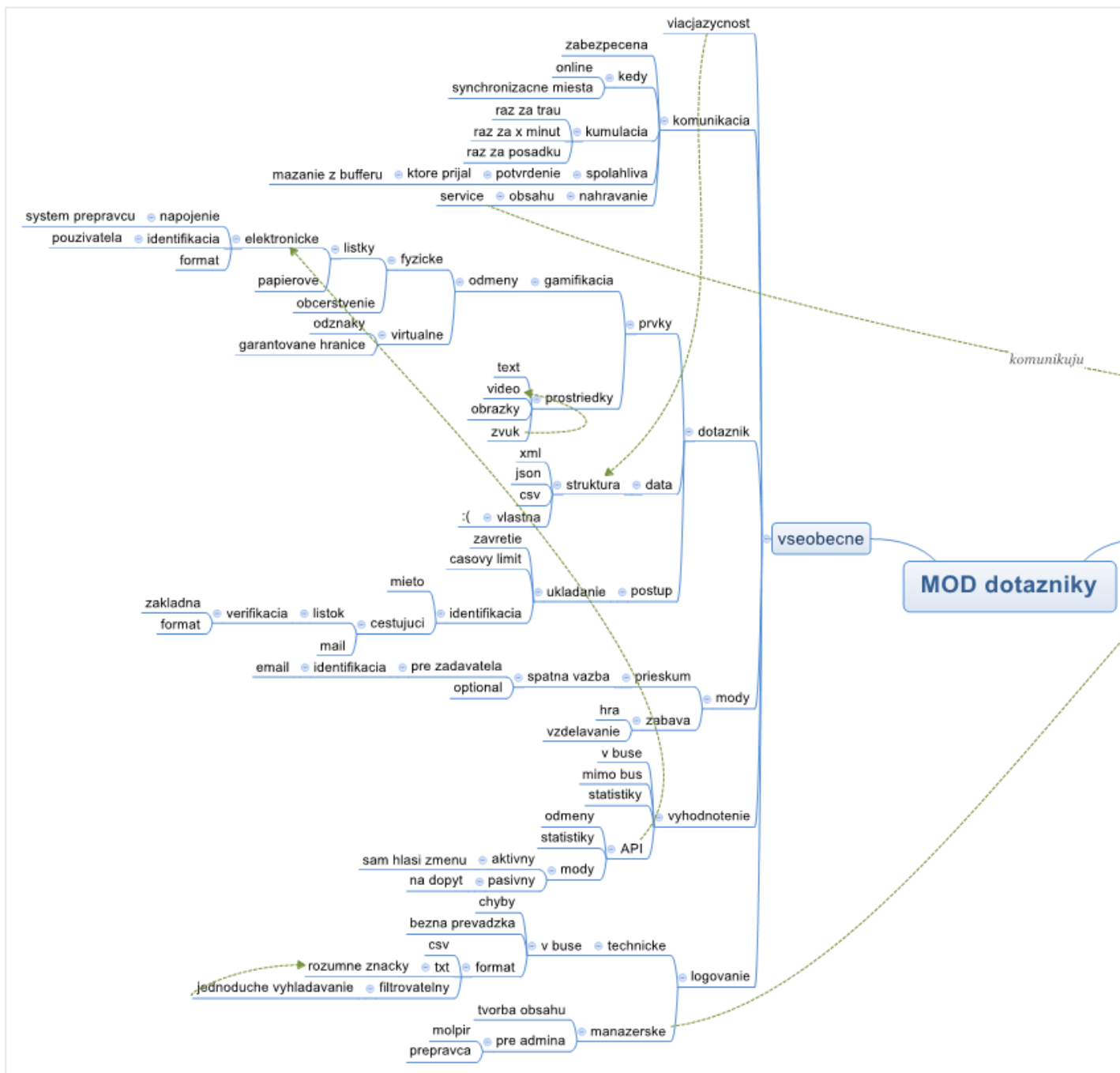
Prvým hlavným cieľom tímového projektu Funtoro MOD pre zimný semester je prenos informácií medzi servermi v autobusoch a webovým serverom majiteľov flotíl. Pri prenose informácií je potrebné zabezpečiť bezpečnú komunikáciu medzi týmito servermi, aby nemohla byť ľahko odchytená treťou stranou. Vďaka komunikácii medzi týmito servermi bude možné odosielať vlastné kvízy (vedomostné, zábavné) majiteľmi flotíl na server v autobusoch, pričom tieto kvízy budú prostriedkom funkcionality tvorenej v ďalších častiach práce. Prenos informácií bude spočívať v prenose súborov, kde sa neskôr budú nachádzať kvízy, medzi týmito servermi. V zimnom semestri je cieľom taktiež umožniť sťahovanie informácií napísaných od používateľa obrazoviek (chápané ako odpoveď na otázku kvízu) do vytvorenej databázy v autobuse (neskôr celkové výsledky kvízov). Tieto informácie majú byť následne zasielané na server mimo autobus.

Ďalším cieľom zimného semestra je vytvorenie webovej aplikácie, do ktorej bude možné prihlasovať sa buď ako admin firmy Molpir, s.r.o., alebo ako admin autobusovej flotily. Na danej webovej stránke bude zabezpečená viacjazyčnosť stránky, čiže je možné prepínanie stránok do ľubovoľne zvolených jazykov (zatiaľ len slovenčina ↔ angličtina). Prihlásenie sa ako admin firmy Molpir, s.r.o. bude mať možnosť pridávať, odoberať alebo meniť zákazníkov vo vytvorenej databáze. Webová aplikácia musí taktiež umožňovať zobrazenie špecifického obsahu.

