

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Dokumentácia k riadeniu projektu

Bc. Michal Cihák

Bc. Michal Gajdoš

Bc. Lukáš Masár

Bc. Pavol Michálek

Bc. Vladimír Osvald

Bc. Matúš Pikuliak

Bc. Tomáš Sýkora

Tím (č. 12): Cats Can Code

Predmet: Tímový projekt I

Ročník: 1.

Akademický rok: 2014/2015

Vedúci práce: Ing. Lukáš Turský

Obsah

1	ÚVOD.....	1
1.1	PODIEL ČLENOV NA DOKUMENTÁCIÍ	2
2	ROLE ČLENOV TÍMU	4
2.1	MANAŽÉRSKE ÚLOHY.....	4
2.2	APLIKÁCIA MANAŽÉRSKÝCH ÚLOH	4
2.2.1	<i>Manažment verziovania</i>	<i>4</i>
2.2.2	<i>Manažment kvality.....</i>	<i>5</i>
2.2.3	<i>Manažment integrácie projektu</i>	<i>5</i>
2.2.4	<i>Manažment dokumentovania</i>	<i>6</i>
2.2.5	<i>Manažment testovania.....</i>	<i>6</i>
2.2.6	<i>Manažment komunikácie.....</i>	<i>6</i>
2.2.7	<i>Manažment plánovania</i>	<i>7</i>
2.2.8	<i>Manažment rizík</i>	<i>8</i>
3	SUMARIZÁCIA ŠPRINTOV	9
3.1	PRVÝ ŠPRINT	9
3.2	DRUHÝ ŠPRINT	9
3.3	TRETÍ ŠPRINT	9
3.4	ŠTVRTÝ ŠPRINT	9
3.5	PIATY ŠPRINT	9
3.6	ŠIESTY ŠPRINT.....	10
3.7	SIEDMY ŠPRINT	10
3.8	ÔSMY ŠPRINT	10
3.9	DEVIATY ŠPRINT	10
3.10	DESIATY ŠPRINT	10
3.11	JEDENÁSTY ŠPRINT.....	10
3.12	DVANÁSTY ŠPRINT	11
4	POUŽÍVANÉ METODIKY	12
4.1	METODIKA TESTOVANIA	12
4.2	METODIKA EVIDENCIE ÚLOH	12
4.3	METODIKA VERZIOVANIA.....	12

4.4	METODIKA DOKUMENTOVANIA	13
4.5	METODIKA REVÍZIE KÓDU	13
4.6	METODIKA KOMUNIKÁCIE.....	13
4.7	METODIKA KOMENTOVANIA KÓDU.....	13
5	 GLOBÁLNA RETROSPEKTÍVNA.....	14
5.1	PO ZIMNOM SEMESTRI	14
5.2	PO LETNOM SEMESTRI	15
6	 ZOZNAM KOMPETENCIÍ TÍMU	1
7	 METODIKY	1
7.1	KOMENTOVANIE ZDROJOVÉHO KÓDU.....	1
7.1.1	<i>Povolenie komentárov slúžiacich na dokumentáciu.....</i>	<i>1</i>
7.1.2	<i>Všeobecné pokyny k písaniu komentárov.....</i>	<i>1</i>
7.1.3	<i>Komentovanie v Ruby.....</i>	<i>2</i>
7.1.4	<i>Komentovanie v JavaScript.....</i>	<i>2</i>
7.1.5	<i>Komentovanie tried.....</i>	<i>2</i>
7.1.6	<i>Komentovanie metód.....</i>	<i>3</i>
7.1.7	<i>Odstránenie komentáru z dokumentácie.....</i>	<i>4</i>
7.1.8	<i>Komentovanie premenných.....</i>	<i>4</i>
7.1.9	<i>Ostatné typy komentárov.....</i>	<i>5</i>
7.1.10	<i>“TODO” komentáre</i>	<i>5</i>
7.1.11	<i>“FIXME” komentáre.....</i>	<i>5</i>
7.1.12	<i>Vygenerovanie dokumentácie.....</i>	<i>5</i>
7.2	EVIDENCIA ÚLOH V NÁSTROJI JIRA	7
7.2.1	<i>Roly a zodpovednosti účastníkov.....</i>	<i>7</i>
7.2.2	<i>Typy úloh.....</i>	<i>7</i>
7.2.3	<i>Pomenovanie úloh</i>	<i>8</i>
7.2.4	<i>Vytvorenie úlohy typu Task.....</i>	<i>8</i>
7.2.5	<i>Vytvorenie úlohy typu Bug.....</i>	<i>9</i>
7.2.6	<i>Zmena názvu úlohy</i>	<i>9</i>
7.2.7	<i>Zmena popisu alebo akceptačných kritérií úlohy.....</i>	<i>10</i>
7.2.8	<i>Priradenie úlohy riešiteľovi / zmena riešiteľa.....</i>	<i>10</i>
7.2.9	<i>Pridanie komentáru k úlohe</i>	<i>11</i>

7.3	METODIKA TESTOVANIA	12
7.3.1	Vytvorenie nového testu	12
7.3.2	Vytvorenie nového testu typu <i>Feature</i>	13
7.3.3	Vytvorenie nového testu typu <i>Feature</i> s použitím <i>Selenium</i>	14
7.3.4	Vytvorenie nového testu typu <i>Model</i>	16
7.3.5	Manuálna kontrola testov.....	16
7.4	METODIKA – CODE REVIEW	18
7.4.1	Súhrn rol a zodpovednosti účastníkov.....	18
7.4.2	Kedy sa vykonáva prehliadka kódu	18
7.4.3	Kto vykonáva prehliadku kódu.....	19
7.4.4	Náhľad metodiky.....	19
7.4.5	Postup metodiky.....	19
7.4.6	Ďoplňujúce informácie a požiadavky	20
7.5	METODIKA VERZIOVANIA V GIT A RUBYMine	22
7.5.1	Vytvorenie nového lokálneho projektu	22
7.5.2	Aktualizácia pracovnej kópie súborov.....	23
7.5.3	Commitovanie.....	24
7.5.4	Aktualizácia vetvy vzdialeného repozitára	25
7.5.5	Správa vetiev.....	26
7.5.6	Vytváranie releasu.....	27
7.5.7	Features vetvy.....	28
7.5.8	Zmena pracovnej vetvy.....	28
7.5.9	Vytvorenie novej vetvy.....	28
7.5.10	Mergovanie vetiev	29
7.5.11	Tagovanie commitov	29
7.6	METODIKA KOMUNIKÁCIE.....	30
7.6.1	Kompetencie členov tímu	30
7.6.2	Interná tímová komunikácia - nástroj <i>HipChat</i>	30
7.6.3	Miestnosti.....	31
7.6.4	Špeciálna interná tímová komunikácia - nástroj <i>Google Hangouts</i>	32
7.6.5	Webové sídlo tímu.....	32
7.6.6	Komunikácia o úlohách.....	32
7.6.7	Zdieľanie súborov - <i>Google Drive</i>	33

7.7	METODIKA DOKUMENTOVANIA INŽINIERSKEHO DIELA	34
7.7.1	<i>Identifikácia rolí</i>	34
7.7.2	<i>Dostupnosť dokumentácie</i>	34
7.7.3	<i>Dokumentovanie</i>	34
7.7.4	<i>Proces editovania</i>	35
7.7.5	<i>Proces finalizácie</i>	37
7.7.6	<i>Pravidlá formátovania výsledného dokumentu</i>	38
8	ZÁPISNICE ZO STRETNUTÍ	1
8.1	ZÁPISNICA STRETNUTIA Č. 1.....	1
8.1.1	<i>Osnova na stretnutie</i>	1
8.1.2	<i>Priebeh stretnutia</i>	1
8.2	ZÁPISNICA STRETNUTIA Č. 2.....	3
8.2.1	<i>Osnova na stretnutie</i>	3
8.2.2	<i>Priebeh stretnutia</i>	3
8.3	ZÁPISNICA STRETNUTIA Č. 3.....	6
8.3.1	<i>Osnova na stretnutie</i>	6
8.3.2	<i>Priebeh stretnutia</i>	6
8.4	ZÁPISNICA STRETNUTIA Č. 4.....	8
8.4.1	<i>Osnova na stretnutie</i>	8
8.4.2	<i>Priebeh stretnutia</i>	8
8.5	ZÁPISNICA STRETNUTIA Č. 5.....	10
8.5.1	<i>Osnova na stretnutie</i>	10
8.5.2	<i>Priebeh stretnutia</i>	10
8.6	ZÁPISNICA STRETNUTIA Č. 6.....	12
8.6.1	<i>Osnova na stretnutie</i>	12
8.6.2	<i>Priebeh stretnutia</i>	12
8.7	ZÁPISNICA STRETNUTIA Č. 7.....	14
8.7.1	<i>Osnova na stretnutie</i>	14
8.7.2	<i>Priebeh stretnutia</i>	14
8.8	ZÁPISNICA STRETNUTIA Č. 8.....	16
8.8.1	<i>Osnova na stretnutie</i>	16
8.8.2	<i>Priebeh stretnutia</i>	16
8.9	ZÁPISNICA STRETNUTIA Č. 9.....	18

8.9.1	<i>Osnova na stretnutie</i>	18
8.9.2	<i>Priebeh stretnutia</i>	18
8.10	ZÁPISNICA STRETNUTIA č. 10.....	20
8.10.1	<i>Osnova na stretnutie</i>	20
8.10.2	<i>Priebeh stretnutia</i>	20
8.11	ZÁPISNICA STRETNUTIA č. 11.....	22
8.11.1	<i>Osnova na stretnutie</i>	22
8.11.2	<i>Priebeh stretnutia</i>	22
8.12	ZÁPISNICA STRETNUTIA č. 12.....	23
8.12.1	<i>Osnova na stretnutie</i>	23
8.12.2	<i>Priebeh stretnutia</i>	23
8.13	ZÁPISNICA STRETNUTIA č. 13.....	27
8.13.1	<i>Osnova na stretnutie</i>	27
8.13.2	<i>Priebeh stretnutia</i>	27
8.14	ZÁPISNICA STRETNUTIA č. 14.....	29
8.14.1	<i>Osnova na stretnutie</i>	29
8.14.2	<i>Priebeh stretnutia</i>	30
8.15	ZÁPISNICA STRETNUTIA č. 15.....	31
8.15.1	<i>Osnova na stretnutie</i>	31
8.15.2	<i>Priebeh stretnutia</i>	32
8.16	ZÁPISNICA STRETNUTIA č. 16.....	33
8.16.1	<i>Osnova na stretnutie</i>	33
8.16.2	<i>Priebeh stretnutia</i>	33
8.17	ZÁPISNICA STRETNUTIA č. 17.....	35
8.17.1	<i>Osnova na stretnutie</i>	35
8.17.2	<i>Priebeh stretnutia</i>	35
8.18	ZÁPISNICA STRETNUTIA č. 18.....	36
8.18.1	<i>Osnova na stretnutie</i>	36
8.18.2	<i>Priebeh stretnutia</i>	36
8.19	ZÁPISNICA STRETNUTIA č. 19.....	38
8.19.1	<i>Osnova na stretnutie</i>	38
8.19.2	<i>Priebeh stretnutia</i>	38
8.20	ZÁPISNICA STRETNUTIA č. 20.....	40

8.20.1	<i>Osnova na stretnutie</i>	40
8.20.2	<i>Priebeh stretnutia</i>	40
9	EXPORTY EVIDENCIE ÚLOH	1
9.1	PRVÝ ŠPRINT	1
9.2	DRUHÝ ŠPRINT	2
9.3	TRETÍ ŠPRINT.....	3
9.4	ŠTVRTÝ ŠPRINT.....	4
9.5	PIATY ŠPRINT	5
9.6	ŠIESTY ŠPRINT.....	6
9.7	SIEDMY ŠPRINT	7
9.8	ÔSMY ŠPRINT	8
9.9	DEVIATY ŠPRINT	9
9.10	DESIATY ŠPRINT	10
9.11	JEDENÁSTY ŠPRINT.....	11
9.12	DVANÁSTY ŠPRINT	12
10	POUŽÍVATELSKÉ TESTOVANIE	1
10.1	PRIEBEH POUŽÍVATELSKÉHO TESTOVANIA.....	1
10.1.1	<i>Prednosti aplikácie</i>	1
10.1.2	<i>Hlavné nedostatky aplikácie</i>	2
10.1.3	<i>Návrhy na vylepšenie aplikácie</i>	2
10.1.4	<i>Nedostatky zistené počas testovania</i>	2
10.2	ZHODNOTENIE TESTOVANIA.....	3
10.2.1	<i>Výsledky testovania</i>	3
11	ZÁVEREČNÁ SPRÁVA	1
11.1	ABSTRAKT	1
11.2	PREHLAD RIEŠENIA	3
11.3	REALIZÁCIA RIEŠENIA.....	5
11.4	ZHRNUTIE	8

1 Úvod

Tento dokument obsahuje podrobnú dokumentáciu k riadeniu tímového projektu. Tím podieľajúci sa na tomto projekte je zložený zo siedmich študentov a nesie názov Cats Can Code.

Cieľom nášho projektu je vytvoriť webovú službu, ktorá používateľom umožní čo najjednoduchšie plánovanie stretnutí za pomoci atraktívneho grafického používateľského rozhrania a bez nutnosti registrácie. Plánovanie stretnutia bude veľmi flexibilné. Zakladateľ udalosti môže definovať miesto a termín stretnutia, vytvoriť hlasovanie za miesto a termín alebo jednoducho nechať pozvaných, aby navrhli čo vyhovuje im. Termín udalosti môže byť konkrétny čas alebo časový úsek. Samozrejmosťou je aj možnosť zahlasovať za viac miest či termínov, pričom bude jasne viditeľné kto za čo hlasoval a ktorý termín, respektíve miesto má aktuálne najviac hlasov.

Druhá a tretia kapitola sa zaoberá kompetenciami členov nášho tímu a ich pridelenými manažérskymi úlohami.

Štvrtá kapitola obsahuje metodiky používané v riadení tímového projektu. Každá metodika tvorí jednu podkapitolu. Obsahom tejto kapitoly sú nasledujúce metodiky: metodika testovania, metodika evidencie úloh, metodika verziovania, metodika dokumentovania, metodika revízie kódu, metodika komunikácie, metodika písania kódu

Prílohami dokumentácie k riadeniu sú zápisnice zo všetkých stretnutí, ktoré opisujú ich priebeh vrátane zoznamu prítomných členov tímu, tiež aj jednotlivé metodiky a výstupy z nástroja JIRA.

Nakoľko súčasťou tohto dokumentu je aj manažment testovania, súčasťou prílohám tohto dokumentu je aj správa ktorá vznikla na základe hromadného testovania našej aplikácie. (Prílohy samotnej správy uč nie sú súčasťou dokumentu.)

Do príloh sme taktiež zaradili záverečnú správu o našom produkte, odovzdávanú v rámci súťaže TP CUP.

1.1 Podiel členov na dokumentácií

Nasledujúca tabuľka zobrazuje podiel členov tímu na tvorbe tohto dokumentu.

Úvod	Tomáš Sýkora
Zoznam kompetencií	Každý za seba
Role členov tímu	Vladimír Osvald
Manažment verziovania	Matúš Pikuliak
Manažment kvality	Michal Gajdoš
Manažment integrácie projektu	Lukáš Masár
Manažment dokumentovania	Vladimír Osvald
Manažment testovania	Michal Cihák
Manažment komunikácie	Pavol Michálek
Manažment rizík	Tomáš Sýkora
Sumarizácia šprintov	Tomáš Sýkora
Používané metodiky	Michal Gajdoš
Globálna retrospektíva	Lukáš Masár
Metodika testovania	Michal Cihák
Metodika evidencie úloh	Tomáš Sýkora
Metodika verziovania	Matúš Pikuliak
Metodika dokumentovania	Vladimír Osvald
Metodika revízie kódu	Lukáš Masár

Metodika komunikácie	Pavol Michálek
Metodika komentovania zdrojového kódu	Michal Gajdoš

Tabuľka 1. Podiel členov na dokumentácií

2 Role členov tímu

2.1 Manažérske úlohy

Rozdelenie manažérskych úloh v našom tíme je nasledovné:

- Matúš Pikuliak - manažér verziovania, front-end
- Michal Gajdoš - manažér kvality
- Lukáš Masár - manažér integrácie projektu, aplikačný server
- Vladimír Osvald - manažér dokumentovania
- Michal Cihák - manažér testovania, jadro aplikácie
- Pavol Michálek - manažér komunikácie, tímová stránka, front-end
- Tomáš Sýkora - manažér plánovania, rizík

2.2 Aplikácia manažérskych úloh

2.2.1 Manažment verziovania

Pre správu verzií kódu nášho projektu používame distribuovaný verziovací systém Git. Na webovej službe bitbucket sme si zriadili vzdialený repozitár, na ktorom udržiavame verzie našej webovej aplikácie v rozličných gitovských vetvách. Verziovanie v Git používame aj pri udržiavaní našej tímovej webovej stránky.

V súčasnosti pri počiatočnom vývoji používame jednu development vetvu. Dôležité milníky vo vývoji označujeme tagmi, pomocou ktorých dokážeme potom jednotlivé verzie odsledovať. V ďalších fázach sa chystáme vytvoriť ďalšie vetvy, staging a release. Tieto budú slúžiť pri vytváraní nasadzovaných releasov do prevádzky.

Takisto budeme podľa potreby vytvárať tzv. *features* vetvy pre skúšanie nových technológií či prebudovanie rozsiahlych častí systému. Ako nástroj pre prácu s repozitármi používame integráciu Git v IDE Rubymine, ktoré v tíme používame. Pre verziovanie sme si špecifikovali metodiku verziovania, podľa ktorej postupujeme, resp. pri niektorých častiach len budeme postupovať. Táto metodika sa nachádza v prílohe 7.5.

2.2.2 Manažment kvality

V rámci vývoja produktu je potrebné zabezpečiť kvalitu nielen výsledného produktu, ale aj jeho jednotlivých častí, počas jeho vývoja. Je nutné, aby jednotlivé úlohy boli vykonávané s vysokou mierou kvality od začiatku práce na nich, aby sa tak zabezpečilo, že náklady na ich neskoršie úpravy ako aj údržbu budú minimálne. Produkt, ktorý vyvíjame nie je na trhu nový. Existujúce riešenia však nie sú veľmi populárne. Je to možno práve z dôvodu ich nedostatočnej kvality. Preto je potrebné, aby náš produkt bol na vysokej úrovni, aby sme tak neodradili našich potenciálnych zákazníkov. Keďže samotný produkt bude vytváraný postupne, a jednotlivé moduly nebudú vždy kompletne, je potrebné dohliadnuť na kvalitu dodávanej verzie a teda jednotlivých častí.

Na vytvorenie kvalitného produktu je však potrebné aby aj jednotlivé procesy v tíme boli vykonávané s zodpovedne a s veľkou mierou kvality. Jednotliví členovia tímu sa preto majú možnosť vyjadriť ku stavu projektu v podstate kedykoľvek, avšak hlavne na tímovom stretnutí počas retrospektívy uplynulého šprintu. Práve na základe retrospektívy dokážeme resp. sme dokázali jednotlivé procesy v tíme značne zlepšiť. Dokázali sme zlepšiť hlavne komunikáciu a využívanie zdrojov, vďaka čomu je náš produkt resp. jeho jednotlivé prototypy vyvíjané kvalitne. Zároveň sa tá pridávanie ďalších modulov a funkcionalít stáva jednoduchšími.

2.2.3 Manažment integrácie projektu

Na vývoji produktu sa podieľajú viacerí developeri, ktorí pracujú na svojich lokálnych strojoch. Výstupy ich práce sa koordinujú pomocou systému Git. Pre zabezpečenie aktuálnej verzie produktu dostupnej cez internet, beží na serveri skript, ktorý pravidelne každých 5 minút aktualizuje verziu produktu na serveri. Skript sťahuje najnovšiu verziu produktu z git repozitáru pre development aj release vetvu. Zároveň, ak je to potrebné, inštaluje doplňujúce gems a vykonáva úpravy v databáze. Obdobným spôsobom fungujú aj skripty, ktoré zabezpečujú aktuálnosť produkčného prostredia, teda release vetvy, aj tímovej stránky.

2.2.4 Manažment dokumentovania

V rámci tímového projektu vnikajú rôzne dokumentácie. Úlohou manažmentu dokumentovania bolo určiť nástroje, pomocou ktorých sa budú dokumentácie vytvárať a stanoviť miesto, kde sa budú dokumenty nachádzať. Vznikli postupy, ktoré je potrebné dodržiavať pri písaní dokumentov a bola jasne stanovená výsledná forma dokumentácie. Podrobný proces písania dokumentácie k inžinierskemu dielu bol spísaný v príslušnej metodike.

2.2.5 Manažment testovania

Pri vyvíjaní softvéru v tímovom projekte je potrebné zachovať integritu vyvíjaného softvéru počas celého vývoja a zabezpečiť aby jednotlivé časti softvéru vyvíjané rôznymi vývojármi správne fungovali pri vytváraní nových verzií. Preto je vhodné vyvíjaný softvér pravidelne testovať a vytvárať potrebné testy. Úlohou manažmentu testovania bolo určiť vhodné nástroje na testovanie a čo a kedy sa bude primárne testovať. Na základe analýzy manažmentom testovania a po dohode s tímom boli zvolené testovacie nástroje, ktoré sú spísané v príslušajúcej metodike k testovaniu nachádzajúcej sa v prílohe 7.3. V tejto metodike sú ďalej popísané spôsoby vytvárania základných testov vhodných pre náš projekt. Okrem testovania pomocou takto vytvorených testov prebiehajú testy v našom tíme aj samotnými členmi - v tomto prípade testerami. A to vždy pri každej splnenej úlohe, ktorá je posunutá na prehliadku. Ďalšie testovanie prebieha vždy aj pred každým koncom šprintu, kedy sa prezentuje momentálna verzia aplikácie (demo), kde sa kontroluje či demo spĺňa každú požadovanú funkcionálnu.

2.2.6 Manažment komunikácie

Bolo potrebné zabezpečiť, aby všetci členovia tímu komunikovali pomocou jednotných komunikačných nástrojov. Po analýze a sumarizácii vhodných nástrojov bolo tímu predložených niekoľko návrhov. Pod vedením manažmentu komunikácie, ktorý dohliadal na správny počet komunikačných kanálov pre všetky situácie, boli v tíme schválené nasledujúce komunikačné nástroje:

- **HipChat** - štandardná online textová komunikácia a rýchle reporty o posledných aktivitách v ostatných používaných nástrojoch. Manažment komunikácie dohliada na dodržiavanie pravidiel stanovených v metodike a stanovuje a usmerňuje nové témy.
- **Google Hangouts** - hovory a videohovory, bez zásahu manažmentu
- **Jira** - evidencia úloh. V tomto nástroji je potrebný dohľad manažmentu komunikácie nad tvorbou a správou zadaných úloh, aby boli všetkým členom zrozumiteľné a jasné.
- **Google Drive** - zdieľanie súborov. Manažment zadal pravidlá práce s týmto úložiskom.
- **Webové sídlo tímu** - externá komunikácia. Manažment schválil formu webovej prezentácie tímu, a dohliada na aktualizovanie obsahu webu.

