

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Webový komunitný systém otázok a odpovedí

Dokumentácia k inžinierskemu dielu

Vedúci tímu: Ing. Ivan Srba

Členovia tímu: Bc. Rastislav Dobšovič, Bc. Marek Grznár, Bc. Jozef Harinek,
Bc. Samuel Molnár, Bc. Peter Páleník, Bc. Dušan Poizl, Bc. Pavol Zbell

Akademický rok: 2013/2014

Obsah

1 Úvod.....	1 - 1
2 Ciele na zimný semester.....	2 - 1
3 Šprint 1 – Drone.....	3 - 1
3.1 Autorizácia.....	3 - 1
3.2 Lokalizácia.....	3 - 1
3.3 Prihlásenie.....	3 - 2
3.4 Profil používateľa.....	3 - 3
3.5 Zaznamenávanie udalostí.....	3 - 6
4 Šprint 2 – Roach.....	4 - 1
4.1 Vloženie novej otázky.....	4 - 1
4.2 Zobrazenie otázky.....	4 - 1
4.3 Zobrazenie nových otázok.....	4 - 2
5 Šprint 3 – Hydralisk.....	5 - 1
5.1 Vloženie novej odpovede.....	5 - 1
5.2 Zobrazenie odpovedí pri otázke.....	5 - 2
6 Šprint 4 – Infestor.....	6 - 1
6.1 Hlasovať za otázku alebo odpoveď.....	6 - 1
6.2 Označiť otázku ako obľúbenú.....	6 - 2
6.3 Označiť odpoveď ako najlepšiu, pomocnú a overenú.....	6 - 4
6.4 Filtrovanie otázok podľa značiek.....	6 - 5
7 Šprint 5 – Ultralisk.....	7 - 1
7.1 Vloženie komentáru pre otázku a odpoveď.....	7 - 1
7.2 Zobrazenie počtu hlasov, odpovedí a zobrazení pri otázke.....	7 - 2
7.3 Zobrazenie zodpovedaných otázok.....	7 - 2
7.4 Kategória ako pomenovaná množina značiek.....	7 - 4
8 Celkový pohľad po prvý kontrolný bod.....	8 - 1
8.1 Architektúra.....	8 - 2
8.2 Dátový model.....	8 - 3
9 Celkový pohľad po druhý kontrolný bod.....	9 - 1
9.1 Architektúra.....	9 - 2
9.2 Dátový model.....	9 - 3

10 Modul aktivita a nasledovanie.....	10 - 1
10.1 Analýza.....	10 - 1
10.2 Návrh.....	10 - 1
10.3 Implementácia.....	10 - 1
10.4 Testovanie.....	10 - 2
11 Modul administrácia, sli.do, označovanie správnych odpovedí, štatistiky.....	11 - 1
11.1 Analýza.....	11 - 1
11.2 Návrh.....	11 - 2
11.3 Implementácia.....	11 - 2
11.4 Testovanie.....	11 - 3
12 Modul úprava, mazanie a anonymné otázky	12 - 1
12.1 Analýza.....	12 - 1
12.2 Návrh.....	12 - 1
12.3 Implementácia.....	12 - 2
12.4 Testovanie.....	12 - 3
13 Modul filtrovanie a vyhľadavanie.....	13 - 1
13.1 Analýza.....	13 - 1
13.2 Návrh.....	13 - 2
13.3 Implementácia.....	13 - 2
13.4 Testovanie.....	13 - 5
14 Modul zoznamy, zmeny v systéme, úvodná strana.....	14 - 1
14.1 Analýza.....	14 - 1
14.2 Návrh.....	14 - 2
14.3 Implementácia.....	14 - 2
14.4 Testovanie.....	14 - 4
15 Modul Markdown, pomoc v systéme.....	15 - 1
15.1 Analýza.....	15 - 1
15.2 Návrh.....	15 - 1
15.3 Implementácia.....	15 - 1
15.4 Testovanie.....	15 - 2
16 Modul notifikácií a sledovania.....	16 - 1
16.1 Analýza.....	16 - 1

16.2 Návrh.....	16 - 2
16.3 Implementácia.....	16 - 2
16.4 Testovanie.....	16 - 4
17 Celkový pohľad.....	17 - 1
17.1 Architektúra.....	17 - 4
17.2 Dátový model.....	17 - 5
18 Čo sme nestihli?.....	18 - 1
19 Čo sme sa naučili?.....	19 - 1
A Používateľská príručka.....	A - 1
1 Pridanie novej otázky.....	A - 1
2 Úprava.....	A - 3
3 Hľadanie informácií.....	A - 4
4 Odpovedanie na otázku.....	A - 6
5 Úprava profilu.....	A - 7
6 Markdown.....	A - 9
B Inštalačná príručka.....	B - 1
C Štatistiky nad repozitármi projektu.....	C - 1

1 Úvod

Naším cieľom je vytvoriť webový systém, ktorý bude umožňovať interakciu medzi študentmi navzájom, alebo medzi študentmi a vyučujúcimi pomocou otázok a odpovedí. Študenti tak budú môcť riešiť svoje problémy so zadaniami alebo nejasnosťami z učiva položením otázky v našom systéme. Študent, ktorý otázku položil, určí správnu odpoveď a tak ostatní študenti s rovnakým problémom môžu rýchlo nájsť správne riešenie. Pre prehľadnosť budú mať všetky otázky rôzne *informačné značky*.