2.1 Brainstorming ku projektu Funtoro MOD



Obrázok 1. Časť brainstormingu pre webové aplikácie



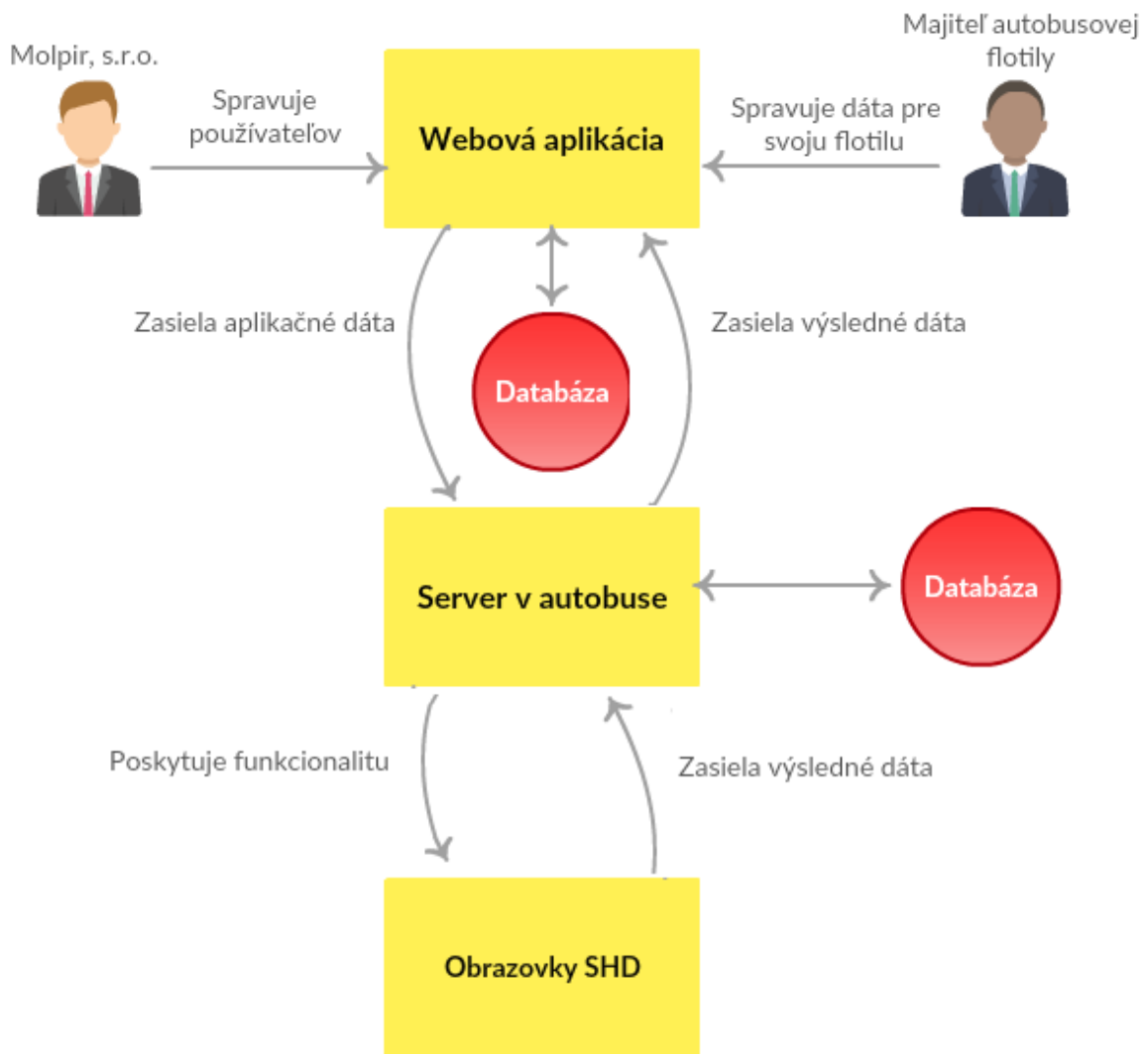
Obrázok 2. Časť brainstormingu pre všeobecné poznatky projektu

3 Celkový pohľad na systém

Náš systém sa skladá z dvoch hlavných častí. Prvá časť je tvorená webovou aplikáciou, do ktorej budú mať prístup Molpir, s.r.o. ako majiteľ produktu a autobusová flotila ako zákazník. Spoločnosť Molpir, s.r.o. bude mať všetky práva nad aplikáciou a bude spravovať všetkých svojich zákazníkov. Bude mať možnosť pridávať nové autobusové flotily a odoberať súčasných zákazníkov, s ktorými im skončí kontrakt. Zákazníci, ktorými budú autobusové flotily, budú mať nad webovou aplikáciou obmedzené práva. Ich prioritné využitie aplikácie je tvorené spravovaním dotazníkov, pre ktoré chceme vyvíjať funkcionality na obrazovky. Cieľom bude vytvoriť otázky pre dotazník, ktorý bude následne zaslaný do autobusov, kde bude sprostredkovaný zákazníkom. Druhou možnosťou pre zákazníkov je taktiež kontrola výsledkov dotazníkov, aby mohli na základe spätnej väzby cestujúcich zlepšovať svoje služby, a týmto spôsobom napredovať. Webová aplikácia je vyvíjaná v jazyku Python pomocou webového frameworku Django.