2.2.7 Manažment plánovania

Táto oblasť manažmentu bola zabezpečená pomocou nástroja JIRA. Pred začiatkom každého šprintu si definujeme úlohy, ktoré chceme stihnúť do konca nadchádzajúceho šprintu. Tieto úlohy následne ohodnotíme relevantným počtom story points a to tak, že každý člen estimuje daný task podľa seba. Potom podľa strednej hodnoty každý člen, ktorý oestimoval vyššie, tak odôvodní svoj počet a tak isto aj člen z nižším počtom ako stredná hodnota ohodnotí tento svoj počet. Následne sa hodnotí znova a takto to iteruje až kým sa nezhodneme. Keď máme oestimované úlohy, tak ich pridávame do nástroja JIRA podľa metodiky evidencie úloh, ktorá je uvedená v 7.2. V metodike je uvedený aj postup pomenovania úlohy.

Keď už máme úlohy uložené v nástroji, vytvorí sa šprint, presunú sa do neho úlohy na ktorých sme sa dohodli a následne sa spustí šprint. Na stretnutí sme si spoločne rozdelili úlohy a vždy mal niekto čo robiť. Každý člen si potom priradí v nástroji JIRA svoju úlohu, a začne na nej pracovať.

Ak už je task hotový, presunie sa do stĺpcu *Review*. Potom daný task iný člen tímu skontroluje a zvaliduje. Táto kontrola sa robí podľa metodiky code review, ktorá je uvedená v 7.4. Ak je v úlohe chyba, vytvorí sa relevantná úloha typu *Bug*, taktiež podľa metodiky 7.2.

Na konci šprintu sa potom jednotlivé vypracované úlohy prezentujú vlastníčkovi produktu, ktorý zhodnotí náš pokrok, spravíme retrospektívu za šprint a potom sa začína príprava na ďalší šprint.

2.2.8 Manažment rizík

Počas vývoja produktu vznikajú rôzne riziká a preto je ich identifikácia a následne aj zamedzenie výskytu dôležité. Každé riziko ma definovanú úroveň rozsahu škôd:

- Nízky - riziko je minimálne, netreba vynaložiť veľkú námahu na jeho vyriešenie,
- Stredný - riziko je stredné, je potrebné ho riešiť,
- Vysoký - riziko je vysoké, je potrebné ho okamžite riešiť.

Za doterajší priebeh vzniklo zopár rizík. Napríklad viaznuca komunikácia medzi členmi tímu, čo je vysoké riziko, keďže pre správny postup vývoja je komunikácia kritická. Toto riziko sme vyriešili použitím nástroja HipChat, ktorý je spomenutý vyššie.

Ďalšie riziko vzniklo pri zistení, že dvaja kolegovia odchádzajú na Erasmus. Ako riešenie tohto problému sme navrhli zopár kolaboračných nástrojov, potrebných pri vývoji, ďalší nástroj na zdieľanie obrazovky a videohovory. Toto riziko je taktiež vysoké, keďže by výrazne ovplyvnilo rýchlosť vývoja produktu.

3 Sumarizácia šprintov

V kapitole sa nachádzajú sumarizácie šprintov.

3.1 Prvý šprint

Navrhli sme architektúru aplikácie a databázový model a vytvorili sme jadro aplikácie. Jednotlivé úlohy vypracované v rámci šprintu je možné vidieť v prílohe 9.1.

3.2 Druhý šprint

Vytvorili sme prvú verziu grafického používateľského rozhrania. Jednotlivé úlohy vypracované v rámci šprintu je možné vidieť v prílohe 9.2.

3.3 Tretí šprint

V šprinte sa kompletizovali dokumenty potrebné pre prvý kontrolný bod. Jednotlivé úlohy vypracované v rámci šprintu je možné vidieť v prílohe 9.3.

3.4 Štvrtý šprint

V tomto šprinte došlo k “zahodeniu” časti doterajšieho produktu, hlavne v stránke dizajnu, do ktorého bola implementovaná časová os. Taktiež prebehlo rozsiahle refaktorovanie. Jednotlivé úlohy je možné vidieť v prílohe 9.4.

3.5 Piaty šprint

Ako je možné vidieť v prílohe 9.5, nastal posun v oblasti týkajúcej sa samotných používateľov nášho produktu. Bola vytvorená profilová stránka pre používateľov, ako aj implementované automatické notifikácie posielane na mailovú adresu používateľa.

3.6 Šiesty šprint

Išlo o šprint na prelome semestrov. Zamerali sme sa pritom na odladenie aplikácie a teda odstránenie hlavných nedostatkov. Jednotlivé bug-y je vidieť v prílohe 9.6.

3.7 Siedmy šprint

Bol vytvorený nový koncept budúceho smerovania našej aplikácie, ktorým je sociálna sieť. Vďaka tomu vznikli nové úlohy, ktoré sme už predtým svojím spôsobom riešili. Tieto úlohy sú súčasťou prílohy 9.7.

3.8 Ôsmy šprint

Ako je možné vidieť z prílohy 9.7-9.8, pokračovali sme v pridávaní nových funkcionalít, ako aj zdokonaľovaní a ladení tých starých. Zároveň sme začali spolupracovať s kolegami z Ekonomickej univerzity v Bratislave

3.9 Deviaty šprint

Prebehlo rozsiahly update a redizajn používateľského rozhrania, ktoré dostalo okrem iného novú, lepšiu farebnú schému. Okrem toho sme pracovali na úlohách viditeľných v prílohe 9.9.

3.10 Desiaty šprint

Prebehlo prvé väčšie testovanie, na základe čoho bolo vytvorených niekoľko nových úloh – vid' príloha 9.10. Taktiež bol navrhnutý nový komponent, na získavania spätnej väzby.

3.11 Jedenásty šprint

Tento šprint bol značne ovplyvnený študentskou vedeckou konferenciou IIT SRC, na ktorej sme v rámci súťaže TP CUP prezentovali našu aplikáciu. Tasky tohto šprintu sa nachádzajú v prílohe 9.11.

3.12 Dvanásť šprint

V tomto šprinte sme sa zamerali hlavne na odladenie implementovanej funkcionality. Prebehlo ďalšie hromadné testovanie, pričom z neho vyplývajúce zistenia sme spracovali a plánujeme aj na ich základe pokračovať vo vývoji. Úlohy zobrazuje príloha 9.12.

4 Používané metodiky

V tejto časti dokumentu sú stručne popísané jednotlivé metodiky, ktorými sa náš tím riadi:

- Metodika testovania
- Metodika evidencie úloh
- Metodika verziovania
- Metodika dokumentovania
- Metodika revízie kódu
- Metodika komunikácie
- Metodika komentovania kódu

4.1 Metodika testovania

Metodika určuje testovacie nástroje, ktoré sa používajú v rámci nášho projektu. Ďalej obsahuje popísané spôsoby vytvárania základných testov vhodných pre náš projekt.

Kompletná metodika je priložená v prílohe 7.3.

4.2 Metodika evidencie úloh

Metodika definuje prácu s úlohami v nástroji JIRA, aké typy úloh používame, ktorý typ a kedy použijeme a postup pri vytváraní úlohy.

Kompletná metodika je priložená v prílohe 7.2.

4.3 Metodika verziovania

Metodika pojednáva o procesoch pri verziovaní kódu a súborov všeobecne pri vyvíjaní implementácie nášho projektu. Procesy slúžia programátorom pri písaní kódu a jeho synchronizácii v rámci tímu.

Kompletná metodika je priložená v prílohe 7.5.

4.4 Metodika dokumentovania

Metodika obsahuje pracovné procesy a postupy potrebné pri písaní dokumentácie k inžinierskemu dielu pomocou nástrojov Wiki a Microsoft Word. Obsahuje pokyny pre editáciu a finalizáciu dokumentu.

Kompletná metodika je priložená v prílohe 7.7.

4.5 Metodika revízie kódu

Účelom tejto metodiky je stanoviť postup, ktorý dokáže zachytiť chyby v kóde pred tým, ako sa dostanú k používateľovi. Je to základná činnosť vykonávaná za účelom zabezpečenia kvality kódu. Jej zanedbanie môže viesť k zbytočne zvýšeným nákladom v ďalších etapách projektu.

Kompletná metodika je priložená v prílohe 7.4.

4.6 Metodika komunikácie

Metodika komunikácie v tíme popisuje postupy používané pri riadení nášho projektu. Ďalej definuje použité nástroje a spôsoby ich použitia v konkrétnych prípadoch.

Kompletná metodika je priložená v prílohe 7.6.

4.7 Metodika komentovania kódu

Táto metodika obsahuje postup a spôsob písania komentárov zdrojového kódu, vo vývojovom prostredí RubyMine, ako aj následné vytvorenie dokumentácie pomocou aplikácie Rdoc.

Kompletná metodika je priložená s prílohe 7.1.

5 Globálna retrospektívna

5.1 Po zimnom semestri

V rámci nášho projektu sa snažíme pracovať čo najefektívnejšie. Ako tím sme však vznikli len pred pár týždňami, a preto neustále zlepšujeme naše postupy, aby tak vyhovovali schopnostiam a znalostiam jednotlivcov a aby sme ako celok dosahovali čo možno najlepšie výsledky. Po každom šprinte preto máme retrospektívu. Práve na základe nich sa nám výrazne podarilo zlepšiť prípadne odstrániť veľké množstvo nedostatkov. Aj napriek tomu je stále možnosť našu prácu v tíme vykonávať efektívnejšie aby naše výsledky boli ešte lepšie.

Začiatky bývajú ťažké, a teda ani ten náš nebol jednoduchý. Problémové boli hlavne prvé šprinty. Jedným z dôvodov bola potreba zvyknúť si na scrum. Ďalším a možno závažnejším problémom boli samotné úlohy, ktoré sme si definovali. Keďže sme začínali na úplne novom projekte z hľadiska vývoja (písania kódu) nebolo vždy možné pracovať paralelne. Práve na základe retrospektív po šprintoch, sa nám podarilo eliminovať tento problém pomocou zadefinovania úloh s interným označením “investigate”, aby tak vznikli aj neimplementačné úlohy a tak sme mohli pracovať paralelne na viacerých úlohách. Pri paralelnej práci vznikali v prvých šprintoch problémy pri zlučovaní lokálnych verzií, ktoré sme ku koncu semestra pociťovali stále menej.

Ďalším problémom je pre nás komunikácia. Aj tá sa každým šprintom zlepšuje, avšak stále nie je na úrovni, ktorú by mohla dosiahnuť. Tento problém si však uvedomujeme, a preto sa snažíme pracovať na jeho odstránení. Ide o informovanie ostatných členov tímu o stave resp. aktuálnych činnostiach. Keďže ako tím (čiže všetci) sa stretávame 1-2 krát týždenne, komunikáciu v tíme riešime v prvom rade zodpovedným písaním komentárov k daným úlohám v nástroji JIRA, aby tak bol jasný pokrok na jednotlivých úlohách. Čo sa týka ostatnej komunikácie, tá prebieha v nástroji HipChat.

Od začiatku semestra sme sa určite posunuli správnym smerom a procesy a postupy sa nám v tíme reálne podarilo (v niektorých smeroch niekoľkonásobne) zlepšiť. Rezervy však stále máme vo využívaní zdrojov - nie všetci členovia tímu pracujú tak, aby plne využili svoj potenciál (aj keď by možno chceli). Tento fakt je spôsobený aktivitami, ktoré sú často spojené s

inými študijnými povinnosťami. Jednotlivým členom tímu tak neostáva dostatočné množstvo času na to, aby prejavili svoje schopnosti.

Do budúca by sme teda chceli zlepšiť komunikáciu a teda aj spôsob a frekvenciu stretnutí. Nakoľko sa v našom tíme nachádzajú dvaja študenti, ktorí idú na výmenný pobyt (Erasmus) zmena resp. úprava komunikácie medzi členmi tímu sa bude musieť zmeniť - bude nutné rozšíriť online komunikáciu, aby tak nebola ohrozená výkonnosť nášho tímu.

5.2 Po letnom semestri

V letnom semestri sme pokračovali v prácach na našej webovej aplikácii Happen. Podobne ako v zimnom semestri, aj v tom letnom sme sa stretli s niekoľkými problémami, novými situáciami, naučili sme sa nové veci, prístupy k riešeniu problémov a posunuli náš projekt o veľký kus dopredu.

Medzi hlavné problémy, s ktorými sme sa stretli, by sme zaradili pretrvávajúci problém so vzájomnou komunikáciou v našom tíme, ktorý sme si preniesli ešte zo zimného semestra. Navyše v letnom semestri tento problém ešte zvýraznil fakt, že dvaja členovia nášho tímu absolvovali semester spoločne na študijnom pobyte erasmus vo Švédsku. Nakoľko mali úplne iným spôsobom organizovaný svoj školský harmonogram, spôsobilo to akoby rozdelenie nášho tímu na dva menšie tímy. Boli s tým spojené komplikácie pri organizovaní spoločných tímových stretnutí, ktoré sme organizovali s pomocou video-hovorov. Tu sme občas narážali na problémy s internetovou konektivitou, ktorá spôsobovala výpadky v našej komunikácii do Švédska. Nakoniec sme sa však s týmito problémami vysporiadali, a určite sme si z toho odniesli veľa skúsenosti, aj ohľadne toho ako pracujú tímy distribuované vo viacerých mestách, či krajinách. Naša komunikácia sa teda postupne zlepšovala, hlavne ku koncu semestra a s blížiacou sa IIT SRC konferenciou sme s ňou už prakticky nemali problémy.

Konferenciu IIT SRC hodnotíme ako veľmi pozitívnu a kvalitnú skúsenosť, keďže nám dala príležitosť prezentovať náš produkt priamo ľuďom z branže, čo nám bez pochyby dalo do budúca cenné skúsenosti. Rovnako nás potešila aj spätná väzba na náš produkt, ktorú sme

dostávali na IIT SRC, čo nás povzbudilo v smerovaní nášho produktu a v ďalšej práci na jeho napredovaní.

Ďalším aspektom, na ktorom sme museli oproti zimnému semestru viac zapracovať bolo využívanie zdrojov. Aj v tomto smere sme spravili oproti zimnému semestru určité pokroky, ale stále to nebolo podľa našich predstáv, čo pripisujeme hlavne mnohým iným povinnostiam, ktoré mali členovia tímu individuálne, a tak sme sa bohužiaľ nemohli projektu venovať ako tím naplno tak, ako by sme si to predstavovali. Napriek tomu nás veľmi potešila pozitívna reakcia na produkt, ktorý sme vystavili testovaniu našim potencionálnym používateľom, ako aj výsledky prieskumu, s ktorým nám pomohli kolegovia s Ekonomickej Univerzity. To nás povzbudilo do ďalšej práce na našom projekte.

6 Zoznam kompetencií tímu

Michal Cihák

Záujmová oblasť: moderné webové technológie, mobilné technológie (ios), grafika

Znalosti: Ruby (Ruby on Rails), HTML5, CSS3, Sass, JavaScript, D3.js, SOAP, webové služby, OpenGL4.x

Doterajšie skúsenosti: Vypracovanie BP ako Ruby on Rails webovej aplikácie, kde som získal bohaté skúsenosti s komunikáciou s webovými službami s následnou vizualizáciou týchto dát klientovi v prehliadači. Tvorba webstránok - získanie skúseností s najnovšími CSS3 technológiami, jazykom Sass a JavaScriptom.

Michal Gajdoš

Záujmová oblasť: mobilné technológie, umelá inteligencia, webové technológie

Znalosti: C# - .NET [MVC], Java, MySQL, PHP, HTML5, CSS3, SOAP, jQuery

Doterajšie skúsenosti: Bakalárska práca, ktorej výsledkom je program v jazyku C# (WPF Aplikácia + MVVM), používaný dopravným podnikom mesta Bratislava. Momentálne pracujem na pozícií Web Developera vo firme Lomtec.

Lukáš Masár

Záujmová oblasť: umelá inteligencia, distribuované systémy, webové technológie, sieťové služby

Znalosti: Erlang, Java, HTML, CSS, PHP, MySQL

Doterajšie skúsenosti: Bakalárska práca, ktorej témou bol distribuovaný rámec pre tvorbu škálovateľných webových aplikácií v jazyku Erlang. Momentálne pracujem v spoločnosti NASES na pozícií špecialistu na prevádzku sieťových služieb.

Pavol Michálek

záujmová oblasť: rozsiahle integrované webové systémy

Znalosti: PHP, HTML5, CSS3, Less, JavaScript (jQuery), D3.js, SOA technológie, Java,

Doterajšie skúsenosti: 1.5 roku web frontend developer a biznis analytik na projekte Multichannel pre banku VÚB (internet/mobile banking), 0.5 roku web developer (e-shopy, backend, administrácia), bakalárka - interaktívna webová vizualizácia

Vladimír Osvald

Zájemová oblast: webové technologie, mobilné technologie (Android), aplikace pro MS Windows

Znalosti: HTML5, CSS3, PHP, C++, C#, Java, MySQL

Doterajšie skúsenosti: Vytvorenie niekoľkých jednoduchších webových stránok, vypracovanie bakalárskej práce v jazyku C#, kde som pracoval najmä s databázou MySQL, pričom som riešil komunikáciu medzi klientom a serverom, čo je využiteľné aj vo webových systémoch a pri mobilných technológiách.

Matúš Pikuliak

zájemová oblast: webové technologie, umelá inteligencia

Znalosti: PHP, CodeIgniter FW, HTML5, CSS3, LESS, JavaScript, MySQL

Doterajšie skúsenosti: Mám skúsenosti s tvorením webových stránok nad MVC frameworkom CodeIgniter, ktoré spočívali v kompletnom spracovaní od grafického návrhu, cez implementáciu až po nasadenie na server. V jazyku PHP som programoval aj svoju bakalársku prácu týkajúcu sa evolučných algoritmov.

Tomáš Sýkora

zájemová oblast: databázové technologie, webové technologie, siete, distribuované systémy

Znalosti: Java, MySQL, C, XML, UML, HTML, CSS

Doterajšie skúsenosti: Paralelná práca s tímovým projektom na prototyp 3D UML, kde som vytvoril aplikáciu serializácie UML diagramov kvôli kompatibilitate s inými CASE nástrojmi (výsledok BP). Ročná skúsenosť na pozícii IT Support v advokátskej firme Čechová & Partners.

7 Metodiky

Táto príloha obsahuje vypracované metodiky od jednotlivých členov tímu.

7.1 Komentovanie zdrojového kódu

Autor: Michal Gajdoš

Nasledujúci materiál obsahuje postup a spôsob písania komentárov zdrojového kódu, vo vývojovom prostredí RubyMine, ako aj následné vytvorenie dokumentácie pomocou aplikácie Rdoc.

7.1.1 Povolenie komentárov slúžiacich na dokumentáciu

Pre správne fungovanie komentárov je potrebné povoliť dokumentačné komentáre zaškrtnutím políčka *“Insert documentation comment stub”*. K políčku sa možno dostať pomocou *“File”>“Settings”>“Editor”>“Smart Keys”*, v časti *“Enter”*.

7.1.2 Všeobecné pokyny k písaniu komentárov

Pri komentovaní kódu je potrebné dodržiavať nasledovné všeobecné zásady:

- komentáre sú písané v anglickom jazyku,
- komentáre sú stručné a výstižné
- používať komentáre len na miestach, kde sú potrebné, aby sa zabránilo neprehľadnosti kódu.

Komentáre je nutné použiť na komentovanie každej:

- triedy,
- metódy a jej vstupných parametrov,
- zložitejšej prípadne nejasnej časti kódu.

7.1.3 Komentovanie v Ruby

Na komentovanie kódu v Ruby sa používa jednoriadková notácia, začínajúca znakom “#” a medzerou. Na takýto zápis možno použiť klávesovú skratku **Ctrl** + “/”, ktorá celý riadok, v ktorom sa nachádza kurzor, označí ako komentár. (Komentár je automaticky ukončený na konci konkrétneho riadka.)

V prípade, že komentár presiahne dĺžku jedného riadka, respektíve sa nachádza vo viacerých riadkoch, začiatok každého riadku je označený “#”.

7.1.4 Komentovanie v JavaScript

Pri komentovaní zdrojového kódu písaného v jazyku JavaScript (.js tried) je možné použiť dva typy komentárov. V prípade jednoriadkového komentáru je možné použiť znaky “//”, ktoré celý riadok resp. zvyšok riadka, nachádzajúci sa za týmito znakmi, označia ako komentár. Takýto komentár je možné použiť aj v prípade že je potrebné označiť ako komentár viacero riadkov. Označenie “//” je však potrebné uviesť na začiatku každého riadku.

V prípade viacriadkového komentáru je taktiež použiť notáciu “/*” na začiatku a “*/” na konci komentáru. Takéto označenie vytvorí komentár z obsahu nachádzajúceho sa medzi uvedenými ohraničeniami. Z dôvodu lepšej prehľadnosti je vhodné každý nový riadok (okrem prvého a posledného) komentáru označiť symbolom “/*”. Takýto komentár je taktiež možné použiť v prípade že je žiadúce označiť za komentár len časť riadku.

V ďalších častiach tejto metodiky sa uvádza postup pri komentovaní zdrojového kódu v jazyku Ruby. Rovnaké pravidlá však platia aj pri komentovaní kódu v jazyku JavaScript, pričom je potrebná len zámena notácií označujúcich komentár.

7.1.5 Komentovanie tried

Každá trieda obsahuje komentár, ktorý danú metódu opisuje. Takýto komentár sa nachádza respektíve začína na prvom riadku konkrétnej metódy. Tento komentár je potrebné pridať pri vytvorení samotnej triedy. V prípade správne nastaveného prostredia RubyMine, je takýto

komentár pridaný automaticky pri vytvorení. Pri manuálnom pridaný komentáru je potrebné zachovať jeho formát – meno člena tímu, ktorý triedu vytvoril, ako aj dátum vytvorenia.

Príklad komentáru v *.js* súboroch:

```
/**  
 * Created by Piko on 29. 11. 2014.  
 */
```

7.1.6 Komentovanie metód

Rovnako ako všetky triedy, tak aj všetky metódy sú uvádzané komentárom. Tento komentár sa nachádza resp. končí o riadok vyššie, ako je riadok, v ktorom je daná metóda definovaná (riadok, v ktorom sa daná metóda začína). Začiatkom respektíve prvou časťou komentáru je opis metódy, ktorej komentár prislúcha (ktorá sa nachádza pod komentárom). Pod týmto opisom sa nachádza prázdny zakomentovaný riadok (riadok obsahujúci len znak “#” a **medzeru** na začiatku).