Naším cieľom bude v prvom semestri vytvoriť funkčnú webovú službu, ktorá bude umožňovať kladenie otázok, odpovedanie. Ďalej bude k dispozícii hlasovanie pre jednotlivé odpovede alebo komentovanie odpovedí. Nakoniec budeme pomocou informačných značiek umožňovať vyhľadávanie v otázkach.

V druhom semestri sa do systému budú pridávať ďalšie vylepšenia. Bude sa pracovať na prispôsobovaní úvodnej obrazovky pre potreby jednotlivých používateľov.

2 Ciele na zimný semester

1. šprint – Drone:

- zaznamenávanie udalostí,
- lokalizácia,
- prihlásenie,
- profil používateľa.

2. šprint – Roach:

- vloženie novej otázky,
- zobrazenie otázky,
- zobrazenie zoznamu nových otázok.

3. šprint – Hydralisk:

- vloženie novej odpovede
- zobrazenie zoznamu zodpovedaných otázok
- výber najlepšej odpovede

4. šprint – Infestor:

- hlasovanie za odpoveď,
- vyhľadávanie v otázkach,
- vloženie komentáru k odpovedi,
- preferencie a práva.

5. šprint – Ultralisk:

- refaktORIZÁCIA.

3 Šprint 1 – Drone

3.1 Autorizácia

3.1.1 Úloha

Navrhnuť rozhranie pre autorizáciu používateľov v systéme na základe ich roli a právomoci. K rozhraniu je potrebné pridať aj príklad použitia.

3.1.2 Návrh

Pre implementáciu rolí a právomoci používateľov sme použili knižnicu *CanCan*¹. Knižnica implementuje jednoduché rozhranie, pomocou ktorého je možné definovať základné pravidlá pre autorizáciu používateľov. Rozhranie spracováva právomoci primárne na strane jazyka *Ruby*. Na základe toho je architektúra rozšíriteľná aj o právomoci perzistované v rámci databázy. Všetky právomoci používateľa sú definované v súbore *ability.rb*, ktorý sa nachádza v adresári *app/models*. Všetky právomoci sú definované pomocou kľúčového slova *can* alebo *cannot*, ktoré je možné použiť v pohľadoch a pri spracovávaní požiadaviek od aktuálneho používateľa na určenie jeho právomocí.

3.1.3 Implementácia a testovanie

Ako príklad pre autorizáciu používateľa sme zvolili možnosť zmeny mena. Používateľ prihlásený údajmi zo systému *AIS*² nemá právomoc na zmenu svojho mena a priezviska. Polia s menom a priezviskom sú pre tohto používateľa v sekcii úpravy profilu vypnuté. Na strane servera sa kontrolujú právomoci používateľa editovať meno a priezvisko a na základe kontroly právomoci sa určujú parametre, ktoré sú povolené v odoslanej požiadavke (angl. request). Právomoc používateľa sme definovali s názvom *change_name*, pričom túto akciu daný používateľ môže vykonať nad modelom *User*. Dané povolenie sme otestovali v jednotkových testoch modelu *User*. Na testovanie sme použili pomocnú metódu *be_able_to* knižnice *CanCan* pre testovací rámec *Rspec*³.

3.2 Lokalizácia

3.2.1 Úloha

Návrh a implementácia lokalizácie aplikácie. Návrh má brať do úvahy rôzne spôsoby a prístupy k lokalizácii v Rails aplikáciách. Cieľom je nájsť a definovať najvhodnejší spôsob a prístup k lokalizácii aplikácie pre potreby tohto projektu.

1 CanCan: <https://github.com/ryanb/cancan>

2 AIS: <http://is.stuba.sk/>

3 Rspec: <https://github.com/rspec/rspec>

3.2.2 Návrh

Pri návrhu sme dôkladne zvážili možnosti implementácie lokalizácie Rails aplikácie. Nerozhodli sme sa pre použitie lokalizovaných pohľadov, t.j. rozdelenia pohľadov napr. pri `app/views/users` do podadresárov `en` a `sk`, kvôli zneprehľadneniu a duplikovania zdrojového kódu samotných pohľadov. Lokalizáciu preto navrhujeme riešiť na úrovni konfiguračných súborov v adresári `config/locales`. V tomto adresári plánujeme vytvoriť prehľadnú štruktúru podadresárov ako napr. `errors`, `helpers`, `models` a `views`. V každom z týchto adresárov budú ďalej podadresáre pre lokalizáciu knižníc tretích strán a podadresár `views` sa bude organizovať podobne ako `app/views`. Podadresáre budú obsahovať konfiguračné súbory `en.yml` a `sk.yml`. Pre potreby lokalizácie sa v zdrojovom kóde budú referencovať preklady pomocou kľúčov, ktoré by mali reflektovať štruktúru podadresárov (najmä v prípade `views`) pomocou metódy `translate(key)`, skrátene `t(key)`.

3.2.3 Implementácia a testovanie

Referenčná implementácia a testovanie je zahrnuté v rámci pohľadu a editovania profilu používateľa. Vid' pohľady v adresároch `app/views/{devise,user}` a konfiguračné súbory v adresároch `config/locales/views/{devise,users}`.