Druhou časťou, ktorú vyvíjame je funkcionality na SHD obrazovky. Obrazovky neobsahujú žiadnu funkcionality a slúžia len na zobrazovanie. Všetka logika a funkcionality je obsiahnutá na serveri Funtoro MOD, ktorý sa nachádza v autobuse. Všetka funkcionality je sprostredkovaná ako webová aplikácia, čiže všetko čo je vyvíjané na SHD obrazovky je vyvíjané v jazykoch HTML, CSS, Javascript a PHP. Vytvorená funkcionality je sprostredkovaná formou grafického rozhrania na monitoroch, a týmto spôsobom k nej môže pristupovať cestujúci. Zmyslom projektu je vytvoriť dotazníky, ktoré budú mať náučno-výchovný charakter, a zároveň budú poskytovať spätnú väzbu majiteľom autobusových flotíl.

Na obrázku 1 (pozri Obrázok 3) je znázornený celkový pohľad na systém. Do webovej aplikácie pristupuje vlastník produktu a zákazník. Vlastník produktu spravuje používateľov a majiteľ flotily autobusov spravuje dáta pre svoju flotilu. Webová aplikácia je prepojená s databázou. V tejto databáze sa ukladajú dáta, ktoré zasielajú majitelia flotíl do svojich autobusov, a zároveň sa v nej nachádzajú informácie zo spätnej väzby od cestujúcich. Webová aplikácia je prepojená so serverom v autobuse. Z webovej aplikácie odosielajú majitelia flotíl aplikačné dáta pre svoje konkrétne autobusy, pričom tieto dáta sú ďalej spracovávané priamo na serveri v autobuse. Server v autobuse obsahuje všetku funkcionality. Na serveri v autobuse sa nachádza databáza, ktorá obsahuje dáta posielané z webovej aplikácie a informácie zo spätnej väzby cestujúcich. Na serveri sa nachádza funkcionality, ktorá vytiahne dáta z databázy a transformuje ich do formy dotazníku, ktorá je následne poskytnutá zákazníkovi vo forme vedomostnej hry. Výsledky z tejto hry sú po jej ukončení uložené v databáze a zasielajú do databázy napojenej na webovú aplikáciu, kde k nim môže pristupovať majiteľ autobusov. Na serveri beží program, ktorý v určených časových intervaloch prehliada databázu, či do nej pribudli nové dáta a v prípade, ak sa v databáze nachádzajú nové položky, odosiela ich do databázy napojenej na webovú aplikáciu. SHD obrazovky predstavujú obrazovky v autobuse, s ktorými prichádza do kontaktu cestujúci. Cestujúci využíva funkcionality týchto obrazoviek, a pokiaľ sa rozhodne vyplniť dotazník, tak sú dáta z obrazovky ukladané do databázy na serveri autobusu.



Obrázok 3. znázorňuje celkový pohľad na systém.

4 Moduly systému

Moduly systému sú delené na dve časti a to funkcionálna v autobuse a webová aplikácia. Niektoré moduly systému si vyžadujú gamifikačné prvky na prilákanie cestujúcich, a preto bola potrebná ich bližšia analýza. Táto analýza je obsahom Prílohy A.

4.1 SHD obrazovky a server v autobusoch

4.1.1 Spracovanie dát z obrazovky

Analýza

Pre účely zlepšenia dopravy je potrebné získať spätnú väzbu od cestujúcich. Toto je možné pomocou spracovania textu z obrazoviek, pričom text musí mať nejakú ucelenú štruktúru. Chceme však docieľiť, aby zákazník vyplnil dotazník, avšak nechceme ho odradiť dlhými otázkami, ale zaujať ho, aby chcel s nami cestovať aj nabadúce. Musíme vymyslieť vhodnú kombináciu zábavy a „povinností“, ktorá poteší obe strany.

Návrh

Ako formu spätnej väzby vyberieme formu otázok. Otázky však musíme zapracovať do nejakej zábavnej formy, aby ju chcel používateľ vyplniť. Otázku zapracujeme do hry ako miliónár, pričom otázku dotazníka dostaneme každú n-tú úroveň hry. Dáta, ktoré sú počas hry vytvárané, je potrebné následne ukladať na serverovú databázu, ktorá sa nachádza na serveri v autobuse.

Implementácia

Keď je cestujúcemu zobrazená otázka, má pri nej editačné pole, do ktorého môže písať otvorený text. Po tom, ako cestujúci vyjadrí svoj názor môže kliknutím na tlačidlo odoslať text do databázy. Implementácia je riešená prostredníctvom HTML, ktoré sprostredkuje cestujúcemu editačné pole, následne je logika riešená cez Javascript a pomocou Ajaxu je prepojený s metódou napísanou v PHP, ktorá nám otvára pripojenie do databázy a vloží do nej následne aj text.

Testovanie

Keďže sme obmedzovaný možnosťami obrazovky, tak všetky testy zatiaľ prebiehali ručne. Prostredníctvom obrazoviek sme ukladali rôzne texty a sledovali aktualizáciu databázy.

4.1.2 Spracovanie textu z databázy na pozadí

Analýza

Spätná väzba uložená v autobuse musí byť nejakým spôsobom sprostredkovaná vedeniu autobusovej flotily, aby mohla byť zužitkovaná. Z tohto dôvodu musí byť vymyslený spôsob ako pravidelne vyťahovať údaje z databázy.

Návrh

Spätná väzba bude z databázy vyťahovať informácie v istých časových intervaloch, aby udržiavala vždy aktuálnu spätnú väzbu. Na pozadí servera musí bežať skript, ktorý bude kontrolovať databázu a vyťahovať len logy, ktoré ešte neboli vytiahnuté aby nevznikali zbytočné duplicity a nebol chaos. Tento modul bude prepojený s komunikačným modulom.