V prípade, že sa v komentári uvádzajú vstupné premenné funkcie (ktoré je nutné uviesť, pokiaľ daná metóda nejaké vstupné parametre obsahuje), sú tieto uvedené samostatne v riadkoch, pričom názov premennej je uvedený medzi dvomi znakmi “+”. Takto uvedenú premennú zároveň predchádza znak “*” a **medzera** (takýto riadok sa vo vygenerovanej dokumentácii stáva položkou nečíslovanom zozname, so zvýraznením názvu premennej). Blok premenných je zároveň uvedený zakomentovaným riadkom, obsahujúcim “=== **Parameters**” (slúži na vytvorenie nadpisu tretej úrovne, v dokumentácii).

Pokiaľ sa v komentári nachádza aj návratová hodnota metódy (ktorú je nutné uviesť v prípade, že metóda návratovú hodnotu obsahuje), je uvedená za prípadnými parametrami respektíve na konci komentára samotnej metódy. Formát takéhoto komentára je “# **+return+**” a **popis návratovej hodnoty**.

Príklad:

```
# === Parameters  
# * +a+ - first number  
# * +b+ - second number  
#  
# +return+ -sum of numbers
```

Takýto komentár je potrebné pridať vždy pri vytvorení konkrétnej metódy, pričom prvotný tvar komentáru sa môže v závislosti od aktualizácie metódy zmeniť. Výnimkou, kedy komentár nie je potrebný sú metódy písané (pomenované) podľa konvencie (napr. metódy controllerov typu *new* alebo *create* v jazyku Ruby)

7.1.7 Odstránenie komentáru z dokumentácie

V prípade, že komentár nemá byť súčasťou dokumentácie – komentáre typu TODO a FIXME, ako aj poznámky autora slúžiace pre jeho vlastné účely, prípadne časti kódu, ktoré chce vývojár z určitého dôvodu zachovať, obsahuje tento komentár špeciálne označenie. Toto označenie pozostáva zo symbolov “#--” a “#++”. Obe tieto skupiny znakov sa musia nachádzať na začiatku riadka, pričom ich poradie je taktiež zachované. “#--” sa musí v kóde nachádzať vyššie, ako “#++”. Medzi dva takto označené riadky, je možné pridať ľubovoľný počet riadkov komentárov, pričom tieto, sú pri automatickej tvorbe dokumentácie ignorované.

Príklad:

```
# This method returns sum of two numbers.  
#--  
# This comment will not be in the documentation.  
# And this one as well  
#++  
# Works only with numbers greater than 0
```

7.1.8 Komentovanie premenných

V prípade, že sa v zdrojovom kóde nachádza premenná, ktorej význam a obsah nie je z jej názvu jasný, sa komentár k takejto premennej nachádza v riadku, v ktorom bola premenná prvý krát uvedená, pričom komentár je vo formáte “# ” a popis premennej. Pokiaľ by sa na danom riadku nachádzalo viacero premenných, ktoré je potrebné komentovať, pred popisom premennej je uvedený jej názov, a jednotlivé premenné s popismi sú od seba oddelené čiarkou.

7.1.9 Ostatné typy komentárov

V prípade potreby je nutné, aby autor použil špecifické typy komentárov:

- “TODO”,
- “FIXME”.

Tieto komentáre sú vývojovým prostredím RubyMine implicitne podporované. Implicitná je taktiež podpora písania kľúčových slov bez rozlíšenia veľkých a malých písmen.

7.1.10 “TODO” komentáre

V prípade, že autor pri písaní kódu neimplementuje celú požadovanú respektíve potrebnú funkcionálnosť, prípadne niektorý z autorov zistí túto skutočnosť, miesto v kóde, na ktoré treba funkcionálnosť doplniť je potrebné označiť. Na označenie chýbajúcej časti použije komentár typu “TODO”. Komentár začína znakom “#”, za ktorým nasleduje **medzera** a kľúčové slovo “**TODO**”. Nasleduje krátky popis chýbajúcej funkcionality, ktorú je potrebné dorobiť.

Príklad:

```
# TODO: implement date check (starting_time<ending_time?)
```

7.1.11 “FIXME” komentáre

V prípade, že autor pri písaní kódu zistí, že program nefunguje dostatočne, prípadne niektorý z autorov zistí túto skutočnosť, miesto v kóde, na ktorom je potrebné vykonať zmeny je potrebné označiť. Na označenie chýbajúcej časti použije komentár typu “FIXME”. Komentár začína znakom “#”, za ktorým nasleduje medzera a kľúčové slovo “**FIXME**”. Nasleduje samotný stručný popis chyby prípadne udalosti, pri ktorej chyba nastala.

Príklad:

```
# FIXME: fails if starting_date is 29th October
```

7.1.12 Vygenerovanie dokumentácie

Dokumentácia sa generuje pomocou aplikácie RDoc.

Predpoklady na vygenerovanie dokumentácie

Na vygenerovanie dokumentácie je potrebný nainštalovaný RDoc gem. Tento je možné nainštalovať pomocou príkazu “gem install rdoc”, ktorý možno zadať do konzoly vo vývojovom prostredí RubyMine.

Vygenerovanie dokumentácie

Samotné vygenerovanie dokumentácie sa uskutočňuje pomocou príkazu “rdoc”. Tento v adresári projektu vygeneruje požadovanú dokumentáciu. Dokumentáciu je možné zobrazit’ vo webovom prehliadači otvorením súboru `.\doc\index.html`.

7.2 Evidencia úloh v nástroji JIRA

Autor: Tomáš Sýkora

Táto časť definuje prácu s úlohami v nástroji JIRA, aké typy úloh používame, ktorý typ a kedy použijeme a postup pri vytváraní úlohy. Pre všetky úlohy je potrebné byť prihlásený pomocou AIS loginu v nástroji JIRA a byť na domovskej stránke.

7.2.1 Roly a zodpovednosti účastníkov

Jednotlivé roly a zodpovednosti účastníkov sú uvedené v tabuľke 2.1.

Rola	Zodpovednosť
Vlastník produktu	Zástupca zákazníka Schválenie zadania úlohy Schválenie vypracovania úlohy Schválenie prezentácie úlohy
Scrum master	Sledovanie stavu úloh v šprinte Report stavu úloh vlastníkovi
Riešiteľ	Riešenie priradenej úlohy Odovzdanie vypracovanej úlohy na testovanie
Tester	Testovanie vypracovanej úlohy Potvrdenie správnosti úlohy Vytvorenie úlohy typu Bug, ak boli zistené chyby vo vypracovaní

Tabuľka 2.1 Roly a zodpovednosti účastníkov

7.2.2 Typy úloh

V našom projekte používame dva typy úloh a to:

- Task
- Bug

7.2.3 Pomenovanie úloh

Názov úlohy pozostáva z počiatočného tagu a zo stručného a výstižného názvu. Tagy sú uzavreté v hranatých zátvorkách. Používame nižšie uvedené tagy:

- [V] – View, úloha týkajúca sa zobrazenia na klientovi,
- [C] – Controller, úloha týkajúca sa serverovej časti produktu,
- [C/V] – Komplexná úloha, ktorá zahŕňa prácu aj na klientskej, aj na serverovej časti produktu,
- [B] – Bug, slúži na identifikáciu chyby v úlohe,
- [?] – Výskumná úloha, slúži na preskúmanie nástrojov alebo technológií, či sú vhodné na použitie vo vývoji produktu,
- [!] – Dôležitá úloha, napríklad odovzdanie časti produktu, dokumentácie.

Potom sú pomocou týchto tagov pomenované konkrétne úlohy. Príklady pomenovania úloh:

- Pre view, controller, komplexnú, výskumnú a dôležitú úlohu je formát názvu rovnaký – “[Tag] (stručný názov úlohy)”
- Pre bug sa názov líši – “[B] Fix bug in (názov pôvodnej úlohy)”.

7.2.4 Vytvorenie úlohy typu Task

Tento typ používame vtedy, ak sme sa dohodli na jednotlivých úlohách, ktoré chceme robiť v ďalšom šprinte. Vytvorené úlohy idú do backlogu. Pre vytvorenie postupujeme nasledovne:

1. Kliknúť na tlačidlo Create na hornej lište,
Alternatívne: Stlačiť klávesu c,
2. Zvoliť projekt CatchMe,
3. Issue type je v tomto prípade Task,
4. Do políčka Summary vpísať stručný a výstižný názov úlohy podľa návodu uvedeného v 2.3 Pomenovanie úloh,
5. Prioritu nezádávame,
6. Due date primárne nezádávať, podľa potreby vložiť,
7. Assignee nastaviť na Unassigned, ak je nastavené inak
8. Do poľa Description vpísať podrobný popis úlohy a akceptačné kritériá,

9. Časové odhady nevyplňat',
10. Pri políčku Attachment kliknúť na Choose Files a načítať relevantný dokument, obrázok,
11. Labels nevyplňat',
12. Do políčka Units vložiť odhad, ktorý bol odhadnutý na stretnutí členmi tímu,
13. Položky PercentDone, DueTime, Epic Link a Sprint nevyplňat',
14. Kliknúť na tlačidlo Create.

Alternatívne: Zrušiť kliknutím na Cancel.

7.2.5 Vytvorenie úlohy typu Bug

Používame vtedy, keď tester našiel pri testovaní chybu v úlohe. Pre vytvorenie takéhoto typu úlohy je postup nasledovný:

1. Kliknúť na tlačidlo Create na hornej lište,
Alternatívne: Stlačiť klávesu c,
2. Zvoliť projekt CatchMe,
3. Issue type je v tomto prípade Bug,
4. Do políčka Summary vpísať stručný a výstižný názov úlohy podľa návodu uvedeného v 2.3 Pomenovanie úloh pre typ úlohy Bug,
5. Prioritu nezadávať,
6. Due date primárne nezadávať, podľa potreby vložiť,
7. Assignee nastaviť na riešiteľa pôvodnej úlohy,
8. Do poľa Description vpísať podrobný popis chyby a akceptačné kritériá,
9. Časové odhady nevyplňat',
10. Pri políčku Attachment kliknúť na Choose Files a načítať relevantný dokument, obrázok,
11. Položky Labels, Units, PercentDone, DueTime, Epic Link a Sprint nevyplňat',
12. Kliknúť na tlačidlo Create.

Alternatívne: Zrušiť kliknutím na Cancel.

7.2.6 Zmena názvu úlohy

Zmena názvu úlohy môže byť vhodná pri neúplnom, nepresnom zadaní úlohy alebo aj pri preklepe.

1. Kliknúť na Agile > Scrum board,
2. V zobrazení šprintu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 2)
Alternatívne: V zobrazení backlogu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 1)
3. Na pravom boku sa zobrazí panel, ktorý obsahuje atribúty úlohy, tu kliknúť na názov úlohy,
Alternatívne: Stlačiť klávesu e > kliknúť do políčka za položku Summary,
4. Upraviť existujúci názov na požadovaný názov,
5. Stlačiť klávesu Enter pre potvrdenie zmeny názvu.
Alternatívne: Stlačiť klávesu Esc pre zrušenie zmeny názvu.

7.2.7 Zmena popisu alebo akceptačných kritérií úlohy

Postup pre zmeny v popise alebo v akceptačných kritériách úlohy sa použije vtedy, keď nebol dostatočne popísaný problém, alebo došlo k (dohodnutým) zmenám v akceptačných kritériách.

1. Kliknúť na Agile > Scrum board,
2. V zobrazení šprintu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 2)
Alternatívne: V zobrazení backlogu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 1)
3. Na pravom boku sa zobrazí panel, ktorý obsahuje atribúty úlohy, tu kliknúť do poľa pod Description,
Alternatívne: Stlačiť klávesu e > kliknúť do poľa pod Description,
4. Upraviť na požadovaný obsah,
5. Stlačiť klávesu Enter pre potvrdenie zmien.
Alternatívne: Stlačiť klávesu Esc pre zrušenie zmien.

7.2.8 Priradenie úlohy riešiteľovi / zmena riešiteľa

Riešiteľ je ten člen tímu, ktorí rieši konkrétnu úlohu. Pre priradenie úlohy postupujeme nasledovne:

1. Kliknúť na Agile > Scrum board,
2. V zobrazení šprintu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 2)
Alternatívne: V zobrazení backlogu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 1)
3. Na pravom boku sa zobrazí panel, ktorý obsahuje atribúty úlohy, tu kliknúť na ikonu troch bodiek v pravom hornom rohu panelu > Assign,
4. Za položkou Assignee vybrať riešiteľa,
Alternatívne: Kliknúť na Assign to me,
5. Potvrdiť kliknutím na Assign.
Alternatívne: Zrušiť kliknutím na Cancel.

Tento postup je funkčný aj pre zmenu riešiteľa.

7.2.9 Pridanie komentáru k úlohe

Ak bola úloha vyriešená, tak sa pridá komentár do akého bodu je úloha vyriešená a čaká na revíziu kódu (Metodika revízie kódu od kolegu Bc. Lukáša Masára). Po revízií kódu sa tiež pridá komentár o možnej existujúcej chybe kde sa tiež pridá odkaz do položky Issue Links na úlohu riešiacu túto chybu. Pridávanie komentáru sa spraví nasledovne:

1. Kliknúť na Agile > Scrum board,
2. V zobrazení šprintu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 2)
Alternatívne: V zobrazení backlogu zvoliť požadovanú úlohu (Zobrazenie znázornené po kliknutí klávesy 1)
3. Na pravom boku sa zobrazí panel, ktorý obsahuje atribúty úlohy, tu kliknúť na tlačidlo Comment,
Alternatívne: Stlačiť klávesu m,
4. Vložiť text komentáru,
5. Potvrdiť kliknutím na Add.
Alternatívne: Potvrdiť stlačením klávesy Enter.
Alternatívne: Zrušiť stlačením klávesy Esc.

7.3 Metodika testovania

Autor: Michal Cihák

Táto časť sa zaoberá procesmi vytvárania testov a priebehom testovania.

Projekt tohto tímu je vyvíjaný v jazyku *Ruby* pod rámcom *Ruby on Rails*. Na základe toho boli zvolené tieto testovacie nástroje:

- RSpec - Testovací nástroj pre jazyk ruby, ktorý zabezpečí lepšiu čitateľnosť pre človeka a ľahké porozumenie toho čo daný test testuje. Preto bol uprednostnený tento nástroj oproti Test::Unit, ktorý je síce súčasťou Ruby, no nie je tak prehľadný.
DOC: <http://relishapp.com/rspec/rspec-rails/v/3-1/doc>
- Capybara - Tento nástroj pomáha pri testovaní simulovať interakciu skutočného používateľa s webovou aplikáciou.
DOC: <http://www.rubydoc.info/github/jnicklas/capybara/master>
- Selenium - Tento nástroj je potrebný na testovanie časti aplikácie, ktorá prebieha u klienta, čiže všetky zmeny, ktoré nastanú pomocou JavaScriptu, či CSS.
DOC: <http://www.rubydoc.info/github/jnicklas/capybara/master/Capybara/Selenium>
- Shoulda - Uľahčuje testovanie nad modelmi, vzťahmi medzi nimi a validáciami.
DOC: <http://thoughtbot.github.io/shoulda-matchers/v2.7.0/>

7.3.1 Vytvorenie nového testu

Tento proces všeobecne opisuje vytvorenie testu akéhokoľvek typu. Hlavné typy, ktoré využíva tento projekt budú popísané ďalej.

1. Vytvorenie súboru v `/spec/TYPE/FOLDER/NAME_spec.rb`
 - TYPE - aký typ testu vytvárame
 - FOLDER - v prípade feature testu vytvárame aj priečinok, inak nie.
 - NAME - názov testu sa líši vždy podľa typu testu (upresnené neskôr)Príklad cesty: `/spec/features/events/edit_event_spec.rb`
2. Vloženie `require 'rails_helper'` do prvého riadku.

7.3.2 Vytvorenie nového testu typu Feature

Tento typ testu sa vytvára, keď je potrebné vytvoriť vysokoúrovňový test napodobňujúci používateľa, ktorý prechádza celou aplikáciou. Takýto test zabezpečuje, že jednotlivé časti a komponenty aplikácie správne spolu fungujú.

Tento proces použijeme vždy, keď doplníme do aplikácie funkcionality, ktorá je uskutočniteľná z pohľadu používateľa a prepája všetky vrstvy aplikácie (MVC).

1. Vytvorenie súboru podľa B.3.1. v `/spec/features/FOLDER/METHOD_spec.rb`
 - FOLDER - zodpovedá tomu, nad ktorým modelom (množné číslo) sa tento test vykonáva, resp. ku ktorému controlleru, ktorý spracováva daný model, test patrí
 - METHOD - konkrétna metóda v controller
2. Pri vytváraní testu sa používajú primárne funkcie, ktoré poskytuje knižnica Capybara
3. Do describe sa vpisuje vždy v angličtine aká funkcionality sa testuje

Príklad: `describe 'Creating events'`

4. Pri vytváraní testov sa často opakujú časti kódu - tomuto je potrebné sa vyhnúť a vytvárať pomocné metódy
5. Do it sa píše v angličtine zrozumiteľným spôsobom čo sa stane keď test prejde

Príklad: `it 'displays an error when new event name has less than 3 characters'`

6. Pri vyplňaní foriem a vytváraní objektov sa používajú názvy relevantné k danému objektu a prislúchajúcich atribútov. Tzn. nepoužívajú sa názvy typu 'Lorem Ipsum', náhodné reťazce znakov, ani žiadne názvy neinformujúce o tom, k čomu prislúchajú. Výnimkou sú názvy keď testujeme napríklad validáciu a je potrebné použiť špecifické názvy.

Príklad: `Event.create(name: "My event name", description: "My long event description")`

Príklad: `Event.update(name: "My new event name", description: "My new long event description")`

Príklad feature testu:

```
describe 'Editing events' do
  let!(:event) { Event.create(name: "My name", description: "My long description.")}
  def update_event(options={})
    event = options[:event]
    event.reload
    visit "/events/#{event.id}"
    click_link "Edit"
    fill_in "Name", with: options[:name]
    fill_in "Description", with: options[:description]
    click_button "Update Event"
    event.reload
  end
  it 'should updates the event information with correct info' do
    update_event event: event, name: "New name",
                 description: "New long description"
    expect(page).to have_content("Event was successfully updated")
    expect(event.name).to eq("New name")
    expect(event.description).to eq("New long description")
  end
  it 'displays error when the name is too short' do
    update_event event: event, name: "xy"
    name = event.name
    event.reload
    expect(event.name).to eq(name)
    expect(page).to have_content("error")
  end
end
```

7.3.3 Vytvorenie nového testu typu Feature s použitím Selenium

Testovanie pomocou Selenium sa používa keď nastáva AJAXová komunikácia alebo hocijaká komunikácia medzi vrstvami a daná akcia sa deje len u klienta (Views v MVC).

Tento proces použijeme vždy keď sa doplní Javascriptová funkcionálna, ktorá sa má aktivovať na základe používateľovej interakcie.

1. Vytvorenie súboru podľa B.3.1. v /spec/features/FOLDER/METHOD_spec.rb
 - FOLDER - v tomto prípade zodpovedá Javascriptovému súboru, nad ktorým sa test vykonáva
 - METHOD - konkrétna metóda v súbore
2. Pre písanie testov platia rovnaké pravidlá ako v B.3.2.
3. Používa sa tento kód na aktivovanie Selenium ovládača:

```
before(:all) do
  Capybara.current_driver = :selenium
end
```

4. Na deaktiváciu:

```
after(:all) do
  Capybara.use_default_driver
end
```

Príklad feature testu s použitím Selenium:

```
describe "Event actions" do
  before(:all) do
    Capybara.current_driver = :selenium
  end
  it "pops up a confirm dialog when we click delete" do
    page.click_link("Delete")
    dialog = page.driver.browser.switch_to.alert
    dialog.text.should == "Are you sure?"
    dialog.dismiss
  end
  after(:all) do
    Capybara.use_default_driver
  end
end
```


7.3.4 Vytvorenie nového testu typu Model

Tento typ testu sa používa, keď je potrebné spraviť nízkoúrovňové testy nad modelmi. Ako napríklad testovanie asociačných vzťahov medzi modelmi, testovanie validácií nad modelmi alebo či modely skutočne reprezentujú údaje v databáze.

Tento proces použijeme vždy keď sa vykoná zmena v modeli, alebo medzi modelmi. Tzn. vytvoria sa nové asociačné vzťahy, nové validačné pravidlá, nové atribúty modelu alebo sa čokovek z tohto zmení.

1. Vytvorenie súboru podľa 2.1. v `/spec/models/MODEL_spec.rb`
- MODEL - nad ktorým modelom tento test vykonávame
2. Pri vytváraní testu sa používa primárne knižnica `shoulda-matchers`

Príklad model testu:

```
describe Participant do
  it { should have_many(:time_votes) }
  it { should have_many(:meeting_times).through(:time_votes) }
  it { should accept_nested_attributes_for(:time_votes) }
end
```

7.3.5 Manuálna kontrola testov

Testy je potrebné skontrolovať po každej zmene v kóde, ktorá zasahuje do nejakej funkcionality, ktoré testy kontrolujú. Takisto po vytváraní testu alebo po jeho zmene je potrebné aby tento test prešiel. Testuje sa viacerými spôsobmi:

Spustenie jediného testu:

- kliknutie pravým tlačidlom na daný test -> 'Run Spec' (v RubyMine)
- alebo cez terminal `rspec /spec/features/edit_event_spec.rb`

Spustenie skupiny testov určitého typu:

- rake spec:models

Spustenie všetkých testov:

- rake spec

V prípade, že niektorý test neprejde je potrebné vykonať niektoré z nasledujúcich:

- upraviť kód tak aby bol test splnený a test prešiel
- ak test už viac nie je relevantný a nie je potrebné aby bol viac splnený -> upraviť test alebo ho úplne zmazať
- ak si to aby test prešiel žiada väčšie zmeny alebo nie je jasné či je stále test vhodný -> vytvoriť úlohu (Metodika evidencie úloh v nástroji JIRA), alebo odkomunikovať s ostatnými členmi tímu (Metodika komunikácie)

7.4 Metodika – Code Review

Autor: Lukáš Masár

Účelom prehliadky kódu je zachytiť chyby v kóde pred tým, ako sa dostanú k používateľovi. Je to základná činnosť vykonávaná za účelom zabezpečenia kvality kódu. Jej zanedbanie môže viesť k zbytočne zvýšeným nákladom v ďalších etapách projektu.