3.3 Prihlásenie

3.3.1 Úloha

Návrh a implementácia prihlasovania a registrácie používateľa. Návrh má brať do úvahy možnosť prihlásenia údajmi z Akademického informačného systému (ďalej len AIS) bez potreby manuálnej registrácie používateľa.

3.3.2 Návrh

Pri návrhu sme využili knižnicu *Devise*⁴, ktorá obsahuje väčšinu funkcionality pre registráciu a prihlasovanie. Knižnica je rozdelená do viacerých modulov, ktoré obsahujú funkcionality pre registráciu, prihlásenie, obnovu hesla, potvrdenie účtu emailom a zamknutie účtu za určitých podmienok. Používateľ je reprezentovaný modelom `User`. Model využíva všetky moduly knižnice *Devise*. Základnými atribútmi používateľa sú `login`, `email` a `heslo`, pričom používateľ sa po registrácii prihlasuje pomocou atribútov `login` a `heslo`.

Pre potrebu prihlásenia používateľa údajmi z AIS sme rozšírili modul prihlasovania knižnice *Devise*. Nahradili sme metódu `create` v triede `SessionsController` vlastnou implementáciou prihlasovania. Overovanie údajov používateľa pre AIS je realizované servisným objektom `Users::Authentication`, ktorý ako atribúty uvažuje vzdialenú službu pre autentifikáciu používateľa a prihlasovacie parametre, ktoré zadal používateľ. Vzdialená služba LDAP

4 Devise: <https://github.com/plataformatec/devise>

Akademického informačného systému je reprezentovaná triedou `Stuba::AIS`, ktorá na základe mena a hesla autentifikuje používateľa. Výsledkom autentifikácie je inštancia objektu `Stuba::User`, ktorá obsahuje atribúty profilu používateľa v akademickom systéme. Pri prvom prihlásení vytvoríme nového používateľa z údajov z AIS profilu. Pre potrebu rozlíšenia AIS používateľa a registrovaného používateľa sme do modelu `User` pridali atribúty

- `ais_login` – login používateľa v systéme AIS,
- `ais_uuid` – identifikačné číslo používateľa v systéme AIS,

pričom oba atribúty sa nastavujú len v prípade prihlásenia údajmi z AIS.

3.3.3 Implementácia a testovanie

Model používateľa je otestovaný len na úrovni validácií a metód, o ktoré sa model rozšíril, keďže väčšina funkcionality pre prihlasovanie je otestovaná v testoch knižnice *Devise*. Autentifikácia pomocou údajov z AIS je izolovaná v rámci vlastnej knižnice. Komponenty tejto knižnice sú otestované v integrácii s knižnicou pre tvorbu LDAP dopytov. Servisný objekt `Users::Authentication`, ktorý implementuje autentifikáciu používateľa pomocou vzdialenej služby, je otestovaný vzorovými údajmi s využitím simulovaného správania vzdialenej služby. Jednotlivé pohľady knižnice *Devise* pre prihlásenie a registráciu sme upravili pre integráciu s rámcom *Bootstrap*⁵. Funkcionalita daných pohľadov je pokrytá akceptačnými testami, pričom akceptačné testy pre prihlasovanie údajmi z AIS opäť simulujú správanie formou metódy `authenticate` triedy `Stuba::AIS` so vzorovými údajmi.

3.4 Profil používateľa

3.4.1 Úloha

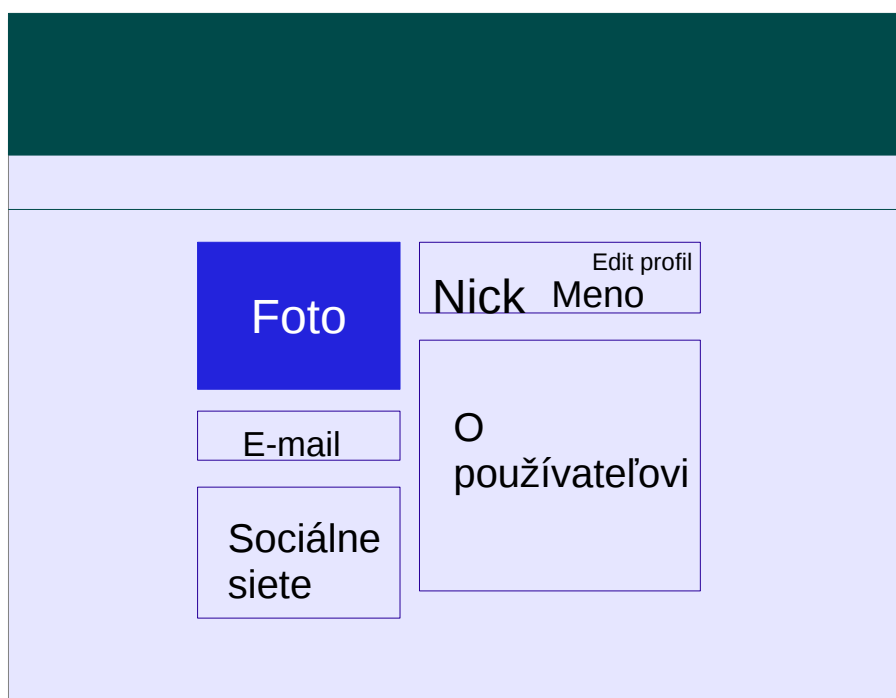
a) Navrhnuť a implementovať rozhranie pre zobrazenie profilu používateľa. Samotný návrh musí obsahovať autorizáciu používateľa. Ak sa používateľ rozhodne nezverejniť svoje meno a email iným používateľom, tak tieto údaje sú viditeľné iba pre neho. Pred inými používateľmi sú tieto informácie skryté.

b) Návrh a implementácia formulárov pre zmenu profilu a účtu používateľa. Návrh má brať do úvahy autorizáciu používateľa, podľa práv v systéme (používateľ prihlásený pomocou AIS konta nemôže editovať svoje meno). Návrh má takisto brať do úvahy rozdelenie menu podľa špecifikácie na logické celky.