Implementácia

Vytvorili sme skripty, ktoré dokážu vytiahnuť dáta z databázy. Tieto skripty sú spúšťané na pozadí službou KRON, ktorá ich dokáže spúšťať v pravidelných časových intervaloch. Z každého pokusu o vytiahnutie údajov z databázy sa vytvorí log, ktorý obsahuje čas, kedy výber z databázy prebehol, aby sme mohli na základe toho selektovať len nové dáta. Ďalším prvkom logu je či prebehol úspešne, alebo nie. V prípade, že neprebehol úspešne považuje sa za nerelevantný a v tom prípade sa selektujú dáta od najbližšieho úspešného logu. Skripty sú vytvorené v jazyku PHP.

Testovanie

Testy prebiehali pravidelným vkladaním dát do databázy a následnou kontrolou logov, ktoré boli vytvárané skriptami na pozadí.

4.1.3 Komunikácia servera v autobuse so serverom mimo autobusu

Analýza

Aby mohla byť spätná väzba sprostredkovaná globálne zo všetkých autobusov a mohla byť kontrolovaná, musí byť zabezpečená komunikácia servera mimo autobus na spoločný server, kde sa budú dáta zhromažďovať a následne vyhodnocovať. Komunikácia musí byť bezpečná, aby bola dostatočná ochrana údajov a nemohol sa k nim hocikto len tak dostať.

Návrh

Pri komunikácii mimo server treba pravidelne overovať internetové pripojenie. Bez pripojenia na internet neodosielame žiadne dáta. Skript, ktorý vyťahuje dáta z databázy, zároveň musí skontrolovať pripojenie a ak je úspešná odozva, dáta sa odošlú na server, kde sú ďalej spracované.

Komunikácia cez JSON, http POST CURL , AUTENTIFIKACCIA = MENO HESLO, TOKEN

Implementácia

Komunikácia medzi servermi prebieha v JSON formátoch, ktoré zabezpečujú dobré čítanie a posielanie. Celá komunikácia je http-POST, pričom sa vykonáva metóda CURL, ktorá transferuje dáta. V rámci prenosu dát je zabezpečená autentifikácia pomocou troch základných údajov a to mena, hesla a čísla tokenu. Týmto spôsobom je ošetrené odchyťovanie dát treťou stranou.

Testovanie

Testovanie komunikácia prebiehalo pravidelným kontrolovaním dát na druhej strane servera. Ak sa dáta v databáze mimo autobusového servera zhodovali s dátami, ktoré sa nachádzali v autobusovom serveri, boli považované testy za úspešné.

4.2 Webová aplikácia

4.2.1 Autentifikácia používateľa

Analýza

V systéme Funtoro MOD sa nachádzajú dva typy používateľov – admin Molpir, s.r.o a admini autobusových flotíl. Webová aplikácia umožňuje adminovi firmy Molpir, s.r.o. pridávať nových zákazníkov do databázy alebo odoberať starých zákazníkov. Adminom autobusových flotíl umožňuje spravovať vlastné autobusy a hlavne vytvárať dotazníky/kvízy podľa určitého obsahu, ktoré budú odoslané do určitých ich autobusov. Pre umožnenie oboch funkcionalít tohto systému je potrebná autentifikácia používateľa webovej stránky.

Návrh

Do webovej aplikácie bude používateľom tohto systému umožnené zaregistrovať sa. Registrácia a prihlasovanie je vykonávané prostredníctvom django admin panelu. Úspešné prihlásenie sa závisí od zaregistrovaných zákazníkov systému. Celkovú správu zákazníkov má na starosti admin Molpir, s.r.o., ktorý môže zákazníkov pridávať, odoberať, meniť.

Implementácia

Pri registrácii používateľ zadáva svoje osobné údaje – email a heslo. Po potvrdení registrácie sa prostredníctvom webovej služby následne overí, či sa používateľ s danými osobnými údajmi v databáze už nenachádza. V prípade, že sa daný používateľ v databáze nenachádza, na zadaný email mu príde správa ohľadom potvrdenia registrácie prostredníctvom webovej služby. Pri potvrdení registrácie cez email sa osobné údaje používateľa zaznamenajú do databázy. Pri prihlasovaní sa overí, či daný používateľ nebol adminom firmy Molpir, s.r.o. deaktivovaný (či sa stále nachádza v databáze, ktorý ma na starosti správu zákazníkov).

Testovanie

Testovanie daného modulu spočíva zatiaľ v manuálnom testovaní. Vytvárané registrácie sú následne kontrolované v databáze a overuje sa aj prihlasovanie chybnými údajmi.

4.2.2 Viacjazyčnosť

Analýza

Webová aplikácia bude poskytovaná medzinárodným klientom – medzinárodné autobusové spoločnosti, a tým pádom musí byť zabezpečená viacjazyčná webová aplikácia. Prostredníctvom rozhrania aplikácie bude môcť vykonávať používateľ špecifické úkony konkretizované na konkrétnu oblasť, a preto je potrebné, aby mu bol poskytnutý jazyk jemu vlastný.

Návrh

Na webovej stránke bude zabezpečené prepínanie jazykov v ktoromkoľvek momente. Používateľ si vyberie zo zoznamu jazykov a na základe zvoleného jazyka sa mu následne sprostredkuje celý obsah v danom jazyku. Viacjazyčnosť nebude spočívať v automatizovanom prekladaní slov na stránke, ale v zobrazovaní špecifického textu pre jednotlivé jazyky.

Implementácia

Na stránke je vytvorených viacero jazykových verzií textov, ktoré sa následne mapujú na webovú stránku. Jedna služba kontroluje potvrdený jazyk na stránke, ktorý tvorí príznak, podľa ktorého sa určuje druh súboru, z ktorého sa načítavajú jednotlivé texty. Jednotlivé súbory sú rozdelené na základe jazykov.