Pri prehliadke kódu sa kontroluje:

- **vizuálna kvalita kódu** – formátovanie, komentovanie kódu
- **logická kvalita kódu** – či kód zabezpečuje funkcionality, ktorú má implementovať

7.4.1 Súhrn rol a zodpovednosti účastníkov

Rola	Zoznam zodpovednosti
Programátor	<ul style="list-style-type: none">• napísanie kódu, ktorý sa bude kontrolovať• sprístupnenie kódu určeného na prehliadku cez internet• úprava kontrolovaného kódu, podľa pripomienok, ktoré vzniknu pri prehliadke kódu
Inšpektor	<ul style="list-style-type: none">• prehliadka kódu• spätná väzba pre programátora• nájdenie chýb v kóde• návrhy na úpravy kódu• kontrola funkčnosti• kontrola formátu

7.4.2 Kedy sa vykonáva prehliadka kódu

Keď je kód pripravený na testovanie a nasadenie do produkčného prostredia, čiže keď je otestovaný a funkčný na lokálnom stroji programátora, ktorý ho napísal. Programátor vtedy

pushne svoj kód na development branchu a sprístupní tak kód cez internet (viď dokument “Verziovanie – Git a jeho integrácia v Rubymine“).

7.4.3 Kto vykonáva prehliadku kódu

Iný člen tímu, ako ten ktorý kód napísal. Prehliadku každého kódu vykonávajú minimálne dvaja členovia tímu, a to buď spoločne alebo samostatne. Člen tímu, ktorý vykonal prehliadku kódu túto aktivitu zaznamená v JIRE v issue, ktorá sa daného kódu týka (viď. dokument „Metodika evidencie úloh v nástroji JIRA“).

7.4.4 Náhľad metodiky

Postup krokov:

- programátor/i napíše kód implementujúci konkrétnu issue
- programátor sprístupní kód cez internet
- inšpektor/i vykoná prehliadku kódu
- zostaví sa zoznam pripomienok, problémov pre danú prehliadku
- programátor vykoná prípadné úpravy kódu pre opravu nájdených chýb, resp. vyhovie pripomienkam inšpektora
- kontrolovaný kód je schválený a pripravený pre testovanie a nasadenie v produkčnom prostredí

7.4.5 Postup metodiky

A) Napísať kód a požiadať o jeho prehliadku

- programátor/i si vyberie issue, ktorú chce implementovať
- programátor implementuje a lokálne otestuje danú issue
- keď je kód lokálne funkčný, programátor kód sprístupní cez internet na development vetve vzdialeného repozitára (viď dokument “Verziovanie – Git a jeho integrácia v Rubymine“)

- programátor požiada o prehliadku kódu pomocou komentáru v danej issue v JIRE, alebo pomocou správy v HipChat-e

B) Prehliadka kódu a spätná väzba

- inšpektor/i prechádza kód prístupný cez internet riadok po riadku a pridáva doňho komentáre, krátke a zrozumiteľné, ako spätnú väzbu
- komentáre, obsahujúce pripomienky na úpravu kódu musia obsahovať na začiatku značkou “TODO”
- “TODO” pripomienky, ktoré majú byť spracované pred schválením kódu, musia byť spísané do zoznamu taskov, ktorý je poskytnutý programátorovi
- inšpektor kontroluje funkčnosť, teda či vlastnosť, ktorú kód implementuje, alebo chyba, ktorú fixuje, funguje
- inšpektor kontroluje, či je kód riadne zdokumentovaný a okomentovaný
- inšpektor kontroluje, či kód dodržiava formátovanie

C) Opraviť objavené chyby a dokončiť prehliadku

- programátor/i, ktorý kód napísal, upraví kód podľa pripomienok, ktoré vznikli pri prehliadke kódu a opraví všetky chyby, ktoré sa našli
- upravení kód sprístupní cez internet na development vetve vzdialeného repozitára a požiada o opätovnú prehliadku
- cyklus sa opakuje, kým kód nie je schválený inšpektormi (minimálne dvomi) bez ďalších pripomienok
- keď je kód schválený, môže byť testovaný pre produkčné prostredie

7.4.6 Ďoplnujúce informácie a požiadavky

- prehliadka kódu sa musí vykonať pred tým ako sa kód testuje v produkčnom prostredí
- o prehliadku žiada ten/tí, kto kód napísal
- všetci členovia tímu by sa minimálne raz mali zúčastniť na prehliadke nejakého kódu

- kód, ktorý je predkladaný na prehliadku musí byť riadne zdokumentovaný a okomentovaný
- kód ktorý je predkladaný na prehliadku, musí dodržiavať formátovanie v súlade so zaužívanými programovacími konvenciami
- ak programátor žiada o prehliadku kódu, ktorý opravuje nejakú chybu, treba podložiť výsledky testov, ako dôkaz, že je chyba opravená
- programátor musí poskytnúť krátky zmysluplný opis toho, čo kód implementuje
- odporúča sa robiť prehliadku najviac na 400 riadkoch kódu a dlhšie kódy rozdeliť na viac prehliadok

7.5 Metodika verziovania v Git a RubyMine

Autor: Matúš Pikuliak

Časť dokumentu pojednáva o procesoch pri verziovaní kódu a súborov všeobecne pri vyvíjaní implementácie nášho projektu. Procesy slúžia programátorom pri písaní kódu a jeho synchronizácii v rámci tímu.

Náš tím používa ako verziovací systém Git. Máme zriadený vzdialený repozitár, kde si tím udržiava kópie projektu. Náš projekt vyvíjame v jazyku Ruby na frameworku Ruby on Rails. Keďže náš tím používa IDE Rubymine, jednotlivé kroky sú opísané práve pre tento program.

Prvá časť tohto dokumentu, Manažment verziovania Git v Rubymine, pojednáva o využívaní nástroja Rubymine pri práci s repozitármi. Druhá časť, Správa vetiev, opisuje náš model vetvenia a procesy, ktoré súvisia s vytváraním, tagovaním a celkovo prácou s vetvami počas vývoja nášho projektu.

7.5.1 Vytvorenie nového lokálneho projektu

Tento proces sa vykonáva, keď chce vývojár vytvoriť zo vzdialeného repozitára lokálny a začať na ňom pracovať. Predpokladá sa, že má prístup do nášho repozitára na Bitbucket. V opačnom prípade kontaktujte vedúceho tímu.

1. VCS > Checkout from Version Control > Git
2. Do prvého poľa skopírovať .git odkaz zo stránky projektu na bitbuckete, napr:
`https://mpikuliak@bitbucket.org/ltursky/catchandmatch.git`
3. Nastaviť adresáre do ktorých sa stiahnu súbory z repozitára.
4. Potvrdiť voľby tlačidlom Clone.
5. Zadať overovacie údaje.
6. Súbory sa stiahnu do lokálneho repozitára.
7. V prípade nutnosti inštalácie nových gemov Rubymine upozorňuje na túto skutočnosť.
Kliknutím na ponúkanú popup správy a gemy sa nainštalujú.
Alternatívne: Tools > Bundler > Install a kliknúť na Install.
8. Nastaviť súbor `config/database.yml` podľa vzoru na bitbuckete

9. Vytvoriť databázu:

Ctrl+Alt+R > db:create

Ctrl+Alt+R > db:migrate

Po tomto postupe by mal byť obsah vzdialeného repozitára skopírovaný do žiadaného priečinku a projekt by mal byť naviazaný na tento repozitár. Malo by byť takisto možné spustiť rails server.

7.5.2 Aktualizácia pracovnej kópie súborov

Tento proces sa vykonáva, keď sa zo vzdialeného repozitára sťahuje do lokálneho aktuálna verzia súborov. Toto sa robí na začiatku každej programovacej *session*, ako aj pred odosielaním lokálnej verzie späť na server.

1. Vybrať správnu vetvu, viď. *Zmena pracovnej vetvy*.
2. VCS > Update Project
Alternatívne: Ctrl + T
3. Vybrať *Update type Merge*.
4. Kliknúť na OK.
5. Zadať overovacie údaje.
6. V prípade rozporov vykonať v ponúkanom rozhraní merge nad súbormi, pri ktorých nastali konflikty.
7. V prípade nutnosti inštalácie nových gemov Rubymine upozoňuje na túto skutočnosť.
Kliknutím na ponúkanú popup správu sa gemy nainštalujú
Alternatívne: Tools > Bundler > Install a kliknúť na Install.
8. Do databázy natiahnite nové migrácie:
Ctrl+Alt+R > db:migrate

Pracovná kópia by mala byť aktualizovaná a je možné spustiť rails server.

7.5.3 Commitovanie

Tento proces sa vykonáva, keď sa vytvára lokálna snímka stavu súborov. Táto sa robí, keď sa ukončí práca na jednej malej logickej časti systému.

V jednom commite sa nachádzajú len súbory týkajúce sa zmeny jednej malej logickej časti systému. Do commitu sa neprikladajú drobné zmeny z častí, ktoré s commitom priamo nesúvisia. Týmto drobným zmenám sa vytvára samostatný commit. Do jedného commitu však možno zaradiť viacero podobných drobných zmien, ktoré priamo nesúvisia s chovaním systému, napr. zmeny v syntaxi. Necommitujú sa ani konfiguračné súbory (napr. database.yml), ktoré by mohli narušiť prácu ostatným developerom.

O správu súborov, ktoré budú commitované sa automaticky stará Rubymine. Vytváranie, mazanie, presúvanie, premenovanie či modifikácia súborov sú v prostredí Rubymine automaticky zaznamenávané. Zmeny možno sledovať v záložke Changes.

Postup samotného commitovania je nasledovný:

1. VCS > Commit changes / Ctrl + K
2. V strome súborov vyberať, ktoré zo zmien sa commitnú.
3. Vyplniť commit správu. Jej formát je opísaný nižšie.
4. Ostatné položky nechať nezaškrtnuté
5. Kliknúť Commit.

Formát commit správ je nasledovný:

```
[<Issue ID>] <Issue Name>
```

```
<subor 1> - <zmena v subore 1>
```

```
<subor 2> - <zmena v subore 2>
```

```
...
```

```
<subor N> - <zmena v subore N>
```

Issue ID a Issue Name sú jedinečný identifikátor a meno issue z projektového issue trackera v systéme Jira (napr. [CATCHME-23] Improve Dates). Každý commit sa týka práve jednej issue. Drobný refactoring, ku ktorému sa nebude vyrábať samostatná issue v Jire označíme tagom [R].

Nasleduje opis zmien v jednotlivých súboroch. Opis zmien sa píše v anglickom jazyku v minulom čase (napr. something was changed, something was implemented). Opis zmien by mal stručne, ale úplne informovať o zmenách v daných súboroch a ich motivácii.

Príklad commit správy:

```
[CATCHME-26] [C] Link Generation
```

```
app/models/event.rb - Implemented token generator in Event model. This token is now used to generate URLs.
```

```
app/controllers/events_controller.rb - Controllers refactored to find event objects by tokens and not by ids.
```

7.5.4 Aktualizácia vetvy vzdialeného repozitára

Tento proces sa vykonáva, keď sa zasiela lokálna verzia súborov na server. Súborny sa aktualizujú na konci každej programátorskej *session*. Aktualizuje sa aj keď sa ukončí práca na jednej väčšej časti systému a presúva sa na inú.

Pred aktualizáciou sa overujú tieto náležitosti:

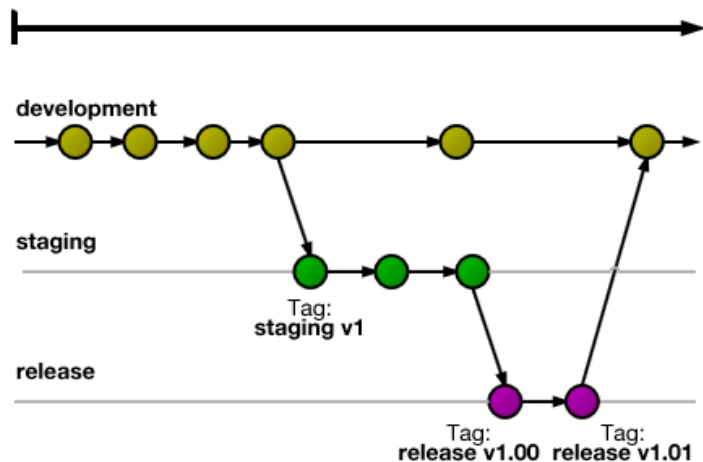
1. Je vybraná správna vetva, ktorá sa bude aktualizovať, vid'. *Zmena pracovne vetvy*.
 2. Sú stiahnuté aktuálne súborny z danej vetvy, vid'. *Aktualizácia pracovnej kópie súborov*.
 3. Súborny tak ako sa budú pushovať prejdú všetkými aktívnymi automatickými testami
- Samotná aktualizácia prebieha nasledovne:

1. VCS > Git > Push
Alternatívne: Ctrl + Shift + K
2. Po prezretí zmien v commitoch kliknúť Push.
3. Pri probléme so synchronizáciou súborov sa celá aktualizácia ruší a súborny sa aktualizujú zo vzdialeného repozitára, vid'. *Aktualizácia pracovnej kópie súborov*.
4. Pri vetvách, nad ktorými funguje Continuous Integration (development a production) overiť či aktualizácia nespôsobila chybu. V prípade že áno postupovať podľa príslušných procesov.

7.5.5 Správa vetiev

Pri vývoji projektu sa používajú tri základné vetvy:

- *development* - vetva, na ktorej prebieha každodenný vývoj projektu. Z tejto vetvy ťahajú a do tejto vetvy odovzdávajú kód vývojari pri bežnej práci. Nad touto vetvou sa vykonáva Continuous Integration a je prístupná na internete.
- *staging* – vetva, ktorá sa obnovuje vždy pre vydaním nového produkčného releasu. Táto vetva slúži ako miesto kde sa aktuálna developerská vetva upravuje do prezentovateľného výsledku v dohodnutej funkcionalite. Úlohou práce na tejto vetve je úprava kódu pre potreby releasu a mazanie nadbytočnej funkcionality.
- *release* – vetva, v ktorej sa nachádzajú funkčné verzie produktu vhodné pre produkciu. Nad touto vetvou sa vykonáva Continuous Integration a je prístupná na internete. Nad touto vetvou sa vykonáva minimum práce. Verzia nasadená na tejto vetve je prezentovaná ako aktuálny stav nášho produktu. Pokiaľ sa nedohodne inak, číslo každého releasu je o jedna väčšie ako číslo predchádzajúceho a začína sa od verzie 1.



Obr. B.5.1 – Model vetvenia pri vytváraní releasu

Okrem týchto troch základných vetiev sa v projekte nachádzajú aj tzv. *features* vetvy, v ktorých sa vyvíjajú nové časti systému, či v ktorých sa skúšajú nové technológie. Tieto vetvy sú

od *developmentu* oddelené vtedy, keď hrozí že by plánované zmeny v kóde mohli narušiť jeho štruktúru a preto sa radšej vytvorí nová vetva, na ktorej sa paralelne vyvíja.

7.5.6 Vytváranie releasu

Tento proces sa uskutočňuje vždy, keď na tímovom stretnutí padne rozhodnutie vytvoriť nový release pre potreby dema či nasadenia produktu. Na mítingu sa špecifikuje funkcionálna daného releasu, aj vo vzťahu k potencionálnym *features* vetvám. Na tomto mítingu je vybraný člen tímu, ktorý zodpovedá za manažment verziovania a vetvenia počas vytvárania tohto releasu. Tento má na starosti aj kroky opisovaná v tomto procese.

1. Vetva *development* sa merge do vetvy *staging*.
2. Tento mergeovací commit sa otaguje číslom príslušnej verzie: `staging v<číslo>`
napr.: `staging v1`

Nad týmto commitom začnú následne pracovať všetci developeri, ktorí upravujú kód tak, aby vyhovoval špecifikácii určenej na stretnutí. Poverený človek sleduje priebeh týchto zmien a je informovaný o stave vetvy. Napokon poverený človek overí u ostatných členov tímu, či ukončili prácu na vetve a či je v stave vhodnom pre release.

3. Vetva *staging* sa merge do vetvy *release*. Do vetvy *staging* sa ďalej neprispieva.
4. Tento mergeovací commit sa otaguje číslom príslušnej verzie: `release v<číslo>`
napr.: `release v1.00`

V správe k danému tagu treba v odrážkach spísať hlavné zmeny vo funkcionalite oproti predošlému releasu.

Nad týmto commitom ešte ďalej prebieha krátky čas vývoj, ktorý má za účel odchytiť drobné chyby. Poverený človek má v tomto prípade povinnosť tagovať nové commity pokračujúcimi číslami verzie, napr.: `release v1.01`. Do správ týchto tagov uvádza zoznam zmien z používateľského hľadiska.

Po ukončení vývoja na pre aktuálny release poverený človek merge túto vetvu späť do *development*, aby sa uchovali prípadne opravy bugov a podobne. Rozhodnutia o pokračovaní či ukončení vývoja na release vetve sa dejú na tímovom mítingu.

7.5.7 Features vetvy

Features vetvy vytvárajú developeri keď chcú v kóde urobiť zmeny, ktoré by mohli narušiť jeho štruktúru či funkčnosť, keď chcú vyskúšať nový prístup či technológiu. Pri vytváraní feature vetvy vývojári nasledujú tento proces:

1. Vytvoriť novú vetvu s menom `feature-<názov feature>`, napr. `feature-angular` alebo `feature-user_roles`
2. Prvému commitu nastaviť tag s menom opisujúcim názov danej feature. V popise tagu uviesť podrobný opis plánovaných zmien a dôvod prečo bolo potrebné vytvoriť novú vetvu.

Vývoj na feature vetve prebieha podobne ako na development s tým rozdielom, že tieto vetvy nemajú nad sebou spustený proces Continuous Integration. Ktokoľvek môže modifikovať a upravovať kód, či ho ďalej vetviť podľa uváženia. Podľa potreby možno takisto feature vetvu mergeovať späť do *development* vetvy.

7.5.8 Zmena pracovnej vetvy

Tento a nasledovné opísané podprocesy slúžia len ako pomôcka pri vykonávaní procesov opísaných vyššie, týkajúcich sa práce s vetvami pri vývoji projektu.

Názov aktuálnej vetvy sa zobrazuje vpravo dole v stavovom riadku v okne Git. Okrem názvu sa tu po kliknutí dá dostať aj k ďalším vetvám z projektu. Pracovná vetva sa mení nasledovne:

1. Kliknúť na Git okno v stavovom riadku
2. Kliknúť na názov požadovanej vetvy.
3. Zvoliť Checkout.
4. V prípade problémov so synchronizáciou zvoliť možnosť Smart checkout a mergnúť súbory

7.5.9 Vytvorenie novej vetvy

1. Kliknúť na Git okno v stavovom riadku.
2. Zvoliť možnosť New Branch.
3. Zadať názov branche.

4. Kliknúť OK

7.5.10 Mergovanie vetiev

1. Prepnúť sa do vetvy, do ktorej chceme mergovať
2. Kliknúť na Git okno v stavovom riadku
3. Kliknúť na vetvu, ktorú chceme do aktuálnej mergovať
4. Kliknúť Merge

7.5.11 Tagovanie commitov

1. Okienko Changes v dolnom paneli > Log
2. Kliknúť na vybraný commit a okopírovať jeho hash.
3. VCS > Git > Tag files
4. Do okienka commit nakopírujte hash commitu a prípadne validujte.
5. Do okienka Tag name uveďte samotný tag.
6. Do okienka Message uveďte prípadne ďalšie informácie o commite.
7. Kliknite OK.

7.6 Metodika komunikácie

Autor: Pavol Michálek

7.6.1 Kompetencie členov tímu

Pomocou neskôr vymedzených komunikačných nástrojov sa členovia tímu CatsCanCode môžu okrem tímových stretnutí prednostne obrátiť s dotazmi na členov tímu:

- Ing. Lukáš Turský - *email: xtursky@gmail.com, tel.č.: 0918219725*
vlastník produktu, smerovanie vývoja,
- Michal Cihák - *mxcihak@gmail.com, tel.č.: 0918966744*
manažér testovania, jadro aplikácie,
- Michal Gajdoš - *michalgajdo@gmail.com, tel.č.: 091892237*
manažér rizík
- Lukáš Masár - *lukas.masar@gmail.com, tel.č.: 0904111729*
aplikačný server, nasadzovanie nových verzií
- Pavol Michálek - *pali.michalek@gmail.com, tel.č.: 0902682727*
tímová stránka, front-end aplikácie
- Vladimír Osvald - *osvald.v@gmail.com*
dokumentácia,
- Matúš Pikuliak - *matus.pikuliak@gmail.com*
front-end aplikácie
- Tomáš Sýkora - *sykora.tomino@gmail.com, tel.č.: 0905384406*
zápisnice

7.6.2 Interná tímová komunikácia - nástroj HipChat

V prípade komunikácie v tíme a s vlastníkom produktu sa používa chatovací nástroj HipChat. Je nakonfigurovaný na konkrétne potreby tímu CatsCanCode. Každý člen tímu má v pracovnej dobe tento komunikačný nástroj zapnutý a v čo najkratšom čase reaguje na nové správy, ktoré sa ho týkajú. Je teda potrebné mať zapnuté notifikácie o nových správach. Preddefinované je nastavenie zasielania notifikačných emailov v čase, keď je člen offline.

Internetová komunikácia členov tímu sa musí v čo najväčšej miere sústrediť v nástroji HipChat. Výnimkou je

- nástroj **Google Hangouts** používaný na zdieľanie obrazovky a internetové (video)hovory,
- nástroj **Jira** používaný na komunikáciu (komentovanie) spojenú s konkrétnymi úlohami (“task”) a chybami (“bug”).

7.6.3 Miestnosti

Na základnej obrazovke (Lobby) je zoznam miestností. Boli vytvorené na špecifické prípady komunikácie:

- *Team Page* - slúži na informovanie ohľadom nových skutočností spojených s tvorbou, prevádzkou a aktualizáciou tímovej stránky (umiestnenej na adrese <http://team12-14.ucebne.fiit.stuba.sk/>). Miestnosť má nastavenú integráciu na webový repozitár BitBucket s projektom tejto stránky a je v nej možné sledovať vývoj stránky každým commitom.
- *Rails* - slúži na informovanie ohľadom vývoja aplikácie CatchMe, jej vývojového prostredia, aplikačného servera, Miestnosť má nastavenú integráciu na webový repozitár BitBucket s projektom tejto aplikácie a je v nej možné sledovať vývoj projektu každým commitom.
- *Jira* - slúži na sledovanie aktivity ostatných členov tímu v nástroji Jira. Miestnosť má nastavenú integráciu na “Activity Stream” nástroja Jira.
- *CatsCanCode* - sústreďuje každodennú komunikáciu v rámci tímu. Zahŕňa oznamy o najbližších stretnutiach, organizácii predmetu, pripomienky blížiacich sa termínov,

Každý člen tímu má rovnaké oprávnenia v týchto chatovacích miestnostiach. Každý člen má možnosť sa **stručne a vecne vyjadriť** v príslušných miestnostiach a informovať tak ostatných členov o nových skutočnostiach. Nesmú zasielať správy nesúvisiace s témami miestností.