3.4.2 Návrh

a) Ako je zobrazené na Obr. 3-1 rozhranie obsahuje tieto základné prvky: foto, email, sociálne siete, o používateľovi, prezývku, meno, presmerovanie na editovanie profilu.

⁵ Bootstrap: <http://getbootstrap.com/>



Obr. 3-1: Návrh prostredia pre zobrazenie profilu

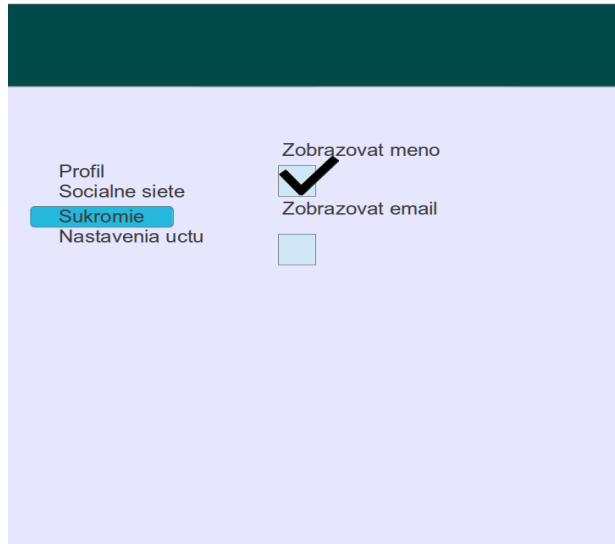
b) Pri návrhu sme podobne ako pre prihlasovanie použili knižnicu *Devise*⁶, ktorá poskytuje funkcionality pre autentifikáciu používateľa, v tomto prípade sme ju použili pre validáciu údajov vo formulári pre editáciu nastavení účtu (4. časť v grafickom návrhu). Na autorizáciu je použitá knižnica *CanCan*⁷, pomocou ktorej je spravované nastavenie práv v systéme. Podľa toho či má na to používateľ práva, má možnosť zmeniť si meno (AIS používateľ si meno zmeniť nesmie).

Ako je znázornené na Obr. 3-3, editácia profilu je rozdelená do štyroch častí:

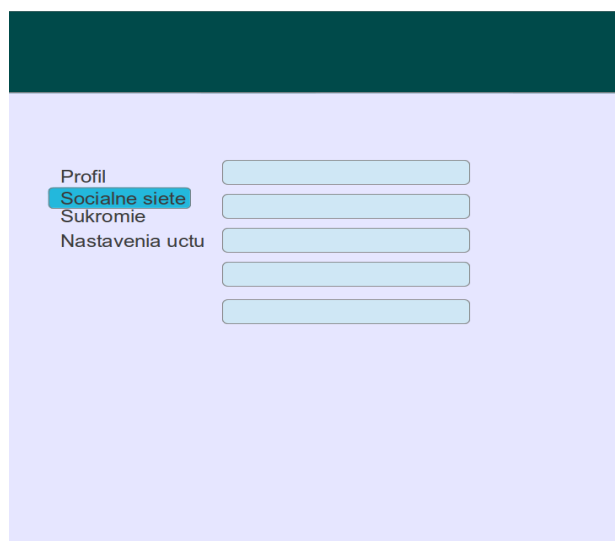
- profil,
- sociálne siete,
- súkromie (náčrt je vyobrazený na Obr. 3-2.),
- nastavenia účtu,
- v každej z častí je príslušný formulár, v ktorom je možnosť meniť údaje.

⁶ Devise: <https://github.com/plataformatec/devise>

⁷ CanCan: <https://github.com/ryanb/cancan>



Obr. 3-2: Návrh používateľského rozhrania pre nastavenia súkromia



Obr. 3-3: Návrh používateľského rozhrania pre úpravu profilu

3.4.3 Implementácia a testovanie

a) Na zobrazenie *Gravatar*⁸ fotografie sme implementovali helper, ktorý pomocou `gravatar_email` vracia používateľovu fotku z *Gravatar*. Keďže mail na *Gravatar* nemusí byť vyplnený zobrazí sa základný obrázok. Na implementáciu dizajnu grafického rozhrania pre zobrazenie profilu sme využili rámec *Bootstrap*⁹.

Rozhranie pre zobrazenie profilu sme otestovali akceptačnými testami. Na spravovanie práv pre zobrazenie voliteľných prvkov je použitá knižnica *CanCan*.

b) Formuláre pre úpravu profilu používateľa sme otestovali akceptačnými testami. V modeli používateľa boli otestované doplnené metódy a validácie. Do modelu bola pridaná metóda vracajúca mail na *Gravatar* – `gravatar_email` (Globally recognized avatar, služba spájajúca mailovú adresu používateľa s jeho globálnym profilom), keďže mail pre službu *Gravatar* nemusí byť rozdielny od mailu účtu – vtedy je toto pole v databáze prázdne a má sa zobraziť mail účtu. Formuláre sú naštýlované pomocou rámca *Bootstrap*.