Testovanie

Testovanie spočíva v mapovaní súborov na stránku a kontrolovaní ako sa namapovali dané texty. Ak sedí jazyková verzia textu s jazykom zvoleným na stránke, považuje sa test za úspešný.

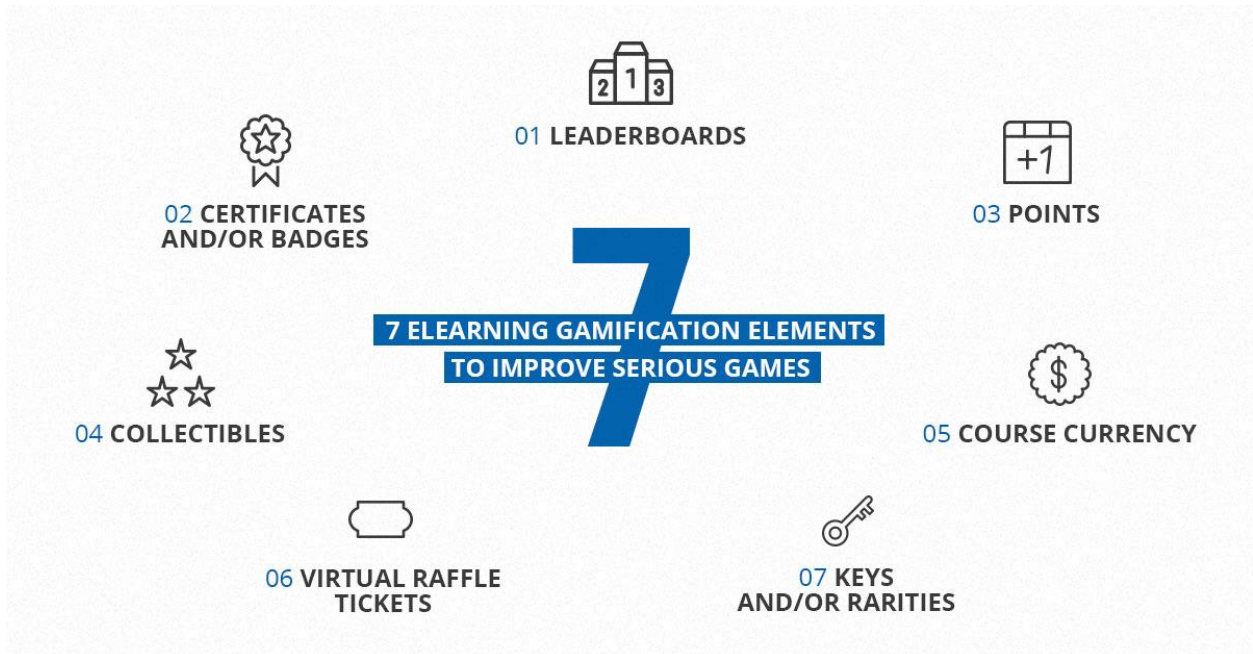
5 Prílohy

- Príloha A: Analýza gamifikačných prvkov (aj v angličtine)

Priloha A

A.1 Game elements

<https://www.docebo.com/2017/01/17/7-gamification-elements-for-elearning-serious-games/>



1. Leaderboards

Leaderboards are one of the most competitive gamification elements, as they encourage corporate learners to surpass their peers in order to achieve “first place” status. It’s the ideal game mechanic for those who want to be publicly recognized or praised, which makes it more extrinsic than intrinsic when it comes to motivation.

To get the most out of leaderboards, allow corporate learners to opt out if they aren’t comfortable competing with their colleagues. This ensures that your introverted corporate learners aren’t alienated, but still gives others the chance to work toward the top tier.

2. Points

Points are earned by completing tasks or participating in voluntary online training activities. Once learners have collected a certain amount of points they can trade them in for a reward or unlock new course levels.

3. Certificates and/or Badge

Certificates and badges are more tangible rewards. Corporate learners must reach their goals or show their expertise in order to receive a certificate. When more “minor” milestones are reached (e.g., if they successfully pass an exam or make their way through a branching scenario), they can receive a badge to track their progress.

Certificates are ideal for compliance online training courses, such as online safety training. If necessary, the certificates can have an expiration date, at which time a corporate learner must renew their certificate by retaking the online training course.

4. Collectibles

Whether it’s baseball cards or in-course items, humans tend to enjoy collecting things. Corporate learners can work toward complete their course collection. This same rule applies to trading. If another

corporate learner has an item they are missing from their collection, they must communicate with them in order to strike a deal. This process makes your online training experiences more social and interactive.

5. Course Currency

Who doesn't like money? You can use this to your advantage by creating a course currency that corporate learners must amass throughout their online training. They can use this currency to unlock online training content, access helpful tips, or even purchase tangible rewards. Be sure that you create a set of guidelines and stress the fact that the currency doesn't hold any monetary value.

6. Virtual Raffle Tickets

Give corporate learners the opportunity to earn tickets during the online training program, then hold a raffle at the end where they can win prizes. This particular game mechanic can be tricky, since corporate learners aren't participating simply for the sake of learning—they are in it to win, as they say, and earning rewards is their primary goal. This is why it's always a good idea to pair it with another game mechanic that is more intrinsic.

7. Keys and/or Rarities

This is a gamification element taken straight from video and PC games. Corporate learners must complete an online training task or module to earn a key. That key will unlock the next level or they are required to save up a certain number of keys to unlock a reward. If you want to take it a step further you can even create keys that will grant access to "hidden" content, treasure chests, or doors that lead to supplemental online training resources, like fun games or puzzles.