Adresovanie správ v miestnostiach

Pri komunikácií v spoločných miestnostiach je vyžadované špecifikovať na začiatku správy adresáta. Robí sa tak **znakom “@” nasledovaným celým menom adresáta** (mená sa automaticky ponúkajú po napísaní znaku @). Je možné napísať aj viacerým adresátov. Pokiaľ je správa smerovaná na každého člena tímu, používa sa reťazec “@all”.

HipChat umožňuje **zasielanie správ len jednému členovi**. Uprednostňujte tento spôsob v prípade, že sa jedná o informácie, ktoré by zaťažovali ostatných členov, svojím rozsahom zneprehľadnili komunikáciu v miestnostiach alebo sa jedná o riešenie konkrétnej úlohy, na ktorej títo dvaja spolupracujú.

7.6.4 Špeciálna interná tímová komunikácia - nástroj Google Hangouts

V prípade, že sa jedná o rozsiahlejšiu komunikáciu viacerých členov tímu, nie je možné osobné stretnutie a nepostačuje textová forma HipChat-u, používa sa komunikačný nástroj Google Hangouts. Používa sa na hovory, videohovory a zdieľanie obrazovky.

7.6.5 Webové sídlo tímu

Slúži na informovanie verejnosti o priebehu projektu.

Zápisnica stretnutia je dokončená najneskôr ďalší deň po stretnutí a je uverejnená na webovom sídle tímového projektu. Uverejňuje sa aj **aktualita** (prípadne dve) z posledného obdobia. Na stretnutí sa určia členovia poverení dokončením dokumentu a uverejnením dokumentu a aktuality na webových stránkach tímu. Na proces dohliadajú členovia s pôvodnými kompetenciami (Tomáš Sýkora - dokumentácia projektu, Pavol Michálek - webové sídlo tímu).

Projekt webového sídla je zdieľaný webový repozitárom BitBucket a zmeny sa nasadzujú na projektový server, ku ktorému majú prístup všetci členovia tímu.

7.6.6 Komunikácia o úlohách

V nástroji Jira, ktorý zaznamenáva priebeh vývoja projektu, prebieha komunikácia v komentároch jednotlivých úloh. Je dôležité, aby čo najjemnejšie zaznamenávali prácu na týchto úlohách, ktorá sa týka integrácie, návazností, spolupráce viacerých členov a finalizácie práce.

Na sprehl'adnenie dlhších komentárov je odporúčané použiť možnosti formátovania textu (krátku dokumentáciu je možné nájsť pod textovým poľom nového komentáru po kliknutí na “?”).

Adresovanie komentáru sa používa len pri vymenúvaní konkrétnych členov tímu, ak je potrebné určiť kompetentného člena (a nestačí pridelenie - *Assignee*). Nemení sa prednastavená viditeľnosť komentáru (*Viewable by All Users*).

7.6.7 Zdieľanie súborov - Google Drive

Všetky súbory súvisiace s prácou na tímovom projekte s výnimkou zdrojových kódov sa sústredia na webovom úložisku Google Drive. Všetci členovia majú rovnaké oprávnenia na zdieľaný spoločný priečinok projektu na tomto úložisku.

Samostatné sú podpriečinky s materiálmi z pravidelných tímových stretnutí. Nachádza sa v nich zápisnica a fotografická dokumentácia zo stretnutia.

Pre ďalšie potreby samostatných tém, je možné vytvárať podpriečinky zaradený v koreňovom priečinku alebo na príslušnom mieste v strome priečinkov. Podpriečinky majú svoj názov vo formáte *XX Názov podpriečinku*, pričom *XX* je dvojčiferné inkrementujúce poradové číslo.

7.7 Metodika dokumentovania inžinierskeho diela

Autor: Vladimír Osvald

Tento dokument obsahuje pracovné procesy a postupy potrebné pri písaní dokumentácie k inžinierskemu dielu pomocou nástrojov Wiki a Microsoft Word. Obsahuje pokyny pre editáciu a finalizáciu dokumentu.

Tento dokument je určený všetkým členom tímu, ktorí sa podieľajú na programovaní inžinierskeho diela a jeho následnom dokumentovaní.

7.7.1 Identifikácia rolí

Na tvorbe dokumentácie sa podieľajú dve používateľské role.

Dokumentátor - každý člen tímu, ktorý sa podieľa na tvorbe dokumentácie k inžinierskemu dielu v nástroji Wiki.

Dokončovateľ - určený člen tímu, ktorý vyexportuje dokumentáciu z Wiki do nástroja Word, kde celý dokument naformátuje do finálnej verzie.

7.7.2 Dostupnosť dokumentácie

Dokumentácia k inžinierskemu dielu je dostupná na stránkach Bitbucket.org vo vytvorenom projekte Itursky / CatchAndMatch v sekcii Wiki. Názov súboru je „Dokumentácia k inžinierskemu dielu“.

Výsledný naformátovaný dokument sa nachádza na gDrive v zložke Dokumentácie. Názov súboru je „Dokumentácia k inžinierskemu dielu“.

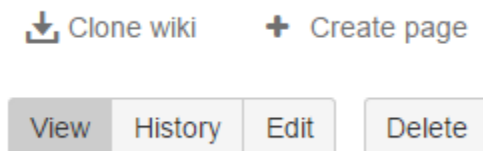
7.7.3 Dokumentovanie

Súbor s dokumentáciou na Wiki (umiestnenie súboru v kapitole 4) obsahuje predpripravenú základnú štruktúru.

7.7.4 Proces editovania

Dokumentátori editujú tento súbor vždy, keď je potrebné doplniť aktuálne informácie do súboru. Súbor s dokumentáciou k inžinierskemu dielu obsahuje prevažne technickú dokumentáciu. Dokumentovanie prebieha bezodkladne po ukončení programovania funkcie alebo dostatočne malého modulu. Dostatočne malým modulom sa rozumie úsek programu, ktorý na seba priveľmi nadväzuje a nie je možné ho rozdeliť na menšie celky.

Po vyhládní súboru a jeho otvorení sa na obrazovke zobrazí hlavná stránka. V pravom hornom rohu sa nachádzajú následovné ovládacie prvky:



Obr 1. - Hlavná ponuka wiki

Edit – umožní prístup k editovaniu dokumentu. Po zvolení tejto možnosti sa zobrazí nasledujúce okno, v ktorom editujeme dokument:

Title*

Content* H1 H2 H3 B I Markdown Preview

```
[TOC]
# Your title here... #
## Your title here... ##
### Your title here... ###
**tucne**
"kurziva"
...
#truby

ukazka kodu
...

[linka](https://bitbucket.org/tursky/catchandmatch/wiki/Dokument%C3%A1cia%20k%20in%C5%BEinierskemu%20dielu)
```

Commit message

Obr 2. Editačné okno

Editačné okno je rozdelené do štyroch horizontálnych sekcií a to Title, Content, Commit message a Save/Cancel tlačidlá.

V procese Editovania sa nezasahuje do sekcia Title, ktorá je prednastavená. Dokumentovanie prebieha v sekcii Content podľa pravidiel uvedených nižšie.

Po Ukončení editovania je potrebné vyplniť sekciu Commit message (viď obr 2.). V tejto sekcii sa stručne ale výstižne opíšu vykonané zmeny na dokumente.

Pravidlá editácie:

1. Dodržiava sa syntax určený pre Wiki
2. Používajú sa nadpisy troch úrovní
 - Prvou úrovňou označujeme hlavné kapitoly
 - Používame nasledovnú syntax : # nadpis prvej úrovne #
 - Druhou úrovňou označujeme podkapitoly
 - Používame nasledovnú syntax : ## nadpis druhej úrovne ##
 - Treťou úrovňou označujeme podrobnejšie sekcie podkapitol
 - Používame nasledovnú syntax : ### nadpis tretej úrovne ###
3. Zvýrazňujú sa dôležité pojmy a vety
 - Zvýrazňujeme pomocou tučného a šikmého písma
 - Syntax pre tučné písmo : **** text ****
 - Syntax pre šikmé písmo : ** text **
4. Vložené obrázky sa vždy popisujú a číslujú
5. Pri dokumentovaní funkcií sa vkladajú ukážky kódu
 - Syntax na vloženie kódu :

```
'''
```

 - začiatok bloku s kódom
 - ```
#!/ruby
```

 - definícia jazyka
  - Ukážka kódu - vložený kód
  - ```
'''
```

 - koniec bloku s kódom
6. Pri vkladaní linkov používame nasledovný syntax: [linka](url adresa)

7. Nevymazáva sa tag [TOC] na začiatku dokumentu predstavujúci samogenerujúci obsah

Postup editovania dokumentácie:

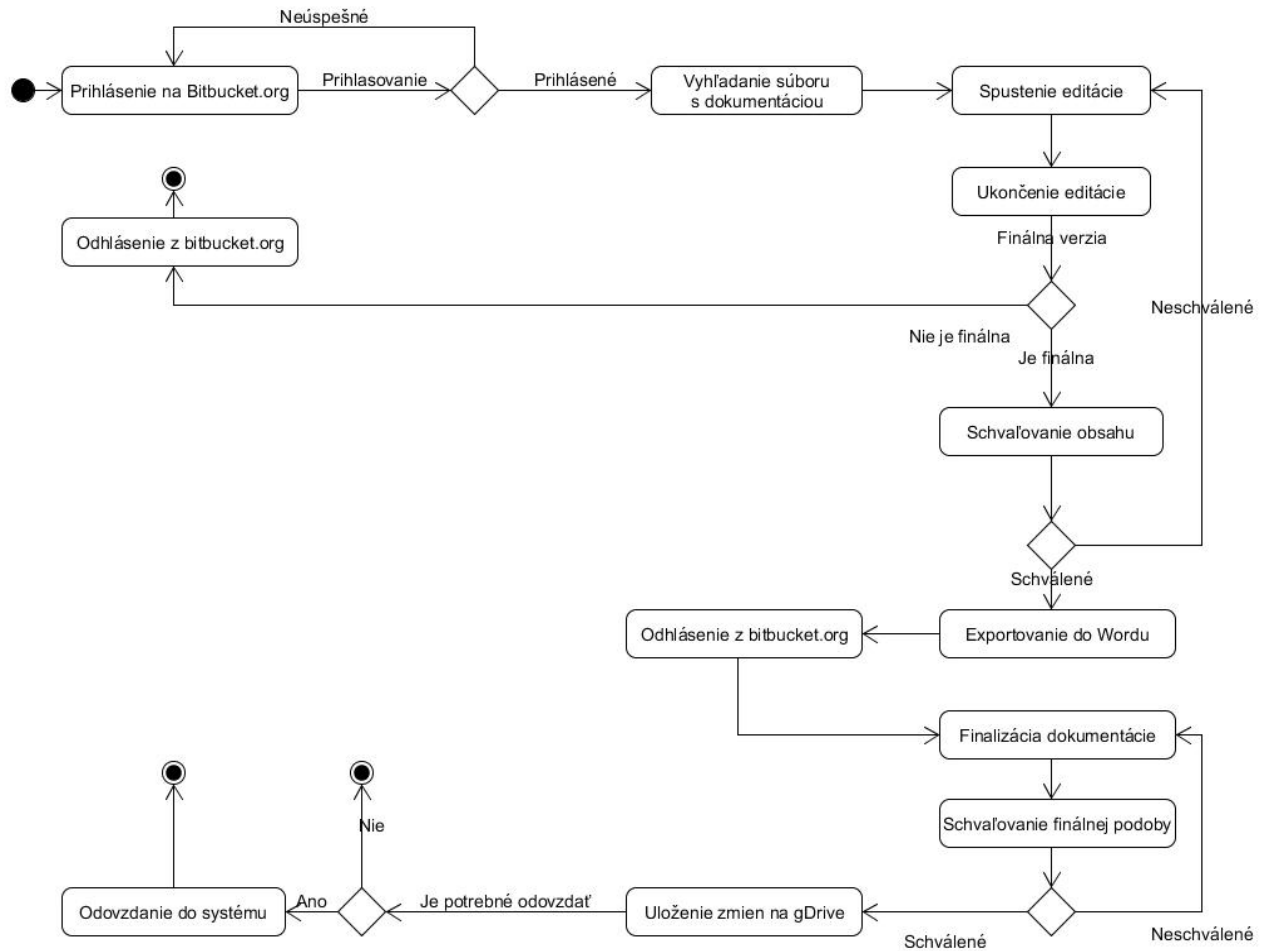
1. Prihlásenie do služby Bitbucket
2. Vyhľadanie projektu CatchAndMatch
3. Vyhľadanie súboru s dokumentáciou k inžinierskemu dielu
4. Zvolenie možnosti “Edit”
5. Vykonanie zmien
6. Okomentovanie zmien
7. Uloženie zmien
8. Kontrola, či boli zmeny vykonané
9. Odlhásenie zo služby Bitbucket

7.7.5 Proces finalizácie

Postup finalizácie je nasledovný:

1. Tím schváli obsah dokumentu na wiki
2. Finalizátor príslušný .md súbor s dokumentáciou prekonvertuje na formát .docx
3. Otvorí sa dokumentácia vo Worde
4. Dokumentátor podľa pravidiel uvedených v sekcii 5.4 naformátuje dokument
5. Po formátovaní sa tímom schváli výsledná podoba
6. Dokument sa uloží na gDrive
7. V prípade potreby sa dokument odovzdá do akademického system

1.1 Diagram procesu dokumentovania



Obr. 3 - Diagram procesu dokumentovania

7.7.6 Pravidlá formátovania výsledného dokumentu

1. Dodržiava sa stanovený obsah titulnej stránky
 - V hlavičke názov fakulty, centrované na stred
 - V strede titulnej strany názov projektu a dokumentu
 - V ľavej dolnej časti mená členov tímu, meno vedúceho, akademický rok - centrované naľavo
2. Prvý odstavec sa neodsadzuje, ostatné o tabulátor
3. Používajú sa nadpisy rôznych úrovní totožné s wiki
4. Nadpisy zarovnávame naľavo

- Prvá úroveň - font Calibri, 16, bold
 - Druhá úroveň - font Calibri, 14, bold
 - Tretia úroveň - font Calibri, 12, bold
5. Text zarovnáваме naľavo
- Font Arial, 12
6. Obrázky zarovnáваме na stred
7. K obrázkom sa pridáva popis v tvare „Obr. # - názov“
- # označuje poradové číslo obrázka v dokumente
 - názov označuje názov obrázku
 - centrované na stred, kurzíva
8. Tabuľky zarovnáваме na stred
9. K tabuľkám sa pridáva popis v tvare „Tab. # - názov“
- # označuje poradové číslo tabuľky v dokumente
 - názov označuje názov tabuľky centrované na stred, kurzíva

8 Zápisnice zo stretnutí

V predloženej prílohe 8 sa nachádzajú zápisnice z jednotlivých tímových stretnutí.

8.1 Zápisnica stretnutia č. 1

Autor	Bc. Tomáš Sýkora
Dátum a čas stretnutia	30.09.2014, 13:00 -16:00
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.1.1 Osnova na stretnutie

- Určiť názov tímu.
- Na základe názvu navrhnúť logo a plagát.
- Špecifikovať požiadavky.

8.1.2 Priebeh stretnutia

- Navrhovali sme základný flow.
- Rozberali sme názov produktu. Všetci sme sa zhodli na tom, že názov bude obsahovať slovo "catch", keďže sa snažíme "pochytať" ľudí na stretnutie. Vznikli dve varianty:
 - Catch me

- Catch and Match
- Na základe názvu tímu sme sa dohadovali na logu tímu. Najbližšie k tejto tematike nám prišla analógia mačka a myš. Zhodli sme sa na mačacej labke a počítačovej myši.
- Ohľadom plagátu sme vyberali layout. Vybrali sme si layout hollywoodskeho plagátu.
- Ďalej sme navrhovali dizajn produktu, definovali sme si základné požiadavky:
 - Minimalistický dizajn, easy-to-use, quick
 - V hlavnej časti stránky budú výrazné štyri kachlíky:
 - What?
 - When?
 - Where?
 - With?
 - Pre skupiny ľudí
 - Podpora dočasných účtov (generovať každému používateľovi ID, ktorý sa prihlási a napr generovať do 10 000 a potom zmazať), neskôr na registráciu (mail, integrácia so sociálnymi sieťami)
 - Notifikácie:
 - mailové
 - pomocou integrácií (???)
 - neskôr pravdepodobne push cez mobilnú appku
 - Módy:
 - neformálne stretnutia (priatelia, kolegovia, spolužiaci, atď.)
 - formálne (projekt vedúci-študent, atď.)
 - Znovupoužitie - pomocou vygenerovaného linku na udalosť, po odohraní udalosti sa udalosť zmaže
 - Vizualizácia stavu (Teplomer, atď.)

8.2 Zápisnica stretnutia č. 2

Autor	Bc. Tomáš Sýkora
Dátum a čas stretnutia	07.10.2014, 13:00 - 17:00
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.2.1 Osnova na stretnutie

- Určiť skupiny ľudí, pre koho bude produkt.
- Definovať vlastnosti produktu.
- Prečo vlastne tento produkt vzniká?
- Určiť technológie, ktoré budú použité na vývoj.

8.2.2 Priebeh stretnutia

- Určili sme skupiny ľudí, pre ktorých bude, resp. nebude tento produkt:
 - + Vek 15-33
 - + Uponáhľaní
 - + Pohodlní
 - + Komunikujúci cez internet
 - - Firmy
 - - Profesionálne eventy / konferencie
- Ujasnili sme si, prečo vlastne tento produkt vzniká:

- Uľahčenie dohadovania
 - Na jeden klik
 - Rýchle
- Matcher
- Definovali sme vlastnosti produktu:
 - Jednoduchý
 - Intuitívny
 - Catch and Match
 - Maching
 - Akceptačné kritériá
 - Automatické
 - Manuálne
 - Priority
 - Účty
 - Dočasné
 - Cez integrácie
 - Eventy
 - Súkromné
 - Verejné
 - Opakovaná návšteva eventu (Revisit), možno z viacerých zariadení?
 - Overenie cez pin alebo nejakú frázu
 - unikátny link na užívateľa
 - Real-time?
 - Vizualizácia stavu
 - Notifikácie
 - Módy
 - Integrácie
 - Mobilná aplikácia
- Určili sme si cieľ produktu:
 - Uľahčiť a zjednodušiť dohadovanie stretnutí na par klikov.
- Rozoberali sme technológie na vývoj:

- Ruby on Rails - pre naše potreby pokrýva všetko
- Databázový server PostgreSQL

8.3 Zápisnica stretnutia č. 3

Autor	Bc. Pavol Michálek
Dátum a čas stretnutia	14.10.2014, 13:00 - 16:30
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.3.1 Osnova na stretnutie

- Nájsť vhodné komunikačné nástroje, version control nástroje, scrum nástroje.
- Určiť "Issues" na prvý šprint.

8.3.2 Priebeh stretnutia

- Našli sme viacero komunikačných, version control a scrum nástrojov ale nakoniec sme sa zhodli na:
 - komunikačný - HipChat
 - version control - BitBucket
 - scrum - Jira
- Nástroje sme hneď aj spojznili.
- Mišo Cihák každého navigoval, ako si naklonovať repozitár CatchAndMatch do RubyMine.

- Určovali sme "Issues" na 1. šprint - prešli sme každý issue a dopisovali popis. Všetky issues sú v angličtine:
 - Implement event flow for first prototype
 - Design event structure
 - Design link generation
 - Implement link generation
 - Design simple user identification
 - Implement user interface
 - Implement user identification
 - Design user interface
 - Create team page
 - Setup Ruby on Rails project
 - Setup server for team page
 - Setup database
 - Design data model
 - Create project structure
 - Setup private git repository
- estimácia jednotlivých issue

8.4 Zápisnica stretnutia č. 4

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	21.10.2014, 13:00 - 17:00
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.4.1 Osnova na stretnutie

- Zistiť ako dokážeme vyriešiť dizajn tak, aby produkt spĺňal základnú podmienku jednoduchosti použitia
- Prejsť navrhnutý základný objektový model
- Koncept architektúry
- DB maintenance
- Technická dokumentácia
- Prebrať usporiadanie stretnutia s potenciálnymi používateľmi (early adopters) pre získanie feedbacku a nápadov na design a používanie CatchAndMatch služby
- Testovanie aplikácie

8.4.2 Priebeh stretnutia

- prebrali a pozmenili sme navrhnutý dátový model, pričom sme sa zhodli, že jeho použitie bude spojené s prototypom, a následne bude upravený

- opätovne sme otvorili otázku UI, pričom sme sa zamerali hlavne na layout stránky - “vertikálny” vs. “kartičky”
- zhodnotili sme doterajší priebeh šprintu

8.5 Zápisnica stretnutia č. 5

Autor	Bc. Michal Cihák
Dátum a čas stretnutia	28.10.2014, 13:00 - 17:30
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.5.1 Osnova na stretnutie

- predviesť demo product owner-ovi
- retrospektíva - zhodnotenie prvého šprintu
- definovať "Issues" na ďalší šprint
- rozdeliť procesy na MSI/MIS (zadanie 2)

8.5.2 Priebeh stretnutia

- na základe retrospektívy sme odhalili problémy, ktoré vznikli počas prvého šprintu (viď dokument "Retrospektíva")
- dodefinovali sme požiadavky na prvý prototyp
 - Linky/Routing (C)
 - Voting -> List (V,C)
 - Link user to EventOptions (C)
 - "Copy to Clipboard" button (V)

- Add duration -> Hide DateTo (C,V)
- Improve (C,V)
 - dates
 - places
 - users - highlight, user selection
- taktiež sme prebrali nadchádzajúce úlohy
 - Event Place/Time representation in DB
 - investigate CI (Jenkins, Bamboo)
 - 17.11. - document
 - Landing page
 - Dev and Prod Enviroment
 - Organize UX session
 - How to finish Event flow?
 - How to represent temporary/logged user in DB

Legenda:

C - controller

V - view

8.6 Zápisnica stretnutia č. 6

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	4.11.2014 13:00 - 17:00
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.6.1 Osnova na stretnutie

- user votes
- front-end javascript framework
- CI

8.6.2 Priebeh stretnutia

- prebehla diskusia k prideleným procesom/metodikám
 - testovanie - Gabo
 - issue flow - Lukáš
 - VCS - Matúš
 - komunikácia - Pali
 - technická dokumentácia - Vlado
 - issue creation - Tomáš
- k uvedeným procesom budú vytvorené tasky v Jire

- prebehol daily stand-up meeting
- upravili sme šprint doplnením issues
- dohodli sme sa na definovaní niektorých procesov do konca týždňa - komunikácia, VCS, technická dokumentácia, stav projektu, issue flow

8.7 Zápisnica stretnutia č. 7

Autor	Bc. Vladimír Osvald
Dátum a čas stretnutia	11.11.2014, 13:00 - 17:00
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.7.1 Osnova na stretnutie

- prebrať “investigate tasks” z predchádzajúceho šprintu
- zadefinovať tasky na ďalší šprint
- demo, planning a retrospective meeting
- existujúce riešenie - DOODLE
- “Panely pre tímové projekty 2014-15”