Pohľad pre editáciu profilu používateľa je rozdelený na štyri časti, ktoré tvoria menu. V každej časti je samostatný formulár. Okrem formulára, ktorý spracúva knižnica *Devise* (zmena nastavení účtu) sú formuláre spracované kontrolórom pre používateľa v metóde `update_profile`.

3.5 Zaznamenávanie udalostí

3.5.1 Úloha

Návrh a implementácia zaznamenávania udalostí, ktoré nastanú v systéme. Návrh má brať do úvahy možnosť automatického aj manuálneho zaznamenávania udalostí vzhľadom na interakciu používateľa so systémom.

3.5.2 Návrh

Pre potreby projektu sme navrhli jednoduché automatické aj manuálne zaznamenávanie udalostí, ktoré vznikajú pri interakcii používateľa so systémom priamo v *Controller* triedach Rails Aplikácie. Pri manuálnom zaznamenávaní udalostí je cieľom zaznamenať udalosť (dáta) v čo najmenšom počte riadkov zdrojového kódu a prehľadnou formou (kvôli zachovaniu prehľadnosti samotnej logiky v moduloch *Controller*).

3.5.3 Implementácia a testovanie

Pri implementácii sme použili vzor *Service Object*, ktorý výrazne uľahčuje zaznamenávanie udalostí aj mimo modulov *Controller* a okrem toho umožňuje aj lepšie testovanie samotnej funkcionality zaznamenávania udalostí. Na databázovej vrstve sú zaznamenané udalosti uchovávané v tabuľke

8 *Gravatar*: <http://www.gravatar.com/>

9 *Bootstrap*: <http://getbootstrap.com/>

Events, ktorá obsahuje čas uloženia udalosti a dáta udalosti vo formáte JSON. Udalosť môže obsahovať (takmer) ľubovoľné JSON dáta, pričom tie sú tesne pred uložením vždy automaticky obohatené o rôzne podporné dáta ako napr. identifikátor sedenia, URL, identifikátor akcie triedy Controller a jej parametre (aj z formulárov – heslá a iné citlivé informácie sú bezpečne automaticky odstránené a do databázy sa neukladajú), dáta o používateľovi ako login alebo e-mail. Pri automatickom ukladaní udalostí sa mechanizmus ukladania spúšťa v rámci `before_action` každej akcie ľubovoľnej triedy, ktorá priamo dedí od triedy `ApplicationController`. Na manuálne ukládanie udalostí priamo v triede `Controller` (napríklad pre potrebu uloženia rôznych dát udalosti podľa vetvenia implementovanej akcie) slúži metóda `log`, ktorej jediný parameter sú JSON dáta udalosti (dátový typ `Hash` v Ruby), ktoré musia mať na v koreni kľúč `action`, ktorého hodnota stručne popisuje udalosť. Okrem kľúča `action` je možné v JSON dátach špecifikovať aj iné (takmer) ľubovoľné kľúče, ktoré budú uložené pre danú udalosť.

4 Šprint 2 – Roach

4.1 Vloženie novej otázky

4.1.1 Úloha

Navrhnuť a implementovať pridávanie nových otázok do systému. Návrh má brať do úvahy asociovanie kategórií a značiek (angl. tag) k vytváranej otázke.

4.1.2 Návrh

Navrhli sme jednoduchý model otázok – `Question`, ktorého základnými atribútmi sú názov a text otázky. K tejto entite sme pridali asociáciu na model kategórie – `Category`, ktorý dopĺňa sémantiku otázky a jej zaradenie. Pre rozšírenie sémantiky otázky sme si zvolili značky, ktoré budú používatelia môcť pridávať k otázkam. Na implementáciu značiek sme si zvolili knižnicu *acts-as-taggable-on*¹⁰. Knižnica implementuje dva modely – `Tag` a `Tagging`. Model `Tagging` slúži ako prepojovacia tabuľka medzi inými entitami a modelom `Tag`. Knižnica je rozširiteľná pre všetky scenáre pridávania značiek alebo iných označení pre entity, pretože využíva polymorfické asociácie.

Model značiek sme rozšírili o normalizáciu, pričom všetky značky sú normalizované na malé písmená a všetky medzery sú nahradené pomlčkou. Pri implementácii rozhrania sme si zvolili knižnicu *select2*¹¹, ktorý unifikuje vzhľad pre pole s možnosťou výberu (angl. select box). Nad knižnicou sme vytvorili jednoduché API pre definovanie vlastností pre polia so značkami. Vzhľad elementov *select2* sme prispôbili rámcu *Bootstrap*. Pre lokalizáciu textov v JavaScript kóde sme použili knižnicu *i18n-js*¹².

4.1.3 Implementácia a testovanie

Pridanie novej otázky sme otestovali na úrovni akceptačných testov. Pre potreby otestovania funkcionality knižnice *select2* sme vytvorili pomocné metódy pre vloženie vstupu do polí *select2*. Vytvorili sme *FactoryGirl*¹³ definície pre model `Category`, `Question` a `Tag`.

4.2 Zobrazenie otázky

4.2.1 Úloha

Zobraziť vybranú otázku. Okrem samotnej otázky a textu otázky je treba zobraziť aj základné informácie o používateľovi ktorý otázku položil.