Corporate learners who are more adventurous or become easily distracted will love the idea of unlocking rarities. For example, let your corporate audience know that there are only 5 rare objects hidden throughout the online training course (e.g., bonus points or one-of-a-kind badges). Corporate learners can unlock these items by completing special tasks or outperforming their peers.

Game mechanics

<https://4brain.ru/gamification/igrovye-mehaniki.php>

<https://techcrunch.com/2010/08/25/scvng-r-game-mechanics/>

1. Achievement

Definition: A virtual or physical representation of having accomplished something. These are often viewed as rewards in and of themselves.

Example: a badge, a level, a reward, points, really anything defined as a reward can be a reward.

2. Appointment Dynamic

Definition: A dynamic in which to succeed, one must return at a predefined time to take some action. Appointment dynamics are often deeply related to interval based reward schedules or avoidance dynamics.

Example: Cafe World and Farmville where if you return at a set time to do something you get something good, and if you don't something bad happens.

3. Avoidance

Definition: The act of inducing player behavior not by giving a reward, but by not instituting a punishment. Produces consistent level of activity, timed around the schedule.

Example: Press a lever every 30 seconds to not get shocked.

4. Behavioral Contrast

Definition: The theory defining how behavior can shift greatly based on changed expectations.

Example: A monkey presses a lever and is given lettuce. The monkey is happy and continues to press the lever. Then it gets a grape one time. The monkey is delighted. The next time it presses the lever it gets lettuce again. Rather than being happy, as it was before, it goes ballistic throwing the lettuce at the experimenter. (In some experiments, a second monkey is placed in the cage, but tied to a rope so it can't access the lettuce or lever. After the grape reward is removed, the first monkey beats up the second monkey even though it obviously had nothing to do with the removal. The anger is truly irrational.)

5. Behavioral Momentum

Definition: The tendency of players to keep doing what they have been doing.

Example: From Jesse Schell's awesome Dice talk: "I have spent ten hours playing Farmville. I am a smart person and wouldn't spend 10 hours on something unless it was useful. Therefore this must be useful, so I can keep doing it."

6. Blissful Productivity

Definition: The idea that playing in a game makes you happier working hard, than you would be relaxing. Essentially, we're optimized as human beings by working hard, and doing meaningful and rewarding work.

Example: From Jane McGonical's Ted Talk wherein she discusses how World of Warcraft players play on average 22 hours / week (a part time job), often after a full days work. They're willing to work hard, perhaps harder than in real life, because of their blissful productivity in the game world.

7. Cascading Information Theory

Definition: The theory that information should be released in the minimum possible snippets to gain the appropriate level of understanding at each point during a game narrative.

Example: showing basic actions first, unlocking more as you progress through levels. Making building on SCVNGR a simple but staged process to avoid information overload.

8. Chain Schedules

Definition: the practice of linking a reward to a series of contingencies. Players tend to treat these as simply the individual contingencies. Unlocking one step in the contingency is often viewed as an individual reward by the player.

Example: Kill 10 orcs to get into the dragons cave, every 30 minutes the dragon appears.

9. Communal Discovery

Definition: The game dynamic wherein an entire community is rallied to work together to solve a riddle, a problem or a challenge. Immensely viral and very fun.

Example: DARPA balloon challenge, the cottage industries that appear around McDonalds monopoly to find "Boardwalk"

10. Companion Gaming

Definition: Games that can be played across multiple platforms

Example: Games that be played on iphone, facebook, xbox with completely seamless cross platform gameplay.

11. Contingency

Definition: The problem that the player must overcome in the three part paradigm of reward schedules.

Example: 10 orcs block your path

12. Countdown

Definition: The dynamic in which players are only given a certain amount of time to do something. This will create an activity graph that causes increased initial activity increasing frenetically until time runs out, which is a forced extinction.

Example: Bejeweled Blitz with 30 seconds to get as many points as you can. Bonus rounds. Timed levels

13. Cross Situational Leader-boards

Definition: This occurs when one ranking mechanism is applied across multiple (unequal and isolated) gaming scenarios. Players often perceive that these ranking scenarios are unfair as not all players were presented with an “equal” opportunity to win.

Example: Players are arbitrarily sent into one of three paths. The winner is determined by the top scorer overall (i.e. across the paths). Since the players can only do one path (and can’t pick), they will perceive inequity in the game scenario and get upset.

14. Disincentives

Definition: a game element that uses a penalty (or altered situation) to induce behavioral shift

Example: losing health points, amazon’s checkout line removing all links to tunnel the buyer to purchase, speeding traps

15. Endless Games

Definition: Games that do not have an explicit end. Most applicable to casual games that can refresh their content or games where a static (but positive) state is a reward of its own.

Example: Farmville (static state is its own victory), SCVNGR (challenges constantly are being built by the community to refresh content)

16. Envy

Definition: The desire to have what others have. In order for this to be effective seeing what other people have (voyeurism) must be employed.

Example: my friend has this item and I want it!

17. Epic Meaning

Definition: players will be highly motivated if they believe they are working to achieve something great, something awe-inspiring, something bigger than themselves.

Example: From Jane McGonical’s Ted Talk where she discusses Warcraft’s ongoing story line and “epic meaning” that involves each individual has motivated players to participate outside the game and create the second largest wiki in the world to help them achieve their individual quests and collectively their epic meanings.

18. Extinction

Definition: Extinction is the term used to refer to the action of stopping providing a reward. This tends to create anger in players as they feel betrayed by no longer receiving the reward they have come to expect. It generally induces negative behavioral momentum.

Example: killing 10 orcs no longer gets you a level up

19. Fixed Interval Reward Schedules

Definition: Fixed interval schedules provide a reward after a fixed amount of time, say 30 minutes. This tends to create a low engagement after a reward, and then gradually increasing activity until a reward is given, followed by another lull in engagement.