8.7.2 Priebeh stretnutia

- aj napriek predchádzajúcej dohode, bolo z časových dôvodov prezentované demo až na tomto utorkovom stretnutí
- okrem prezentácie dema, boli prezentované výsledky z “investigate taskov” ako aj vypracované metodiky k procesom

- následne prebehla retrospektíva, ktorej výsledkom je kladné zhodnotenie práce a pozitívne dojmy z uplynulého šprintu
- zhodli sme sa, že nasledujúci šprint bude trvať len jeden týždeň
- čo sa týka produktu, dohodli sme sa, že pri výbere času je podstatný začiatok udalosti - ostatné informácie (koniec/trvanie) nie sú natoľko podstatne, resp. ich zatiaľ nebudeme špeciálne uvažovať a riešiť

8.8 Zápisnica stretnutia č. 8

Autor	Bc. Matúš Pikuliak
Dátum a čas stretnutia	18.11.2014,13:00 - 18:00
Miesto stretnutia	JOBSOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.8.1 Osnova na stretnutie

- odovzdávaná dokumentácia - kto čo urobil
- angular
- dôležitosť rolí pri eventoch
- long-term plan
- finalize event flow
- retrospektíva

8.8.2 Priebeh stretnutia

- prebrali sme vytvorené metodiky riadenia
- zamerali sme sa na finalizáciu dokumentácie, ktorú je potrebné odovzdať

- nakoľko išlo o týždňový šprint, prebehla retrospektíva, na základe ktorej sme sa rozhodli že aj ďalší šprint bude týždňový, aj napriek tomu, že nie všetky tasky z minulého šprintu boli dokončené
- aj napriek tomu, že zobrazenie stránky nefunguje správne, rozhodli sme sa neopravovať vzniknuté chyby, keďže samotný front-end sa bude ešte meniť
- základná myšlienka a požiadavka na produkt je, aby bol ASAP (as stupid as possible), to znamená, že musí byť v základnej verzii veľmi jednoduchý na ovládanie

8.9 Zápisnica stretnutia č. 9

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	25.11.2014, 13:00 - 16:00
Miesto stretnutia	JOB SOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.9.1 Osnova na stretnutie

- demo
- ďalší šprint
- retrospektíva
- prebrať/predniesť modul “rezervácia termínov”

8.9.2 Priebeh stretnutia

- oboznámili sme sa so stavom CI, pričom budeme používať nástroj na to určený (CruiseControl.cb)
- prebehlo demo produktu, v ktorom sme product ownerovi predviedli pokrok a nové funkcionality produktu, ako aj nový layout stránky s časovou osou
- prebehli rozsiahle diskusie o možnostiach autentifikácie používateľov, ako aj o možnostiach a spôsobe ukončenia eventu
- predstavili sme nový potenciál v oblasti využitia stránky - nový modul rezervácie miesta

v podnikoch (miesto pri stole v reštaurácií)

- prebrali a zadefinovali sme si tasky do tohto šprintu, ktorý sme predĺžili na dva týždne

8.10 Zápisnica stretnutia č. 10

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	2.12.2014, 13:00 - 16:00
Miesto stretnutia	JOB SOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.10.1 Osnova na stretnutie

- demo
- ďalší šprint
- retrospektíva
- zhodnotiť prezentáciu investorovi, ktorá prebehla v pondelok

8.10.2 Priebeh stretnutia

- odprezentovali sme pokrok na úlohách definovaných v Jire, avšak k odprezentovaniu dema nedošlo
- keďže išlo o náročný týždeň, a tasky po tomto šprinte (ktorý trval len 1 týždeň) neboli splnené v dostatočnej miere, ako aj v z iných organizačných pohnutí spojených priamo s predmetom *Tímový projekt* sme sa po dohode s vlastníkom produktu (a naším vedúcim) rozhodli predĺžiť aktuálny šprint o jeden týždeň

- na základe predchádzajúceho bodu sme doplnili Jiru (tento šprint) o ďalšie úlohy

8.11 Zápisnica stretnutia č. 11

Autor	Bc. Tomáš Sýkora
Dátum a čas stretnutia	9.12.2014, 13:00 - 16:00
Miesto stretnutia	JOB SOVO softvérové štúdio (1.31a)

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.11.1 Osnova na stretnutie

- demo po šprinte
- dohodnúť sa na postupe k dokončeniu dokumentácie
- upresniť smer, ktorým sa bude produkt uberať

8.11.2 Priebeh stretnutia

- zastavil sa u nás Alexander Vengrin a dal nám cenné rady k produktu a nasmeroval nás správnym smerom k finálnemu produktu
- pozreli sme videá, ako správne odprezentovať produkt aj bez fungujúcej verzie
- konzultovali sme dôležitosť timeliny
- spresnili sme základné vlastnosti, ktoré bude obsahovať verzia do konca decembra

8.12 Zápisnica stretnutia č. 12

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	16.2.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.12.1 Osnova na stretnutie

- Finálne sa dohodnúť na smerovaní projektu
- Rozdeliť náš cieľ na tasky
- Dohodnúť sa na taskoch v nasledujúcom šprinte
- Diskusia o dizajne pre landing page a event page

8.12.2 Priebeh stretnutia

SPRINT 7

- Redesign event page (places, mapka, participanti)
- Nová timeline na event page (zobrazenie, pridanie času, voting, selektovanie)
- Dizajn uzatvoreného eventu
- Dizajn Profil page

- IITsrc abstrakt
- Prezentácia pre ekonómov
- Event status on event page (Public, Private)
- CI na steroidoch
- Team page na steroidoch (najkrajšia zo všetkých)
- Navrhnut Topic / Tagy
- Navrhnuť model pre Suggestion
 - Time, Place, Tagy (Crawler služba)
 - Udalosti z viacerých kalendárov

VERZIE

Free verzia

- len jeden public kalendár (profil page)
- všetko okolo eventu
- prepojenie s google kalendárom
- follow ostatných

Business

- promote eventov
- rezervačný systém

Enterprise

PAGE

Event page

- event status: public, pre pozvaných, private (zobraziť že nemôže, nie konkrétne)
- timeline dorobiť
 - suggestions (add, remove)
 - voting
 - rozťahnutie na mesačný / týžňový náhľad

- zoznam časových návrhov
- mapka
 - nefunguje
 - suggestions
- topic / tags / filtre (Food, Fun, Sport, Business, School, Personal)
- uzatvorenie eventu
- matcher

Profil page

- zobrazit' len mesačný kalendár (viacero kalendárov)
- following (len udalosti, ktoré vlastní)
- owned
- zobrazit' nadchádzajúce udalosti ako zoznam?
- žiadne places
- správa profilu

Search profiles

DEADLINE

- MAREC - free verzia - 1.0
- APRÍL - mesiac ladenia, pridávania funkcionality, nové releasy
- MÁJ - ladit' timeline

Otazky:

Mobil verzia - responzivita?

- Pali vravi, ze to pojde

Login?

- email?
- prepojenie s FB, G+

Gmail calendar?

Hosting?

8.13 Zázpisnica stretnutia č. 13

Autor	Bc. Tomáš Sýkora
Dátum a čas stretnutia	23.2.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.13.1 Osnova na stretnutie

- Prediskutovanie príspevku na IIT.SRC
- Stretnutie s ekonómami

8.13.2 Priebeh stretnutia

- Diskusia k príspevku na IIT.SRC a dopísanie abstraktu.
- Príprava na prezentáciu pre EUBA
- Debata o funkcionalite a návrhu komponentu kalendár pre našu aplikáciu

Celé stretnutie malo v podstate dve hlavné témy. Prvou bola príprava na prezentáciu nášho produktu kolegom z EUBA. Druhou bol abstrakt, ktorým sme chceli zaujať, v tomto prípade však v rámci študentskej vedeckej konferencie IIT.SRC. Doladili sme tak posledné detaily v prezentácií a spoločne sa pustili aj do písania spomínaného abstraktu.

8.14 Zázpisnica stretnutia č. 14

Autor	Bc. Lukáš Masár
Dátum a čas stretnutia	1.4.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.14.1 Osnova na stretnutie

- článok na blog.
- ekonómovia, ako ďalej s nimi
 - ukázať im produkt
 - biznis verziu sa popýtať
 - free verzia – propagácia
- feedback na design
- IIT SRC - poster a podobne
- mail od p. prof. Bielikovej
- Web stránka update
- Demo 1.4.2015

8.14.2 Priebeh stretnutia

- vďaka minulotýždňovej prezentácii sme nadviazali “spojenectvo” s kolegami z EUBA, ktorí v rámci svojho predmetu boli pripravení reagovať na naše požiadavky
- prebehla preto diskusia o tom, čo pre nás môžu urobiť
 - poskytli sme im materiály, ktoré sme získali/vytvorili v rámci našej predchádzajúcej činnosti
 - zhodli sme sa na tom, že v rámci svojho prostredia, pre nás vykonajú prieskum trhu
- v priebehu uplynulého týždňa bol náš produkt prezentovaný kolegom z odboru, vďaka ktorým sme získali cenný feedback, na základe ktorého sme vytvorili nové úlohy
- prediskutovali sme mail, a určili si zodpovedného za vyplnenie a odoslanie dotazníka v rámci TP CUP-u
- diskutovali sme o blížiacej sa študentskej vedeckej konferencii IIT.SRC, pričom sme si povedali o tom, čo všetko chceme stihnúť

8.15 Zápisnica stretnutia č. 15

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	8.4.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.15.1 Osnova na stretnutie

- článok na blog
- dotazník pre p. prof. Bielikovú

HAPPEN:

- zoradenie časov podľa počtu hlasov – neprehľadné
- pridanie času - neviem pridať čas na inú, ako celú hodinu
- pridanie času - pri vpisovaní času automaticky prechádzať z hodín do minút
- odstrániť scrollbary pri zobrazení všetkých, ktorí hlasovali za čas eventu + pri tomto pohľade odstrániť číslo + x attending - prehodnotiť škálu + získavanie informácií
- Filtrovanie zoznamu podľa (asc aj desc):
 - času
 - počtu hlasov

8.15.2 Priebek stretnutia

- na stretnutí sa nám podarilo prebrať a prediskutovať všetky vyššie uvedené body resp. nedostatky
 - úlohy z nich vyplývajúce boli zadané do Jiry
- s blížiacou sa vedeckou konferenciou je potrebné spracovať článok na blog robíme.it
 - ujasnili sme si preto čo a ako chceme na blogu prezentovať
- bol upravený resp. dokončený a schválený dotazník týkajúci sa nášho tímu a produktu
- Chalani vo Švédsku pracovali na novom dizajne, ktorý nám dnes predviedli
 - tento dizajn mal úspech
 - objavili sme na ňom pár nedostatkov
 - taktiež nie všetky elementy boli už v tomto dizajne
 - každopádne sme sa rozhodli, že ide o dizajn, ktorý budeme na IIT.SRC prezentovať, a dôjde už len k jeho dotiahnutiu

8.16 Zázpisnica stretnutia č. 16

Autor	Bc. Lukáš Masár
Dátum a čas stretnutia	15.4.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.16.1 Osnova na stretnutie

- demo
- článok na blog robime.it
- návrh plagátu na IIT.SRC
- diskusia o novom dizajne

8.16.2 Priebeh stretnutia

- ukázali sme, čo je nové a pozmenené v našej aplikácii
- dohodli sme si tasky do ďalšieho šprintu
- prebrali sme, o čom chceme písať v článku na blog
- na základe posledných zmien v dizajne sme
 - navrhli možné vylepšenia
 - určili, čo by sa mohlo odstrániť

- diskutovali sme o možnostiach prezentácie a propagácie nášho produktu na IIT.SRC
 - plagát
 - prezentácia na monitore pomocou automatického skrolovania landing page
 - doska s logom Happen – vyhotovenie vo FabLabe
 - možno používateľská príručka?

8.17 Zázpisnica stretnutia č. 17

Autor	Bc. Tomáš Sýkora
Dátum a čas stretnutia	22.4.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.17.1 Osnova na stretnutie

- IIT.SRC

8.17.2 Priebeh stretnutia

- toto stretnutie malo len jedinú tému, ktorou bolo zajtrajšie IIT.SRC
- dohodli sme sa na spôsobe prezentácie projektu
- pozreli sme poster, a doladili na ňom, posledné detaily
- prebrali sme taktiež výsledky z používateľského tetovania, a určili sme posledné nedostatky, ktoré do zajtra odstránime
- spustili sme prevádzku Beta verzie Happen pre verejnosť
- povedali sme si organizačné pokyny na zajtra

8.18 Zázpisnica stretnutia č. 18

Autor	Bc. Lukáš Masár
Dátum a čas stretnutia	29.4.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.18.1 Osnova na stretnutie

- IIT SRC – zhodnotenie
- spracovanie pripomienok získaných na konferenciách
- Čo stihneme do konca semestra?
- Aká teda bude business verzia?

8.18.2 Priebeh stretnutia

- IIT SRC nám prinieslo cennú spätnú väzbu, ktorú sme na dnešnom stretnutí prediskutovali
- prebrali sme funkcionalitu, ktorá v našej aplikácii doposiaľ nie je, bola však častým terčom otázok účastníkov IIT SRC
 - responzivnosť
 - integrácia s Google kalendárom (resp. iným kalendárom)

- nové možnosti Business verzie
 - naše návrhy Business verzií, nie všetkých porotcov oslovili, neboli však určite úplne zlé, a potešila nás snaha porotcov, pomôcť nám pri špecifikácii potenciálneho trhu a teda aj funkčnosti aplikácie
- rozhodli sme sa, že do konca semestra v prvom rade odladíme súčasnú verziu aplikácie - o niektorých chybách sme vedeli už dávnejšie, niektoré sa vyskytli aj počas prezentácie na IIT.SRC a očakávame, že nejaké nedostatky budú odhalené na základe ďalšieho používateľského testovania
- prebrali sme návrhy a pripomienky ohľadne Business verzie

8.19 Zápisnica stretnutia č. 19

Autor	Bc. Michal Gajdoš
Dátum a čas stretnutia	11.5.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.19.1 Osnova na stretnutie

- Spätná väzba IIT.SRC
- Demo
- nová funkcionality? alebo len odstránenie bugov
- naplánovať väčšie testovanie

8.19.2 Priebeh stretnutia

- prebehlo Demo
- opätovne sme sa bližšie pozreli na nedostatky aplikácie, ktoré je potrebné odstrániť
- rozhodli sme sa pridať možnosť rozosielania pozvánok resp. minimálne notifikácií o udalosti, v rámci aplikácie a jej používateľmi
- dostali sme priamu spätnú väzbu od porotcov z IIT SRC
 - potešili nás kladné reakcie na Happen

- v rámci negatívnych vlastností nám je ľúto, že sme nedokázali niektoré veci povedať lepšie
 - dúfame, že dostaneme možnosť aspoň na niektoré z negatívnych ohlasov reagovať pri postupe do semifinále súťaže
- naplánovali sme si, že do konca semestra spravíme skupinové testovanie

8.20 Zápisnica stretnutia č. 20

Autor	Bc. Tomáš Sýkora
Dátum a čas stretnutia	18.5.2015, 10:00 - 13:00
Miesto stretnutia	4.20

Prítomní

- Ing. Lukáš Turský
- Bc. Michal Cihák (online)
- Bc. Michal Gajdoš
- Bc. Lukáš Masár
- Bc. Pavol Michálek (online)
- Bc. Vladimír Osvald
- Bc. Matúš Pikuliak
- Bc. Tomáš Sýkora

8.20.1 Osnova na stretnutie

- demo
- záverečná správa
- technická dokumentácia
- prebrať výsledky z používateľského testovania

8.20.2 Priebeh stretnutia

- prebrali sme bug fixy, ktoré boli opravené v poslednom šprinte, konkrétne
 - login
 - drobné úpravy layoutu
 - zoradenie navrhovaných časov podľa času
 - zakázanie pridania času/miesta, ako aj hlasovania, pokiaľ používateľ nie je prihlásený

- fix pridávania viacerých časov
- ukázali sme si novú funkcionálnosť, ktorou bolo pozývanie ľudí na udalosť priamo cez stránku tejto udalosti a v rámci Happen
- povedali sme si o priebehu a výsledku používateľského testovania
- dohodli sme sa na ďalšom postupe a určili si úlohy do ďalšieho obdobia

9 Exporty evidencie úloh

V tejto časti príloh sa nachádzajú exporty evidencie úloh k jednotlivým šprintom.

9.1 Prvý šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	CATCHME-21		Create application form for TP Cup 2015	Vladimir Osvaid	Lukas Tursky	↑	Closed	Fixed	21/Oct/14	28/Oct/14	28/Oct/14
	CATCHME-20		Design data model	Matus Pikuliak	Michal Cihak	↑	Closed	Fixed	14/Oct/14	28/Oct/14	
	CATCHME-19		Implement User Identification	Lukas Tursky	Michal Cihak	↑	Closed	Fixed	14/Oct/14	28/Oct/14	
	CATCHME-18		Implement Link Generation	Unassigned	Michal Cihak	↑	Closed	Fixed	14/Oct/14	28/Oct/14	
	CATCHME-17		Implement User Interface	Matus Pikuliak	Michal Cihak	↑	Closed	Fixed	14/Oct/14	28/Oct/14	
	CATCHME-16		Implement event flow for first prototype	Michal Cihak	Michal Cihak	↑	Closed	Fixed	14/Oct/14	28/Oct/14	
	CATCHME-15		Setup RoR Project	Michal Cihak	Matus Pikuliak	↑	Closed	Fixed	09/Oct/14	25/Oct/14	
	CATCHME-14		Design simple user identification	Unassigned	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	28/Oct/14	
	CATCHME-13		Design Link generation	Tomas Sykora	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	28/Oct/14	
	CATCHME-12		Design User Interface	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	28/Oct/14	
	CATCHME-11		Setup DB	Lukas Masar	Lukas Tursky	↑	Closed	Done	08/Oct/14	25/Oct/14	
	CATCHME-10		Design Event structure	Unassigned	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	28/Oct/14	
	CATCHME-8		Create Team page	Pavol Michalek	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	28/Oct/14	
	CATCHME-7		Create project structure	Michal Cihak	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	25/Oct/14	
	CATCHME-4		Setup server for team page	Lukas Masar	Lukas Tursky	↑	Closed	Done	08/Oct/14	19/Oct/14	
	CATCHME-1		Setup private Git repository	Lukas Tursky	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	14/Oct/14	

Obr. D.1. Export evidencie úloh pre prvý šprint

9.2 Druhý šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	CATCHME-40		[P] Propose version management process	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	09/Nov/14	12/Nov/14	
	CATCHME-37		[?] Represantion of accounts/permissions/roles	Michal Gajdos	Michal Cihak	↑	Closed	Done	28/Oct/14	12/Nov/14	
	CATCHME-36		[UX] Organize UX session	Lukas Tursky	Michal Cihak	↑	Open	Unresolved	28/Oct/14	04/Nov/14	
	CATCHME-35		[S] Setup production environment	Lukas Masar	Michal Cihak	↑	Closed	Incomplete	28/Oct/14	12/Nov/14	
	CATCHME-33		[I] Odovzдание dokumentacie 17.11.	Unassigned	Michal Cihak	↑	Open	Unresolved	28/Oct/14	17/Nov/14	17/11/14
	CATCHME-30		[V] Improve Places	Michal Gajdos	Michal Cihak	↑	Closed	Done	28/Oct/14	12/Nov/14	
	CATCHME-29		[V] Improve Dates	Pavol Michalek	Michal Cihak	↑	Closed	Done	28/Oct/14	12/Nov/14	
	CATCHME-28		[V] Copy to clipboard	Matus Pikuliak	Michal Cihak	↑	Closed	Fixed	28/Oct/14	05/Nov/14	
	CATCHME-27		[V] Prepare few layout options	Michal Gajdos	Michal Cihak	↑	Closed	Incomplete	28/Oct/14	11/Nov/14	
	CATCHME-26		[C] Link Generation	Matus Pikuliak	Michal Cihak	↑	Closed	Fixed	28/Oct/14	10/Nov/14	
	CATCHME-25		[V] Voting	Pavol Michalek	Michal Cihak	↑	Closed	Done	28/Oct/14	11/Nov/14	
	CATCHME-24		[C] Voting	Michal Cihak	Michal Cihak	↑	Closed	Done	28/Oct/14	11/Nov/14	
	CATCHME-23		[V] User Selection	Pavol Michalek	Michal Cihak	↑	Closed	Done	28/Oct/14	11/Nov/14	
	CATCHME-22		[C] Link User to Event Options	Michal Cihak	Michal Cihak	↑	Closed	Done	28/Oct/14	11/Nov/14	
	CATCHME-2		Agree on documentation structure and tool	Vladimir Osvald	Lukas Tursky	↑	Closed	Fixed	08/Oct/14	12/Nov/14	

Obr. D.2. Export evidencie úloh pre druhý šprint

9.3 Tretí šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Di
<input checked="" type="checkbox"/>	CATCHME-49		[Proces] Improve Version Management	Matus Pikuliak	Matus Pikuliak	↑	Resolved	Fixed	14/Nov/14	16/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-48		[V] Fix list of event times	Unassigned	Michal Cihak	↑	Open	Unresolved	11/Nov/14	11/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-47		[C/V] Add Duration	Unassigned	Michal Cihak	↑	Open	Unresolved	11/Nov/14	11/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-46		[V] Create page	Unassigned	Michal Cihak	↑	Open	Unresolved	11/Nov/14	11/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-45		[Proces] Code Documentation	Michal Gajdos	Michal Cihak	↑	Resolved	Fixed	11/Nov/14	18/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-44		[Proces] Issue Flow	Tomas Sykora	Michal Cihak	↑	Resolved	Fixed	11/Nov/14	16/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-43		[Proces] Communication	Pavol Michalek	Michal Cihak	↑	Resolved	Done	11/Nov/14	18/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-42		[Proces] Code Review	Lukas Masar	Michal Cihak	↑	Resolved	Done	11/Nov/14	17/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-41		[Proces] Testovanie	Michal Cihak	Michal Cihak	↑	Resolved	Fixed	11/Nov/14	18/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-33		[I] Odovzdanie dokumentacie 17.11.	Unassigned	Michal Cihak	↑	Open	Unresolved	28/Oct/14	17/Nov/14	1
<input checked="" type="checkbox"/>	CATCHME-32		[S] Implement CI solution	Lukas Masar	Michal Cihak	↑	Resolved	Done	28/Oct/14	17/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-3		[?] Investigate Angular	Matus Pikuliak	Lukas Tursky	↑	Resolved	Fixed	08/Oct/14	17/Nov/14	