10 *acts-as-taggable-on*: <https://github.com/mbleigh/acts-as-taggable-on>

11 *select2*: <http://ivaynberg.github.io/select2/>

12 *i18n-js*: <https://github.com/fnando/i18n-js>

13 *FactoryGirl*: https://github.com/thoughtbot/factory_girl

4.2.2 Návrh

Zobrazenie otázky zabezpečí samostatný pohľad. Otázka bude zobrazená v troch stĺpcoch. V prvom stĺpci bude zobrazené informácie o autorovi ako sú meno, jeho fotka. Meno. V strednom stĺpci sa bude zobrazovať nadpis otázky, dátum polozenia otázky, značky a samotný text otázky. Pravý stĺpec bude obsahovať hlasovanie k otázke a či je otázka zodpovedaná.

4.2.3 Implementácia a testovanie

Na zobrazenie otázky slúži štandardná metóda `show` v triede `QuestionsController`. V tejto metóde sa nájde otázka s `id` predaným ako parameter. Nájdenie otázky zabezpečuje trieda `Question` implementujúca dátový model. Triedy implementujúce dátový model dedia od `Base::ActiveRecord`. Pohľad je implementovaný pomocou *Bootstrap*.

Prezývka a fotka slúžia ako odkaz na profil používateľa. Dátum a čas polozenia otázky zobrazuje špeciálna knižnica ktorá automaticky zobrazuje čas vo forme času ktorý uplynul od polozenia otázky. Napríklad pred 15 minútami, dvoma dňami a podobne. Od určitej doby ale zobrazuje už len čistý dátum. Informácie o používateľov sú vyčlenené do samostatného *partial view* v súbore `views/users/_square.html.erb`. Hlasovanie k otázke je vyčlenené taktiež do samostatného súboru `views/questions/_voting.html.erb`.

4.3 Zobrazenie nových otázok

4.3.1 Úloha

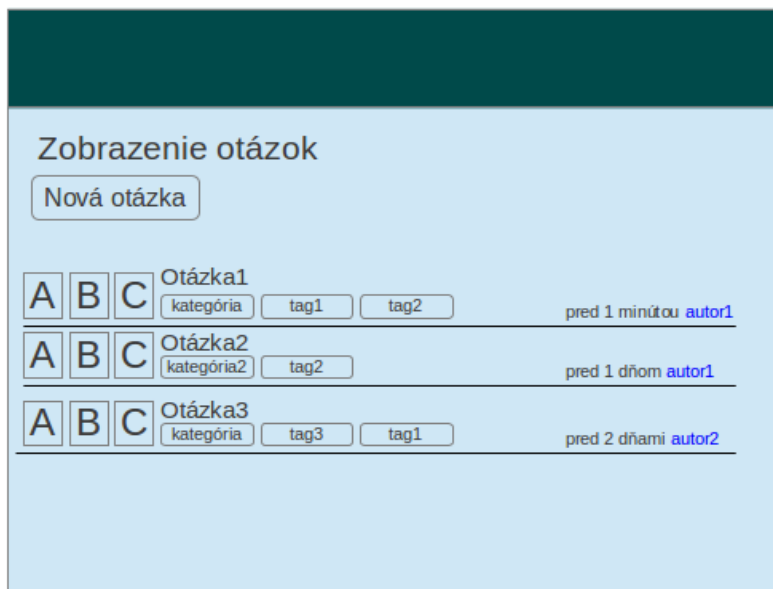
Návrh a implementácia zobrazenia nových otázok. Zobrazovať sa budú len nové otázky, ktoré budú zoradené podľa ich dátumu vzniku. Spolu s otázkami sa budú zobrazovať aj značky, kategórie, meno používateľa, ktorý otázku položil a doba, kedy bola otázka položená.

4.3.2 Návrh

V rozhraní pre zobrazenie nových otázok sú podľa Obr. 4-1 uvedené prvky:

- tlačidlo na pridanie novej otázky,
- názov otázky s kategóriami a značkami,
- doba kedy bola otázka pridaná,
- autor otázky,
- A – počet hlasov,
- B – počet odpovedí,

- C – počet videní.



Obr. 4-1: Zobrazenie zoznamu nových otázok

Do návrhu sa pridalo stránkovanie pomocou knižnice *kaminari*¹⁴, ktorá sprehládňuje prehľadávanie aj vo väčšom množstve otázok.

4.3.3 Implementácia a testovanie

Na implementáciu dizajnu grafického rozhrania pre zobrazenie profilu sme využili rámec *Bootstrap*¹⁵ doplnené o *custom CSS*. V zložke `views/kaminari` za pomoci `_paginator` a ďalších sme vytvorili stránkovanie, pričom sme v `questions_controller` pomocou `Question.order(:created_at).page(params[:page]).per(10)` zadefinovali, aby sa radili otázky podľa dátumu pridania a nastavili sme desať otázok na stránku. Pohľad pre zobrazenie otázok vypisuje otázky, kategórie spolu so značkami, ktoré sú farebne odlišené. Informácie o otázke ako jej názov, autor, dátum vytvorenia a jej štatistiky sa spracovávajú z databázy.

Pre overenie funkcionality sme vytvorili akceptačný test. Testujeme korektnosť vypisovaných údajov na stránke.