Example: Farmville, wait 30 minutes, crops have appeared

20. Fixed Ratio Reward Schedule

Definition: A fixed ratio schedule provides rewards after a fixed number of actions. This creates cyclical nadirs of engagement (because the first action will not create any reward so incentive is low) and then bursts of activity as the reward gets closer and closer.

Example: kill 20 ships, get a level up, visit five locations, get a badge

21. Free Lunch

Definition: A dynamic in which a player feels that they are getting something for free due to someone else having done work. It's critical that work is perceived to have been done (just not by the player in question) to avoid breaching trust in the scenario. The player must feel that they've "lucked" into something.

Example: Groupon. By virtue of 100 other people having bought the deal, you get it for cheap. There is no sketchiness b/c you recognize work has been done (100 people are spending money) but you yourself didn't have to do it.

22. Fun Once, Fun Always

Definition: The concept that an action is enjoyable to repeat all the time. Generally this has to do with simple actions. There is often also a limitation to the total level of enjoyment of the action.

Example: the theory behind the check-in everywhere and the check-in and the default challenges on SCVNGR.

23. Interval Reward Schedules

Definition: Interval based reward schedules provide a reward after a certain amount of time. There are two flavors: variable and fixed.

Example: wait N minutes, collect rent

24. Lottery

Definition: A game dynamic in which the winner is determined solely by chance. This creates a high level of anticipation. The fairness is often suspect, however winners will generally continue to play indefinitely while losers will quickly abandon the game, despite the random nature of the distinction between the two.

Example: many forms of gambling, scratch tickets.

25. Loyalty

Definition: The concept of feeling a positive sustained connection to an entity leading to a feeling of partial ownership. Often reinforced with a visual representation.

Example: fealty in WOW, achieving status at physical places (mayorship, being on the wall of favorite customers)

26. Meta Game

Definition: a game which exists layered within another game. These generally are discovered rather than explained (lest they cause confusion) and tend to appeal to ~2% of the total gameplaying audience. They are dangerous as they can induce confusion (if made too overt) but are powerful as they're greatly satisfying to those who find them.

Example: hidden questions / achievements within world of warcraft that require you to do special (and hard to discover) activities as you go through other quests

27. Micro Leader-boards

Definition: The rankings of all individuals in a micro-set. Often great for distributed game dynamics where you want many micro-competitions or desire to induce loyalty.

Example: Be the top scorers at Joe's bar this week and get a free appetizer

28. Modifiers

Definition: An item that when used affects other actions. Generally modifiers are earned after having completed a series of challenges or core functions.

Example: A X2 modifier that doubles the points on the next action you take.

29. Moral Hazard of Game Play

Definition: The risk that by rewarding people manipulatively in a game you remove the actual moral value of the action and replace it with an ersatz game-based reward. The risk that by providing too many incentives to take an action, the incentive of actually enjoying the action taken is lost. The corollary to this is that if the points or rewards are taken away, then the person loses all motivation to take the (initially fun on its own) action.

Example: Paraphrased from Jesse Schell "If I give you points every time you brush your teeth, you'll stop brushing your teeth b/c it's good for you and then only do it for the points. If the points stop flowing, your teeth will decay."

30. Ownership

Definition: The act of controlling something, having it be **your** property.

Example: Ownership is interesting on a number of levels, from taking over places, to controlling a slot, to simply owning popularity by having a digital representation of many friends.

31. Pride

Definition: the feeling of ownership and joy at an accomplishment

Example: I have ten badges. I own them. They are mine. There are many like them, but these are mine. Hooray.

32. Privacy

Definition: The concept that certain information is private, not for public distribution. This can be a demotivator (I won't take an action because I don't want to share this) or a motivator (by sharing this I reinforce my own actions).

Example: Scales the publish your daily weight onto Twitter (these are real and are proven positive motivator for staying on your diet). Or having your location publicly broadcast anytime you do anything (which is invasive and can should be avoided).

33. Progression Dynamic

Definition: a dynamic in which success is granularly displayed and measured through the process of completing itemized tasks.

Example: a progress bar, leveling up from paladin level 1 to paladin level 60

34. Ratio Reward Schedules

Definition: Ratio schedules provide a reward after a number of actions. There are two flavors: variable and fixed.

Example: kill 10 orcs, get a power up.

35. Real-time v. Delayed Mechanics

Definition: Realtime information flow is uninhibited by delay. Delayed information is only released after a certain interval.

Example: Realtime scores cause instant reaction (gratification or demotivation). Delayed causes ambiguity which can incent more action due to the lack of certainty of ranking.

36. Reinforcer

Definition: The reward given if the expected action is carried out in the three part paradigm of reward schedules.

Example: receiving a level up after killing 10 orcs.

37. Response

Definition: The expected action from the player in the three part paradigm of reward schedules.

Example: the player takes the action to kill 10 orcs

38. Reward Schedules

Definition: the timeframe and delivery mechanisms through which rewards (points, prizes, level ups) are delivered. Three main parts exist in a reward schedule; contingency, response and reinforcer.

Example: getting a level up for killing 10 orcs, clearing a row in Tetris, getting fresh crops in Farmville

39. Rolling Physical Goods

Definition: A physical good (one with real value) that can be won by anyone on an ongoing basis as long as they meet some characteristic. However, that characteristic rolls from player to player.

Example: top scorer deals, mayor deals

40. Shell Game

Definition: a game in which the player is presented with the illusion of choice but is actually in a situation that guides them to the desired outcome of the operator.