Obr. D.3. Export evidencie úloh pre tretí šprint

9.4 Štvrtý šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
<input checked="" type="checkbox"/>	CATCHME-62		Dokumentacia riadenia a inzinierske dielo na Web z 1. kontrolneho bodu na webe	Lukas Tursky	Lukas Tursky	↑	Closed	Fixed	29/Nov/14	09/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-59		(Think of) Matcher	Vladimir Osvald	Lukas Tursky	↑	Closed	Fixed	27/Nov/14	04/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-58		Decide event lifecycle end options	Lukas Tursky	Lukas Tursky	↑	Closed	Fixed	27/Nov/14	04/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-57		Possibility to update name and description of created event	Pavol Michalek	Lukas Tursky	↑	Closed	Fixed	22/Nov/14	09/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-56		[V] Timeline improvement	Pavol Michalek	Pavol Michalek	↑	Closed	Incomplete	20/Nov/14	10/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-55		[V/C] Refactoring	Matus Pikuliak	Michal Cihak	↑	Closed	Fixed	19/Nov/14	04/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-54		[I] 1.12. Prezentacia	Tomas Sykora	Michal Cihak	⊘	Closed	Fixed	19/Nov/14	02/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-53		[I] 24.11. Business Model Canvas & Customer development	Michal Gajdos	Michal Cihak	↑	Closed	Fixed	19/Nov/14	25/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-52		(Think of) Preparation for design session	Lukas Tursky	Lukas Tursky	↑	Closed	Fixed	19/Nov/14	02/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-51		User registration support	Michal Cihak	Lukas Tursky	↑	Closed	Incomplete	19/Nov/14	10/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-50		Add event states to app	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	19/Nov/14	04/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-48		[V] Fix showing of first element in list of event times	Pavol Michalek	Michal Cihak	↑	Closed	Fixed	11/Nov/14	04/Dec/14	
<input checked="" type="checkbox"/>	CATCHME-46		[V] Simplify creation page	Pavol Michalek	Michal Cihak	↑	Closed	Done	11/Nov/14	23/Nov/14	
<input checked="" type="checkbox"/>	CATCHME-32		[S] Implement CI solution	Lukas Masar	Michal Cihak	↑	Closed	Done	28/Oct/14	02/Dec/14	

Obr. D.4. Export evidencie úloh pre štvrtý šprint

9.5 Piaty šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	 CATCHME-70		Create User profile page	Michal Gajdos	Lukas Tursky	↑	Closed	Incomplete	02/Dec/14	10/Dec/14	
	 CATCHME-69		Real time data synchronization on client site	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	02/Dec/14	08/Dec/14	
	 CATCHME-67		Deploy current functional version to Prod env	Lukas Masar	Lukas Tursky	↑	Closed	Fixed	02/Dec/14	09/Dec/14	
	 CATCHME-65		Change Voting to just Yes Selection	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	02/Dec/14	08/Dec/14	
	 CATCHME-64		Everybody put photo as avatar in Jira	<i>Unassigned</i>	Lukas Tursky	↑	Closed	Fixed	02/Dec/14	09/Dec/14	
	 CATCHME-63		Introduce versioning for easier deployment of stable version on Prod env	Pavol Michalek	Lukas Tursky	↑	Closed	Fixed	30/Nov/14	09/Dec/14	
	 CATCHME-62		Dokumentacia riadenia a inzinierske dielo na Web z 1. kontrolneho bodu na webe	Lukas Tursky	Lukas Tursky	↑	Closed	Fixed	29/Nov/14	09/Dec/14	
	 CATCHME-57		Possibility to update name and description of created event	Pavol Michalek	Lukas Tursky	↑	Closed	Fixed	22/Nov/14	09/Dec/14	
	 CATCHME-56		[V] Timeline improvement	Pavol Michalek	Pavol Michalek	↑	Closed	Incomplete	20/Nov/14	10/Dec/14	
	 CATCHME-51		User registration support	Michal Cihak	Lukas Tursky	↑	Closed	Incomplete	19/Nov/14	10/Dec/14	

Obr. D.5. Export evidencie úloh pre piaty šprint

9.6 Šiesty šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
☒	CATCHME-111		Ako .gitignorovat database.yml	Tomas Sykora	Matus Pikuliak	↑	Closed	Fixed	02/Mar/15	04/Mar/15	
☒	CATCHME-108		Rozumovanie nad prácou s časom	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	26/Feb/15	28/Feb/15	
☒	CATCHME-106		[C] Landing Page	<i>Unassigned</i>	Michal Cihak	↑	Closed	Fixed	19/Feb/15	04/Mar/15	
☒	CATCHME-105		[V] Landing Page	Vladimir Osvald	Michal Cihak	↑	Closed	Fixed	19/Feb/15	04/Mar/15	
☒	CATCHME-104		[V] Event page frontend v1	<i>Unassigned</i>	Michal Cihak	↑	Closed	Duplicate	19/Feb/15	20/Feb/15	
☒	CATCHME-103		[C] Event page backend	<i>Unassigned</i>	Michal Cihak	↑	Closed	Fixed	19/Feb/15	04/Mar/15	
☒	CATCHME-102		[V] Profile Page	Lukas Masar	Michal Cihak	↑	Closed	Fixed	19/Feb/15	04/Mar/15	
☒	CATCHME-101		[C] Profile Page	Michal Cihak	Michal Cihak	↑	Closed	Fixed	19/Feb/15	04/Mar/15	
☒	CATCHME-97		team page na steroidoch	Tomas Sykora	Lukas Masar	↑	Closed	Fixed	18/Feb/15	23/Feb/15	
☒	CATCHME-96		CI na steroidoch	Lukas Masar	Lukas Masar	↑	Closed	Fixed	18/Feb/15	04/Mar/15	
☒	CATCHME-95		prezentacia pre ekonomov	Lukas Masar	Lukas Masar	↑	Closed	Fixed	18/Feb/15	23/Feb/15	
+	CATCHME-92		Ribbon a.k.a. kalendár	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	18/Feb/15	24/Feb/15	
☒	CATCHME-91		IIT SRC - abstract	Lukas Masar	Lukas Masar	↑	Closed	Done	16/Feb/15	22/Feb/15	
☒	CATCHME-90		[B] da sa pridať ten istý mail	<i>Unassigned</i>	Lukas Masar	↑	Closed	Won't Fix	15/Feb/15	20/Feb/15	
☒	CATCHME-87		Napísať nové maily	Vladimir Osvald	Matus Pikuliak	↑	Closed	Fixed	08/Feb/15	04/Mar/15	
☒	CATCHME-76		Vytvoriť dizajn	Pavol Michalek	Lukas Tursky	↑	Closed	Fixed	18/Jan/15	04/Mar/15	
☒	CATCHME-75		Zobrazovať verziu aplikácie	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	18/Jan/15	21/Feb/15	

Obr. D.6. Export evidencie úloh pre šiesty šprint

9.7 Siedmy šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
🔴	CATCHME-88		Uzalvaranie eventu nefunguje ak je menej ako 3 možnosti	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	08/Feb/15	09/Feb/15	
🟢	CATCHME-85		Event closing notification	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	08/Feb/15	08/Feb/15	
🟡	CATCHME-84		Sending mail while adding participant takes too long	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	08/Feb/15	08/Feb/15	
🔴	CATCHME-82		Timeline voting doesn't work	<i>Unassigned</i>	Matus Pikuliak	↑	Closed	Won't Fix	22/Jan/15	18/Feb/15	
🔴	CATCHME-81		Copy-to-clipboard button doesn't work	Matus Pikuliak	Matus Pikuliak	↓	Closed	Fixed	22/Jan/15	18/Feb/15	
🟢	CATCHME-80		Create alias in HipChat for mail notifications	Tomas Sykora	Tomas Sykora	↑	Closed	Fixed	18/Jan/15	08/Feb/15	
🟢	CATCHME-79		Navrhnut zobrazenie casu pre rezervacny system	Michal Gajdos	Lukas Tursky	↑	Closed	Fixed	18/Jan/15	04/Mar/15	
🟢	CATCHME-76		Vytvorit' dizajn	Pavol Michalek	Lukas Tursky	↑	Closed	Fixed	18/Jan/15	04/Mar/15	
🔴	CATCHME-74		Upravit' login/signup page do peknej podoby	<i>Unassigned</i>	Pavol Michalek	↑	Closed	Won't Fix	29/Dec/14	24/Feb/15	
🔴	CATCHME-73		Nejde pridať čas na novom evente	Pavol Michalek	Pavol Michalek	🚫	Closed	Fixed	29/Dec/14	08/Feb/15	
🔴	CATCHME-72		Skryť funkcionality neprístupnú používateľom	Matus Pikuliak	Pavol Michalek	↑	Closed	Fixed	29/Dec/14	08/Feb/15	
🟢	CATCHME-68		Dokumentacia k ZS	<i>Unassigned</i>	Lukas Tursky	↑	Closed	Fixed	02/Dec/14	08/Feb/15	
🟢	CATCHME-66		Implement event closing	<i>Unassigned</i>	Lukas Tursky	↑	Closed	Fixed	02/Dec/14	08/Feb/15	
🟢	CATCHME-61		[?] Investigate integration with FB and Google accounts	Tomas Sykora	Lukas Tursky	↑	Closed	Fixed	27/Nov/14	02/Apr/15	
🟡	CATCHME-38		Create Business model canvas	<i>Unassigned</i>	Lukas Tursky	↑	Closed	Duplicate	04/Nov/14	08/Feb/15	01/De

Obr. D.7. Export evidencie úloh pre siedmy šprint

9.8 Ôsmy šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated
☐	CATCHME-121		Vyrobit v db.seeds development data	Lukas Masar	Matus Pikuliak	↑	Closed	Incomplete	04/Mar/15	20/Mar/15
☐	CATCHME-120		Vyriesit ekonomov	Tomas Sykora	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	20/Mar/15
☐	CATCHME-119		Prezentacny panel	Lukas Masar	Matus Pikuliak	↑	Closed	Done	04/Mar/15	09/Mar/15
☐	CATCHME-117		Nastylovat landing page	Vladimir Osvald	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	20/Mar/15
☐	CATCHME-116		Users/Following	Michal Cihak	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	02/Apr/15
☐	CATCHME-115		Opravit mapu	Michal Gajdos	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	30/Apr/15
☐	CATCHME-114		Pridavanie casov naparovat na kalendar	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	20/Mar/15
☐	CATCHME-112		Zobrazovat hlasy v hlasovani	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	20/Mar/15
☐	CATCHME-109		Nastavit Asset Pipeline	Lukas Masar	Matus Pikuliak	↓	In Progress	Unresolved	01/Mar/15	14/Apr/15
☐	CATCHME-61		[?] Investigate integration with FB and Google accounts	Tomas Sykora	Lukas Tursky	↑	Closed	Fixed	27/Nov/14	02/Apr/15

Obr. D.8. Export evidencie úloh pre ôsmy šprint

9.9 Deviaty šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated
☑	CATCHME-132		Upratat resources	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	29/Mar/15	02/Apr/15
☑	CATCHME-131		Suggesting	Michal Cihak	Michal Cihak	↑	Closed	Fixed	24/Mar/15	02/Apr/15
☑	CATCHME-130		Going	Michal Cihak	Michal Cihak	↑	Closed	Fixed	24/Mar/15	02/Apr/15
☑	CATCHME-129		Hlasovanie	Michal Cihak	Michal Cihak	↑	Closed	Fixed	24/Mar/15	02/Apr/15
☑	CATCHME-128		IIT SRC dokument	Michal Gajdos	Tomas Sykora	↑	Closed	Fixed	20/Mar/15	23/Mar/15
☑	CATCHME-127		Opravit sortovanie po pridavani eventov	Matus Pikuliak	Vladimir Osvald	↑	Closed	Fixed	20/Mar/15	02/Apr/15
☑	CATCHME-126		Uzatvorenie udalosti	Pavol Michalek	Matus Pikuliak	↑	Closed	Done	11/Mar/15	15/Apr/15
☑	CATCHME-124		Úpravy dizajnu	Michal Cihak	Matus Pikuliak	↑	Closed	Fixed	11/Mar/15	29/Apr/15
☑	CATCHME-122		Zobrazovanie dat v kalendari	Pavol Michalek	Matus Pikuliak	↑	Closed	Done	04/Mar/15	02/Apr/15
☑	CATCHME-116		Users/Following	Michal Cihak	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	02/Apr/15
☑	CATCHME-115		Opravit mapu	Michal Gajdos	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	30/Apr/15
☑	CATCHME-113		Urobit share komponent	Pavol Michalek	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	27/Apr/15
☑	CATCHME-109		Nastavit Asset Pipeline	Lukas Masar	Matus Pikuliak	↓	In Progress	Unresolved	01/Mar/15	14/Apr/15
☑	CATCHME-78		Zaviest testovanie	Unassigned	Lukas Tursky	↑	Closed	Duplicate	18/Jan/15	02/Apr/15
☑	CATCHME-61		[?] Investigate integration with FB and Google accounts	Tomas Sykora	Lukas Tursky	↑	Closed	Fixed	27/Nov/14	02/Apr/15

Obr. D.9. Export evidencie úloh pre deviaty šprint

9.10 Desiaty šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Du
+	CATCHME-155		Pridanie dalsich guestov (invite)	Michal Cihak	Pavol Michalek	↑	Closed	Fixed	14/Apr/15	20/May/15	
▣	CATCHME-153		Suggestions miest obcas nefunguju	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	21/Apr/15	
▣	CATCHME-152		Nespravny datum pri pridani casu PLUSkom	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	21/Apr/15	
▣	CATCHME-151		Nepridava pouzivatela do zoznamu eventu	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	17/Apr/15	
▣	CATCHME-150		Stale zobrazene pridanie casu	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	17/Apr/15	
▣	CATCHME-149		Pridavanie rovnakych casov	Matus Pikuliak	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	20/May/15	
▣	CATCHME-148		Pridanie casu do kalendara	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	17/Apr/15	
▣	CATCHME-147		Nespravne prihlasovacie udaje	Matus Pikuliak	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	27/Apr/15	
▣	CATCHME-144		Clanok na blog	Michal Gajdos	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	20/Apr/15	
▣	CATCHME-143		Testovanie	Vladimir Osvald	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	17/Apr/15	
▣	CATCHME-142		Percentualne vyjadrenie uskutocnenia eventu	Lukas Masar	Tomas Sykora	↑	Closed	Incomplete	02/Apr/15	10/Apr/15	
▣	CATCHME-141		Vyhľadavanie	Pavol Michalek	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	10/Apr/15	
▣	CATCHME-138		Landing page - predstavenie produktu	Michal Cihak	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	15/Apr/15	
▣	CATCHME-137		Komponent na feedback	Tomas Sykora	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	27/Apr/15	
▣	CATCHME-134		Funkcnost nastaveni	Lukas Masar	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	20/May/15	
▣	CATCHME-126		Uzatvorenie udalosti	Pavol Michalek	Matus Pikuliak	↑	Closed	Done	11/Mar/15	15/Apr/15	
▣	CATCHME-124		Úpravy dizajnu	Michal Cihak	Matus Pikuliak	↑	Closed	Fixed	11/Mar/15	29/Apr/15	
▣	CATCHME-115		Opravit mapu	Michal Gajdos	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	30/Apr/15	
▣	CATCHME-113		Urobiť share komponent	Pavol Michalek	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	27/Apr/15	
▣	CATCHME-107		Frontendové validácie	Pavol Michalek	Michal Cihak	↑	Closed	Fixed	19/Feb/15	15/Apr/15	

Obr. D.10. Export evidencie úloh pre desiaty šprint

9.11 Jedenásty šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Du
	CATCHME-165		Search nejde	Unassigned	Matus Pikuliak	↑	Closed	Fixed	20/Apr/15	20/Apr/15	
	CATCHME-164		Closing zoznamy su rozhadzane	Pavol Michalek	Matus Pikuliak	↑	Closed	Fixed	20/Apr/15	27/Apr/15	
	CATCHME-163		Doplnit texty na prazdne miesta	Unassigned	Matus Pikuliak	↑	Open	Unresolved	20/Apr/15	20/May/15	
	CATCHME-162		Schovat moznost lyzdenneho zobrazenia v kalendari	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	20/Apr/15	27/Apr/15	
	CATCHME-161		Opravit maily	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	20/Apr/15	27/Apr/15	
	CATCHME-160		Dlhy text v description preteka cez kalendar	Matus Pikuliak	Michal Gajdos	↑	Closed	Fixed	20/Apr/15	27/Apr/15	
	CATCHME-159		Custom scrollbar	Vladimir Osvald	Pavol Michalek	↑	Closed	Fixed	18/Apr/15	20/May/15	
	CATCHME-158		Upratovacie prace na CLOSING	Matus Pikuliak	Pavol Michalek	↑	Closed	Fixed	18/Apr/15	27/Apr/15	
	CATCHME-157		Zarovnavanie casov ??	Unassigned	Matus Pikuliak	↑	Closed	Fixed	15/Apr/15	20/Apr/15	
	CATCHME-156		Weekly calendar (events)	Unassigned	Pavol Michalek	↑	Open	Unresolved	14/Apr/15	20/May/15	
	CATCHME-155		Pridanie dalsich guestov (invite)	Michal Cihak	Pavol Michalek	↑	Closed	Fixed	14/Apr/15	20/May/15	
	CATCHME-154		Reopen event	Pavol Michalek	Matus Pikuliak	↑	Closed	Fixed	13/Apr/15	21/Apr/15	
	CATCHME-153		Suggestions miest obcas nefunguju	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	21/Apr/15	
	CATCHME-152		Nespravny datum pri pridani casu PLUSkom	Pavol Michalek	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	21/Apr/15	
	CATCHME-149		Pridavanie rovnakych casov	Matus Pikuliak	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	20/May/15	
	CATCHME-147		Nespravne prihlasovacie udaje	Matus Pikuliak	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	27/Apr/15	
	CATCHME-146		Mailly sa neposielaju	Michal Cihak	Tomas Sykora	↓	Closed	Fixed	08/Apr/15	20/Apr/15	
	CATCHME-144		Clanok na blog	Michal Gajdos	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	20/Apr/15	
	CATCHME-137		Komponent na feedback	Tomas Sykora	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	27/Apr/15	
	CATCHME-135		Rozlisovanie eventov na profilovej stranke	Unassigned	Tomas Sykora	↓	Closed	Fixed	02/Apr/15	20/Apr/15	
	CATCHME-134		Funkcnost nastaveni	Lukas Masar	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	20/May/15	
	CATCHME-124		Úpravy dizajnu	Michal Cihak	Matus Pikuliak	↑	Closed	Fixed	11/Mar/15	29/Apr/15	
	CATCHME-115		Opravit mapu	Michal Gajdos	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	30/Apr/15	
	CATCHME-113		Urobit share komponent	Pavol Michalek	Matus Pikuliak	↑	Closed	Fixed	04/Mar/15	27/Apr/15	

Obr. D.11. Export evidencie úloh pre jedenásty šprint

9.12 Dvanásť šprint

T	Key	Components	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Du
🔴	CATCHME-173		Description update style	Matus Pikuliak	Matus Pikuliak	↑	Closed	Fixed	14/May/15	20/May/15	
🔵	CATCHME-172		Optimalizacia aplikacie	Unassigned	Tomas Sykora	↑	Open	Unresolved	29/Apr/15	20/May/15	
🟢	CATCHME-171		Moznost zrusit / zmazat vytvarany event	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	28/Apr/15	20/May/15	
🟢	CATCHME-169		Pridat notifikacie do aplikacie	Unassigned	Lukas Tursky	↑	Open	Unresolved	23/Apr/15	20/May/15	
🔴	CATCHME-168		Mozem robit suggestions aj ked nie som nalogovany, staci ze mam link. Ale korektne nemozem spravit vote.	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	23/Apr/15	20/May/15	
🔴	CATCHME-167		Event page layout buggy	Matus Pikuliak	Lukas Tursky	↑	Closed	Fixed	23/Apr/15	20/May/15	
🔴	CATCHME-166		Pri pokuse o vytvorenie eventu bez prihlasenia vyhodi chybu	Unassigned	Vladimir Osvald	↑	Open	Unresolved	22/Apr/15	20/May/15	
🔵	CATCHME-163		Doplnit texty na prazdne miesta	Unassigned	Matus Pikuliak	↑	Open	Unresolved	20/Apr/15	20/May/15	
🔴	CATCHME-159		Custom scrollbar	Vladimir Osvald	Pavol Michalek	↑	Closed	Fixed	18/Apr/15	20/May/15	
🔵	CATCHME-156		Weekly calendar (events)	Unassigned	Pavol Michalek	↑	Open	Unresolved	14/Apr/15	20/May/15	
🟢	CATCHME-155		Pridanie dalsich guestov (invite)	Michal Cihak	Pavol Michalek	↑	Closed	Fixed	14/Apr/15	20/May/15	
🔴	CATCHME-149		Pridavanie rovnakych casov	Matus Pikuliak	Vladimir Osvald	↑	Closed	Fixed	08/Apr/15	20/May/15	
🟢	CATCHME-134		Funkcnost nastaveni	Lukas Masar	Tomas Sykora	↑	Closed	Fixed	02/Apr/15	20/May/15	
🟢	CATCHME-118		Komentare k eventom	Unassigned	Matus Pikuliak	↑	Open	Unresolved	04/Mar/15	20/May/15	

Obr. D.12. Export evidencie úloh pre dvanásť šprint

10 Používateľské testovanie

Nižšie uvedený dokument je výsledkom hromadného testovania našej aplikácie.

10.1 Priebeh používateľského testovania

Za účelom testovania sme projekt odovzdali testovaciemu tímu o veľkosti 20 ľudí s vekom od 20 do 40 rokov. Väčšina z nich má dlhé skúsenosti s vývojom a testovaním moderných aplikácií, preto sme sa rozhodli zveriť im tento projekt. Testovanie prebiehalo hromadne v jednej miestnosti, avšak nedochádzalo k ovplyvňovaniu jednotlivých členov vzájomne. Predstavenie aplikácie a úvodné pokyny k používaniu boli prezentované celému tímu pred začatím testov.

Počas testov bol vždy k dispozícii člen tímu Cats Can Code, ktorý dohliadal na testy a objasňoval nezrovnalosti, ak taká situácia nastala. Členovia testovacieho tímu dostali formulár, do ktorého podľa priložených pokynov vpisovali svoje zistenia a výsledky testov. Na otestovanie mali k dispozícii sedem hlavných scenárov. Testovanie prebiehalo z časti na lokálnej kópii aplikácie a z časti na verejne dostupnej beta verzii aplikácie. Na konci testov prebehlo zhodnotenie. Spísali sa objavené nedostatky, návrhy na zmeny a hlavné prednosti aplikácie.

10.1.1 Prednosti aplikácie

Na základe vyhodnotenia používateľského testovania boli ako hlavné prednosti aplikácie definované nasledujúce vlastnosti:

- zaujímavý a nápaditý dizajn,
- možnosť registrácie a prihlásenia prostredníctvom existujúcich účtov,
- celková myšlienka projektu,
- použiteľnosť,
- bohaté používateľské nastavenia,
- možnosť šírenia vytvorenej udalosti viacerými cestami,
- veľmi dobre spracovaný kalendár,
- nezávislé hlasovanie za miesto a čas,

- zobrazenie zahlasovaných miest na mape.