14 Kaminari: <https://github.com/amatsuda/kaminari>

15 Bootstrap: <http://getbootstrap.com/>

5 Šprint 3 – Hydralisk

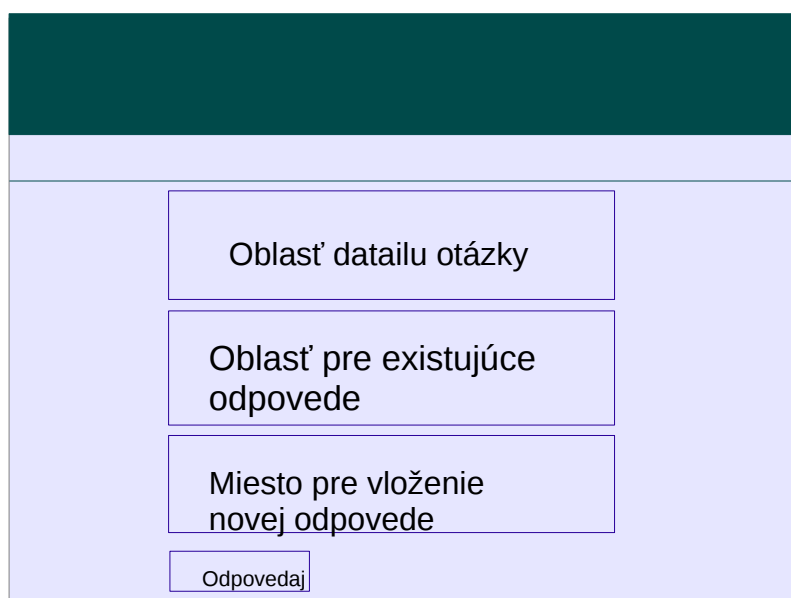
5.1 Vloženie novej odpovede

5.1.1 Úloha

Navrhnúť a implementovať rozhranie pre vloženie novej odpovede. Používateľ bude môcť odpovedať na otázku, ktorej detail si zobrazí. Návrh má umožniť odpovedi priradiť rôzne „stavy“.

5.1.2 Návrh

Navrhli sme model odpovedí. Hlavnými atribútmi sú text odpovede a identifikátor otázky, ku ktorej sa viaže a identifikátor používateľa, ktorý odpoveď vytvoril. Na Obr. 5-1 je zobrazené rozhranie pre vloženie novej odpovede. Oblasť detailu otázky a oblasť pre existujúce odpovede neboli našou úlohou.



Obr. 5-1: Návrh grafického rozhrania pre vloženie novej odpovede

5.1.3 Implementácia a testovanie

Grafický návrh rozhrania pre vloženie novej otázky sme sa rozhodli implementovať ako *partial view* `_answer_form.html.erb`, ktorý sa pomocou príkazu `<%= render 'answer_form' %>` zavolá vo view pre zobrazenie otázky (kapitola 3.2. Zobrazenie otázky). Implementácia pomocou *partial view* je hlavne z dôvodu prehľadnosti zdrojového kódu. Na implementovanie „stavov“ sme použili gem *acts-as-tagable-on*¹⁶, ktorý nám umožní pohodlné značkovanie (udelenie „stavov“).

Vytvorenie otázky sme otestovali akceptačnými testami.

5.2 Zobrazenie odpovedí pri otázke

5.2.1 Úloha

Zobraziť odpovede na predtým zodpovedanú otázku

5.2.2 Návrh

Otázky sa budú zobrazovať pod otázkou. Doplní sa teda pohľad zobrazenia otázky. Každá otázka bude mať viacero odpovedí. Odpoveď sa bude zobrazovať podobne ako otázka. Odpoveď bude mať text odpovede, čas, autora. Podobne ako otázka je aj odpoveď rozdelená do troch stĺpcov. Prvý bude obsahovať informácie o používateľovi, druhý dátum vyplnenia odpovede, a tretí hlasovanie k odpovedi.

5.2.3 Implementácia a testovanie

Zobrazenie odpovedí pod otázkou je implementované v samostatnom *partial view* `_answers.html.erb`. V tomto pohľade je zobrazený počet odpovedí. Zobrazenie jednotlivých odpovedí je implementované v `/views/questions/_answer.html.erb`. Hlasovanie k odpovedi je v samostatnom *partial view* `views/answers/_voting.erb.html`.

16 Acts-as-tagable-on: <https://github.com/mbleigh/acts-as-tagable-on>

6 Šprint 4 – Infestor

Štvrtý šprint mal trvanie dva týždne a jeho cieľom bolo vytvoriť roly pre používateľov a rôzne vylepšenia pre odpovede ako napr. prídanie komentárov alebo označovanie odpovedí.