Example: 3 Card Monty, lotteries, gambling

41. Social Fabric of Games

Definition: the idea that people like one another better after they've played games with them, have a higher level of trust and a great willingness to work together.

Example: From Jane McGonigal's TED talk where she suggests that it takes a lot of trust to play a game with someone because you need them to spend their time with you, play by the same rules, shoot for the same goals.

42. Status

Definition: The rank or level of a player. Players are often motivated by trying to reach a higher level or status.

Example: white paladin level 20 in WOW.

43. Urgent Optimism

Definition: Extreme self motivation. The desire to act immediately to tackle an obstacle combined with the belief that we have a reasonable hope of success.

Example: From Jane McGonigal's TED talk. The idea that in proper games an "epic win" or just "win" is possible and therefore always worth acting for.

44. Variable Interval Reward Schedules

Definition: Variable interval reward schedules provide a reward after a roughly consistent amount of time. This tends to create a reasonably high level of activity over time, as the player could receive a reward at any time but never the burst as created under a fixed schedule. This system is also more immune to the nadir right after the receiving of a reward, but also lacks the zenith of activity before a reward in unlocked due to high levels of ambiguity.

Example: Wait roughly 30 minutes, a new weapon appears. Check back as often as you want but that won't speed it up. Generally players are bad at realizing that.

45. Variable Ratio Reward Schedule

Definition: A variable ratio reward schedule provides rewards after a roughly consistent but unknown amount of actions. This creates a relatively high consistent rate of activity (as there could always be a reward after the next action) with a slight increase as the expected reward threshold is reached, but

never the huge burst of a fixed ratio schedule. It's also more immune to nadirs in engagement after a reward is achieved.

Example: kill something like 20 ships, get a level up. Visit a couple locations (roughly five) get a badge

46. Viral Game Mechanics

Definition: A game element that requires multiple people to play (or that can be played better with multiple people)

Example: Farmville making you more successful in the game if you invite your friends, the social check-in

47. Virtual Items

Definition: Digital prizes, rewards, objects found or taken within the course of a game. Often these can be traded or given away.

Example: Gowalla's items, Facebook gifts, badges

A.2 Analýza

Glosár:

Vzdelávací dotazník	VD
Kvizový dotazník	KD
Dotazník pre spätnú väzbu	SD

Tabuľka 1. Gamifikačné prvky.

	Prvok	Kde	Kedy	Ako
1	Leaderboard	VD, KD	Počas alebo po vyplnení	Počas vyplnenia ukázať najlepších hráčov alebo v akom percente najlepších je výsledok hráča (možno nie po každej odpovedi, ale po 3-5). Po vyplnení celého dotazníka ukázať najlepších hráčov, a na ktorom mieste je výsledok hráča.
2	Points	VD, KD	Počas alebo po vyplnení	Počas alebo po vyplnení ukázať body získané hráčom.
6	Virtual Raffle Tickets	KD	Po vyplnení	Keď je výsledok dobrý, dať hráčovi tiket, ktorý sa potom dá vymeniť za kavičku, zľavu na lístok atď.
7	Keys and/or Rarities	VD, KD	Počas vyplnenia	Nasledujúci level dotazníka sa odblokuje, keď vyplníme predchádzajúci.

Tabuľka 2. Gamifikačné mechaniky

	Mechanika	Kde	Ako
1	Achievement	VD, KD	Body, levely.
5	Behavioral Momentum	VD, KD, SD	Keď dáme niekoľko zaujímavých otázok a uprostred nich jednu z SD, tak si to hráč ani nevšimne, samotný dotazník na niekoľko otázok je tiež príkladom.
7	Cascading Information Theory	VD, KD	Niekoľko prvých otázok môže byť jednoduchých, ďalej komplikovanejšie.
8	Chain Schedules	VD, KD, SD	Vyplnenie niekoľkých otázok správne môže odblokovať tip, právo na chybu alebo ďalší level.
11	Contingency	VD, KD, SD	S nejakou pravdepodobnosťou dáme hráčovi otázku iného typu z iného dotazníka, alebo s obmedzením či modifikátorom.
12	Countdown	VD, KD, SD	Obmedzíme minimálny a/alebo maximálny čas na vyplnenie dotazníka. Vieme tiež použiť spolu s (11).
14	Disincentives	VD, KD	Chyby znižujú počet získaných bodov, niekoľko chýb za sebou ukončí hru.
16	Envy	VD, KD, SD	Povieme hráčovi, že už %počet% ľudí vyplnilo dotazník a vyhralo keksík či zľavu na lístok.
17	Epic Meaning	SD	Vyplnením tohto dotazníka viete ovplyvniť robotu celej dopravy (resp. stvorenie novej trasy). Vyplnením tejto elektronickej verzie dotazníka zachránite prírodu, stromček atď.
24	Lottery	KD, SD	Vyplnenie dotazníka otvára možnosť účasti v tombole.
25	Loyalty	VD, KD, SD	???identifikácia???
26	Meta Game	VD, KD, SD	Nejaká skrytá mini-hra. Môže byť kombinované s (11).
28	Modifiers	VD, KD	x2, x3 bodov za odpoveď. Môže byť kombinované s (11, 12, 26).

33	Progression Dynamic	VD, KD, SD	Progressbar či niečo iné, čo ukáže hráčovi jeho pokrok.
35	Real-time v. Delayed Mechanics	VD, KD	Môžem rozdeliť dotazník na bloky a ukazovať výsledky len na konci bloku.
39	Rolling Physical Goods	KD, SD	Materiálna odmena najlepším.
40	Shell Game	VD, KD, SD	Môžeme dať na výber niekoľko typov otázok alebo možnosť meniť ich poradie. Môže byť kombinované s (11).