10.1.2 Hlavné nedostatky aplikácie

Používateľské testovanie objavilo tieto hlavné nedostatky:

- rýchlosť odozvy aplikácie,
- problémy s prihlásením cez sociálne siete,
- nízka miera zabezpečenia,
- problémy so zle čitateľným textom pri zadávaní popisu udalosti,
- zoradovanie pridaných časov až po opätovnom načítaní stránky,
- pridanie miniatúr do kalendáru až po opätovnom načítaní stránky,
- neošetrené vstupy niektorých polí.

10.1.3 Návrhy na vylepšenie aplikácie

Testovanie prinieslo tieto návrhy na vylepšenie aplikácie:

- spraviť aplikáciu responzívnu s ohľadom pre využitie na mobilných zariadeniach,
- optimalizácia rýchlosti aplikácie,
- vítaná možnosť synchronizácie s kalendármi iných služieb,
- možnosť zvoliť farebnú schému pre každého používateľa osobitne,
- zvýšenie miery zabezpečenia.

10.1.4 Nedostatky zistené počas testovanie

Počas používateľského testovania sme si všimli nasledovné problémy, s ktorými sa následne budeme zaoberať:

- po ukončení testov používateľa dlho hľadali tlačidlo na odhlásenie,
- niektoré texty neboli dobre čitateľné kvôli zle zvolenej farebnej schéme,
- používatelia si spočiatku nevšimli možnosť podať spätnú väzbu priamo cez aplikáciu, nakoľko tlačidlo v hornej časti je nevýrazné a na väčších monitoroch sa strácalo aj druhé na pravej strane,

- pri pokuse pozvať ďalších ľudí na udalosť chvíľu trvalo nájsť príslušné tlačidlo.

10.2 Zhodnotenie testovania

Záverom tohto používateľského testovania je prevažne kladné hodnotenie nášho projektu. Keďže išlo o testovanie s asistenciou členov tímu, spätnú väzbu sme dostávali priamo formou vyplnenia formulárov k testom. Radi by sme sa týmto poďakovali nášmu testovaciemu tímu za ochotu podieľať sa na testovaní nášho projektu a za ich cenné zistenia behom používania našej aplikácie. Vďaka sledovaniu priebehu testovania sme došli k mnohým užitočným záverom, ktoré nám pomôžu aplikáciu vylepšiť.

10.2.1 Výsledky testovania

Tento dokument pôvodne obsahoval prílohu s pokynmi pre vyplnenie formuláru k používateľskému testu a príklady jednotlivých testov. Jedná sa o reprezentatívne vzorky formulárov, nakoľko obsahovali zhruba rovnaké pripomienky. Preto neuvádzame do príloh všetkých 20 používateľských testov. Celý dokument aj s prílohou je dostupný na adrese uvedenej v poznámke pod čiarou¹.

¹ http://team12-14.ucebne.fiit.stuba.sk/teampage12/doc/Vystup_z_pouzivatelskeho_testovania.pdf

11 Záverečná správa

Táto správa je finálnym odovzdaním informácie o produkte v rámci súťaže TP CUP.

11.1 Abstrakt

Žijeme v rušnej dobe, v ktorej je často veľmi náročné zorganizovať stretnutie pre viacerých ľudí vo vyhovujúcom čase a na vyhovujúcom mieste. Avšak, zároveň žijeme v dobe moderných technológií, ktoré by nám mali pomáhať žiť naše životy pohodlnejším spôsobom. Očakávania dnešných používateľov IT služieb a produktov sa neustále zvyšujú rovnako, ako napredujú aj samotné technológie.

Táto správa opisuje našu webovú aplikáciu Happen. Ide o novú sociálnu sieť založenú na udalostiach, čo ju odlišuje od iných sociálnych sietí. Jej hlavným cieľom je zdieľanie udalostí, ktoré sú zobrazované v osobnom kalendári každého používateľa Happen. Hlavnou výhodou našej aplikácie je možnosť organizovania udalosti a následne ju pridať do osobného kalendára veľmi jednoduchým a intuitívnym spôsobom.

Pozreli sme sa na problém manažovania času a organizovania udalosti z iného uhla pohľadu, ako existujúce riešenia. Sústreďujeme sa na životný cyklus udalosti, ktorý začína návrhmi časov a miest stretnutia a pokračuje dohadovaním. Kým sa skupina ľudí dohodne na mieste a čase stretnutia, môže to trvať aj niekoľko iterácií. My tento proces zjednodušujeme, robíme ho intuitívnym a zbavujeme ho bremena zbytočného vypisovania a orientovania sa v správach od viacerých ľudí.

Každá udalosť má svoj čas, miesto a účastníkov a je súčasťou používateľovho profilu. V každej udalosti je možné hlasovať za parametre udalosti ako sú návrhy miesta a času konania udalosti, čo dáva možnosť pre jednotlivých účastníkov udalosti vyjadriť svoje preferencie ohľadne konania udalosti.

System priebežne vyhodnocuje hlasovania za jednotlivé parametre udalosti, pričom používateľ ktorý udalosť vytvoril, môže udalosť kedykoľvek uzavrieť, a tým sa ukončí toto vyhodnocovanie. System následne ponúkne najlepšie možnosti na čas a miesto stretnutia na základe vyhodnotenia hlasovania jednotlivých používateľov. Zakladateľ udalosti si následne môže vybrať jeden z týchto návrhov, alebo si vyberie iný návrh na základe vlastných preferencií.

Po uzavretí udalosti, sú o tom ostatní používatelia notifikovaní mailom, a udalosť sa im uloží do ich osobného kalendára.

11.2 Prehľad riešenia

Súčasná doba nepochybne patrí moderným technológiám. Ich podstatou už nie je len uľahčenie a zvýšenie efektivity práce. poskytujú taktiež rozsiahle možnosti zábavy a aktivít pre voľný čas. Takýto druh zábavy má však často za následok, že sa z človeka, ktorý bol pôvodne tvorom spoločenským, stáva asociálna bytosť, tráviaca väčšinu svojho času sama. Aj toto bol jeden z dôvodov, prečo sme sa rozhodli vytvoriť Happen.

Happen je úplne novou sieťou, ktorá má všetky predpoklady na to, aby bola sociálnou. Sociálna sieť by totiž podľa nás mala v prvom rade ľudí spájať a to je práve hlavnou myšlienkou našej aplikácie. Máme za to, že súčasné sociálne siete, nie sú tak úplne sociálne. Naoko síce ľudí spájajú, avšak realita je veľmi často úplne iná. Používateľ má veľké množstvo priateľov, followerov... Otázkou však ostáva, do akej miery sú takýto ľudia ozajstnými priateľmi. O čom je teda ten náš Happen?

Ako už bolo napísané, Happen je novou sociálnou, ktorá je riadená udalosťami. Udalosťou pritom môže byť v podstate čokoľvek, čo sa používateľ rozhodne zdieľať. Z tohto dôvodu preto pochopiteľne tvorí základ celej aplikácie. Používateľ našej aplikácie ju dokáže behom niekoľkých sekúnd vytvoriť a pozvať svojich priateľov, aby sa jej zúčastnili. Unikátnosť Happen však nespočíva až tak v myšlienke dohodnutia stretnutia, ale v spôsobe jej spracovania. V tomto smere sme sa rozhodli ponúknuť používateľom jednoduchý dizajn, s funkcionalitou prispôbenou primárnemu účelu aplikácie.

Práve vďaka zameraniu sa na dohodnutie udalosti, resp. stretnutia, sme získali výhodu nad riešeniami, ktoré sa v súčasnej dobe používajú ako alternatíva pre dohodnutie stretnutia. Všetkým veľmi dobre známa sociálna sieť s modrým logom taktiež ponúka možnosť relatívne jednoduchého vytvorenia udalosti. Avšak ide len o akúsi doplnkovú službu, a jej funkcionalita je značne obmedzená. Vo svojej podstate sa potenciálni účastníci udalosti sa nedohadujú, ale iba vyjadrujú k navrhovanému času a miestu konania. To môže byť na základe ich komentárov, prípadne výsledkov ankety upravené zakladateľom udalosti, čo môže mať za následok rozpútanie ďalšej diskusie.

Samozrejme, že dohodnutie stretnutia nie je primárnym cieľom len našej aplikácie, a ani sa nesnažíme tváriť, že sme prišli s niečím, čo tu ešte nebolo. Tieto riešenia však trpia dvomi nedostatkami. Prvým z nich je dizajn. Ako sme sa sami presvedčili, reprezentácia dát nie je jednoduchou úlohou, avšak súčasné riešenia z pohľadu dizajnu nie vždy možno nazvať

súčasnými. Často pripomínajú aplikácie z pred desiatky rokov, ktoré boli zamerané v prvom rade na funkčnosť a otázke grafického spracovania a jednoduchosti používania nebola venovaná až tak veľká pozornosť. Niektoré z týchto aplikácií reálne z tohto obdobia a sú používané dodnes.

Druhým problémom riešení, zameraných na dohodnutie stretnutia, ktorý nás inšpiroval k vytvoreniu Happen, je opäť funkcionálnosť. V tomto smere sme sa bližšie pozreli aj na aplikáciu Doodle, ktorá patrí v množine súčasných riešení k najpoužívanejším. Funkčnosť Doodle rozhodne nespochybňujeme. Avšak myslíme si, že nie úplne pokrýva funkcionálnosť dohodnutia stretnutia. V jednom z najjednoduchších prípadov použitia používateľ aplikácie vyberie niekoľko termínov, kedy má čas na stretnutie s osobou, ktorej odošle pozvánku. V prípade že tejto osobe nevyhovuje žiadny termín, používateľ musí vytvoriť novú udalosť, určiť nové termíny a opätovne odoslať pozvánku. Tento postup sa pritom môže opakovať niekoľko krát. Samozrejme, že takýto prípad použitia je pre niektoré situácie postačujúci. Situácia sa však komplikuje pridaním ďalších účastníkov. Zároveň, uvedený príklad podľa nás nie úplne spĺňa výklad slova dohoda. Myslíme si, že dohoda by mala byť obojstranná. V Happen preto ponúkame možnosť vyjadrenia sa k termínom oboj stranám resp. všetkým zúčastneným. Jednotliví účastníci môžu pridávať nielen svoje návrhy času, ale aj miesta. Tieto sú automaticky zobrazené ostatným účastníkom, ktorí môžu jednoducho vyjadriť svoj súhlas s návrhom. Zároveň tak pokrývame prípad použitia, kedy zakladateľ udalosti nemá predstavu o čase alebo mieste.

Používateľ Happen má totiž možnosť kliknutím na tlačidlo, ktoré je dostupné z ktoréhokoľvek miesta v aplikácii, vytvoriť novú udalosť tak, že jediné čo musí zadať, je názov udalosti. všetko ostatné je na návrhoch a dohode všetkých zúčastnených resp. pozvaných.

Funkčnosť a jednoduchosť boli pre nás pri realizácii projektu dvomi základnými požiadavkami na výslednú aplikáciu. Snažili sme sa pritom do problematiky dohodnutia stretnutia vniesť vlastný pohľad, ktorý alternatívne riešenia problému podľa nás neponúkajú.

Samotná aplikácia je pritom vyvíjaná vo viacerých verziách. Všetka funkcionálnosť opisovaná vyššie, ako aj v časti Realizácia riešenia a Zhrnutie, je pritom súčasťou Free verzie, pričom pracujeme ešte na dvoch ďalších. Ich hlavnou funkcionálnosťou resp. jadrom je stále udalosť. Tá však naberá komerčnejší význam, a už nejde len o stretnutie za účelom spoločnej zábavy, ale aj zisku, pričom za rovnakým účelom boli vytvorené aj samotné verzie Happen. V Business verzií majú jej predplatitelia možnosť priblížiť sa k svojim zákazníkom a ich

potrebám. Možností je viac. Ide napríklad o cieleú reklamu svojich produktov a služieb pre používateľov Free verzie, pomocou ponuky, šitej na mieru. Pokiaľ ste vlastníkom baru, a chcete aby sa o Vašich službách dozvedela skupinka študentov, ktorá sa chce ísť zabaviť, prípadne ste ochotní ponúknuť im zľavu, táto verzia je určená aj pre Vás.

Táto verzia je taktiež správnym riešením, pre kaderníkov, zubárov, či iných podnikateľov, ktorí tak získajú prehľadný rezervačný systém. Ich zákazníkom je zároveň uľahčený výber vďaka zobrazeniu ich kalendára v prieniku s možnými termínmi. Pri výbere termínu je tento zároveň automaticky pridaný do ich osobného kalendára. Nemôže sa tak stať, že by si nevedomky naplánovali viaceré udalosti na jeden termín.

Vývoj tretej verzie Happen – Enterprise je spojený s negatívnou skúsenosťou pri výbere pracovnej doby. Firma na tento účel používala tabuľku, do ktorej si zamestnanec (v tomto prípade brigádnik) vložil svoje meno k času, v ktorom chcel pracovať. Happen v tomto prípade ponúka možnosť prídania termínov viacerými zamestnancami, pričom tieto návrhy vyhodnocuje, a určí kombináciu tak, aby bol pokrytý celý pracovný čas potrebným množstvom pracovníkov. Táto verzia teda slúži na manažment ľudských zdrojov a je šitá na mieru konkrétneho zákazníka.

11.3 Realizácia riešenia

Happen môže byť chápaný ako kalendár, do ktorého si používateľ pridá svoje aktivity a udalosti, avšak hlavnou myšlienkou našej sociálnej siete je pomôcť pri dohodnutí stretnutia s ostatnými. Táto myšlienka bola základom od prvého návrhu vyvíjanej aplikácie. Odvtedy však riešenie prešlo mnohými iteráciami, počas ktorých sa upravovala nielen funkcionálnosť, ale hlavne dizajn. Práve jeho jednoduchosť, s podporou intuitívneho ovládania bola jednou z hlavných požiadaviek na výslednú aplikáciu.

Primárnou vlastnosťou je vytvorenie udalosti, pričom Happen umožňuje vytvoriť stretnutie so zatiaľ neurčitým časom a miestom. Vytvorenie je tak veľmi rýchle. Zároveň je udalosť možné vytvoriť z ktoréhokoľvek miesta v aplikácii. Následne je účastníkom poskytnutý priestor, aby sa jednoducho dohodli.

Pred samotným dohadovaním je však potrebné, aby sa Vaši priatelia o udalosti dozvedeli. Happen ponúka viacero možností. V prvom rade ide o notifikácie v rámci aplikácie, kedy sa používateľovi udalosť zobrazí v zozname, spolu s ostatnými. Aplikácia taktiež umožňuje following-u a followerov. Takýmto sledovaním sa používatelia jednoducho a rýchlo dozvedia

o udalostiach svojich priateľov, známych, či záujmových skupín. Ďalšou možnosťou zdieľania sú sociálne siete, pričom stačí jednoduché kliknutie na príslušné tlačidlo a prípadný komentár k udalosti a Vaša udalosť je zobrazená vo Vašom profile. Samozrejmosťou je aj odoslanie e-mailu, alebo zdieľanie pomocou linky, ktorá je pre každú udalosť unikátna. Zároveň bola navrhnutá tak, aby bola jednoducho zapamätateľná.

Ďalším krokom je hlasovanie za konkrétne návrhy času a miesta. Aplikácia pritom ponúka možnosť nielen hlasovať za návrh, ale aj pridať vlastný. Tento návrh automaticky vidia ostatní pozvaní, čo znamená, že zaň môžu hlasovať.

Dohodnutie sa väčšieho množstva ľudí na jednom mieste a jednom čase je štatisticky veľmi nepravdepodobné, a preto musí existovať „vyššia moc“ ktorá napokon o týchto veciach rozhodne. V prvom rade je však potrebné, aby sa mala na základe čoho rozhodnúť. V Happen je touto mocou vlastník resp. zakladateľ udalosti. V prípade že chce udalosť uzavrieť, a ukončiť tak hlasovanie o jednotlivých možnostiach, urobí tak jednoduchým kliknutím na tlačidlo. Systém mu automaticky ponúkne zoznamy zoradené podľa preferencií väčšiny hlasujúcich, pričom za miesto a čas stretnutia zvolí práve tie, s najvyšším počtom hlasov. Samozrejme, že používateľ má možnosť si zo zoznamu vybrať inú možnosť. Následne udalosť uzavrie. Na základe tejto akcie sa zmení status stránky udalosti, čím sa samozrejme zmení aj jej obsah a dizajn. Pochopiteľne už nie je možné hlasovanie za jednotlivé návrhy. Tieto dokonca ani nie sú zobrazené. Na stránke udalosti je vidieť čas a miesto udalosti, ktoré je okrem iného zobrazené pomocou mapy. Taktiež sú zobrazení účastníci udalosti. Používatelia už nemajú možnosť hlasovať za návrh. Môžu však stále udalosť zdieľať, ako aj potvrdiť svoju účasť.

Jednou zo základných vlastností našej aplikácie je zobrazovanie udalostí pomocou kalendára, umiestneného na profilovej stránke každého používateľa. Udalosti sú v aktuálnej verzii v kalendári zobrazované v prípade, že k nej má používateľ nejaký vzťah. To znamená, že označil, že sa udalosti zúčastní, prípadne že je na ňu pozvaný a podobne.

Nikto nemá rád nové účty, nové registrácie, nové heslá... Happen preto podporuje možnosť prihlásenia sa pomocou účtu na sociálnej sieti Facebook.

Samozrejmosťou ako aj osobnej profilovej stránky, tak aj stránky udalosti sú nastavenia. Čo sa týka nastavení profilu, ide hlavne o vizuálne nastavenia, ako napríklad zobrazenie kalendára či profilovej fotky, s možnosťou jej zmeny. Pri udalosti je možné vybrať nastavenia už pri samotnom vytváraní. Tieto je však možné jednoducho zmeniť neskôr. Okrem vizuálnej

stránky a teda zobrazovania udalosti, sa tieto nastavenia týkajú možnosti pridávania návrhov, ako aj obmedzenia zdieľania udalosti pozvaným osobám.

Neoddeliteľnou súčasťou vývoja Happen je používateľské testovanie, ktoré prebehlo vo viacerých fázach, ako aj samotné požiadavky potenciálnych používateľov. Tie boli okrem iného zozbierané pomocou prieskumu, ktorý pre nás vypracovali kolegovia z Ekonomickej Univerzity v Bratislave. Na základe výsledkov dotazníka sme sa uistili v smerovaní produktu, ako aj upravili prioritu implementácie jednotlivých funkcionalít.

Nedávno taktiež prebehlo veľké testovanie na vzorke 20 ľudí, vo veku od 20 do 40. Títo naši testerí pritom majú pritom skúsenosti nie len s používaním moderných aplikácií, ale aj s ich vývoj a samotným testovaním. Preto sme sa rozhodli využiť ich skúsenosti z tejto oblasti a dali sme im za úlohu vykonanie niekoľkých úloh. Kladným hodnotením našej aplikácie je fakt, že všetci testerí dokázali zadané úlohy splniť. Získali sme však sekundárnu predstavu o tom, ako ďalej pokračovať a rozšíriť produkt. Správa z testovania je dostupná na stránke nášho tímu, v časti dokumenty (<http://team12-14.ucebne.fiit.stuba.sk>).

Výsledky testovania resp. návrhov na ďalšie smerovanie a rozšírenia pre nás v istom zmysle neboli prekvapivé. Boli veľmi podobné pripomienkam, ktoré sme získali vďaka prezentácií na študentskej vedeckej konferencii IIT.SRC.

Keďže názor používateľov aplikácie je pre nás veľmi dôležitý, pridali sme do aplikácie komponent Feedback. Sme si totiž vedomí toho, že aj pokiaľ by sa nám podarilo odladiť všetky nedostatky, stále bude existovať možnosť rozšírenia. Na základe takto zozbieraných názorov tak budeme schopní dynamicky reagovať na požiadavky samotných používateľov a pracovať na nových funkcionalitách tak, aby boli zohľadnené preferencie reálnych používateľov.

11.4 Zhrnutie

V tejto správe bola popísaná novo vytvorená udalosťami riadená sociálna sieť Happen, spolu s jej hlavnou funkcionalitou. Uvedené vlastnosti sú súčasťou aktuálnej verzie, dostupnej na <http://team12-14.ucebne.fiit.stuba.sk/happen>. Súčasne však pracujeme na analýze a implementácií ďalšej funkcionality, ktorá zohľadňuje výsledky testovania, ako aj pripomienky zo študentskej vedeckej konferencie.

Prvou z nich je responzivnosť. V súčasnej dobe existujú viaceré framework-y, podporujúce zobrazovanie na rôznych zariadeniach. V našom prípade sme však z viacerých dôvodov žiaden z nich nepoužili. Ako bolo uvedené vyššie, na dizajn sme kládli veľké požiadavky, a vedeli sme, že bude niekoľko krát zmenený. Preto sme sa rozhodli responzivnosť našej aplikácie riešiť až v druhej fáze.

V ďalšej fáze vývoja taktiež plánujeme integráciu s kalendárom od spoločnosti Google. Ten je totiž pravdepodobne najviac rozšírený, pričom pri mnohé iné kalendáre majú taktiež umožnenú integráciu práve s Google Kalendárom. Vďaka tomu budeme schopný nepriamo získať údaje aj z ostatných kalendárov používateľa. Samotnú integráciu chceme pritom implementovať obojsmerne. Happen tak bude schopný po udelení povolenia zobrazovať udalosti z Google Kalendára, ktorý bude zároveň poznať a zobrazovať udalosti priradené do kalendára pomocou aplikácie Happen. V tomto smere chceme taktiež pridať možnosť prihlásenia sa do našej aplikácie pomocou Google účtu.

V súčasnej dobe sme si taktiež vedomí nedostatkov z pohľadu zabezpečenia aplikácie. Opätovne ide o vlastnosť, s ktorou sme od začiatku rátali, avšak v rámci prototypovania nebola pre nás až tak podstatnou, nakoľko sme sa potrebovali zamerať na primárne vlastnosti vznikajúcej aplikácie.

Taktiež pracujeme na rozširujúcich nastaveniach pre udalosti, aby tak bol Happen schopný pokryť čo možno najväčšie množstvo prípadov použitia.

Aj napriek spomenutým chýbajúcim vlastnostiam, ktoré si z dôvodu prebiehajúceho vývoja nedovolíme nazvať nedostatkami, je možné už teraz Happen využívať a užívať si tak spoločné chvíle s Vašimi priateľmi. Vďaka Happen môžete vidieť, čo sa deje okolo Vás, Vašich priateľov, alebo Vašich záujmov a jednoducho sa k jednotlivým udalostiam pripojiť. Toto všetko Happen ponúka v modernom dizajne, podporujúcom intuitívne používanie.