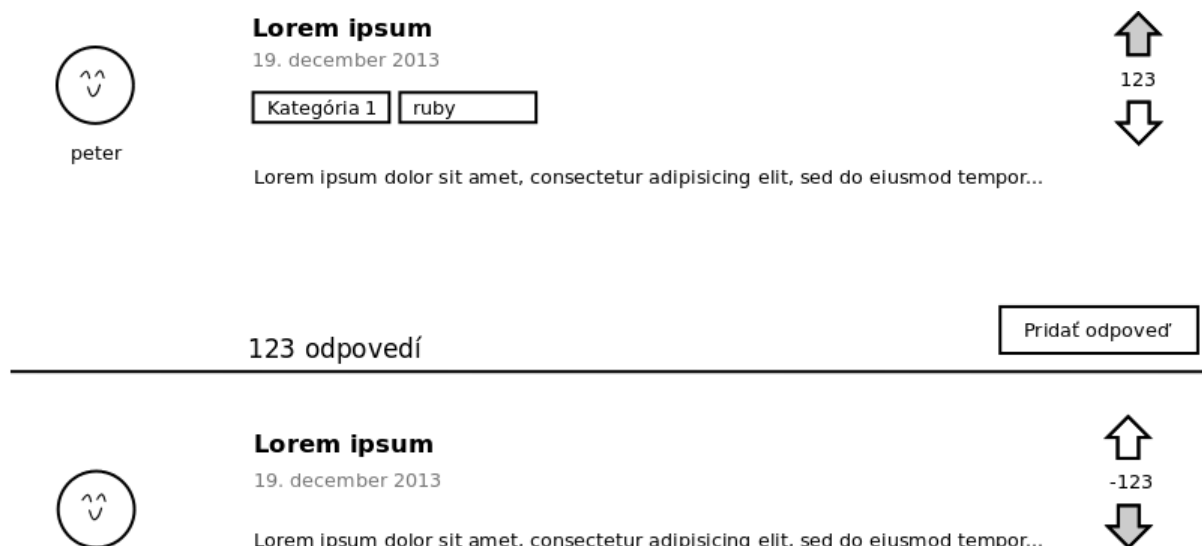
6.1 Hlasovať za otázku alebo odpoveď

Ako používateľ chcem pozitívne alebo negatívne hlasovať za otázku alebo odpoveď.

6.1.1 Analýza

Používateľ bude môcť hlasovať za otázku alebo odpoveď jedným hlasom. Výsledok hlasovania sa bude zobrazovať ako rozdiel medzi počtom kladných a záporných hlasov.

6.1.2 Návrh



Obrázok 6-1: Návrh rozhrania pre hlasovanie

Vedľa otázky alebo odpovede budú dve šípky slúžiace na hlasovanie. Šípka hore bude pridávať kladný hlas a šípka dole zase negatívny. Po kliknutí sa zmení farba šípky ako indikácia pripočítania hlasu. Kliknutie na už zadaný hlas tento hlas zruší.

Do dátového modelu sa pridá tabuľka Votes ktorá bude obsahovať hlasy používateľov. Bude obsahovať stĺpec upvote typu boolean určujúce či ide o kladný alebo záporný hlas.

6.1.3 Implementácia

Kedže hlasovanie je spoločné pre otázky aj odpovede tak je funkcionálna implementovaná v spoločnom `Concern`. Hlasovanie na úrovni modelu je implementované v `Votable` a pre kontroler vo `Voting`. Samotné hlasovanie je implementované v metóde `toggle_vote_by!`. Ak daný používateľ ešte nehlasoval vytvorí nový záznam. Pokiaľ sa mení hlas z negatívneho na pozitívny alebo opačne len sa prehodí logická hodnota v poli `upvote`. Nakoniec ak používateľ odznačil svoje hlasovanie tak sa záznam odstráni.

6.1.4 Testovanie

Testujú sa štyri scenáre použitia:

- hlasovanie pri otázke ktorá nemá hlas
- zmena hlasu
- zrušenie hlasu
- pridanie hlasu k už existujúcemu hlasu

V každom scenári sa kontroluje či sa správne zarátala vykonaná akcia do celkového výsledku hlasovania.

6.2 Označiť otázku ako obľúbenú

Ako používateľ chcem označiť otázku ako obľúbenú, aby som sa k nej mohol neskôr vrátiť.

6.2.1 Analýza

Existujúce prístupy využívajú hviezdu ako grafický prvok pre obľúbené prvky systému. Rovnakú sémantiku sme sa rozhodli využiť aj my, pričom okrem symbolu hviezdy ponúkame používateľom aj možnosť vidieť obľúbenosť otázky podľa počtu označení.



Obr. 6-2: Návrh rozhrania pre označenie otázky ako obľúbenej

6.2.2 Návrh

Tlačidlo so symbolom hviezdy pre označenie sme sa rozhodli umiestniť pod hlasovanie spolu s počtom označení.

6.2.3 Implementácia

Pre reprezentáciu obľúbených otázok v databáze sme vytvorili model `Favorite`. Tento model obsahuje referenciu na používateľa, ktorý otázku označil ako obľúbenú a referenciu na samotnú obľúbenú otázku.

Pohľad na otázku sme rozšírili o akciu `favor`. Akcia označenia otázky ako obľúbenej sa realizuje asynchrónne pomocou atribútu `data-remote`. Po dokončení akcie sa spustí spätné JavaScript volanie, ktoré nahradí element pre označenie otázky novým obsahom. Preto nie je potrebné, aby sa celá stránka generovala pri jednoduchej akcii ako označenie obľúbenej otázky. Pri každej otázke sa zobrazuje aj počet používateľov, ktorí ju označili ako obľúbenú.

6.2.4 Testovanie

Pre označenie otázky sme v modeli implementovali dve metódy – `toggle_favoring_by!` a `favored_by?`. Pre obe metódy sme vytvorili jednotkové testy pre model `Question`. Funkcionalitu označenia otázky sme realizovali pomocou akceptačného testu s využitím rozhrania Selenium¹⁷.

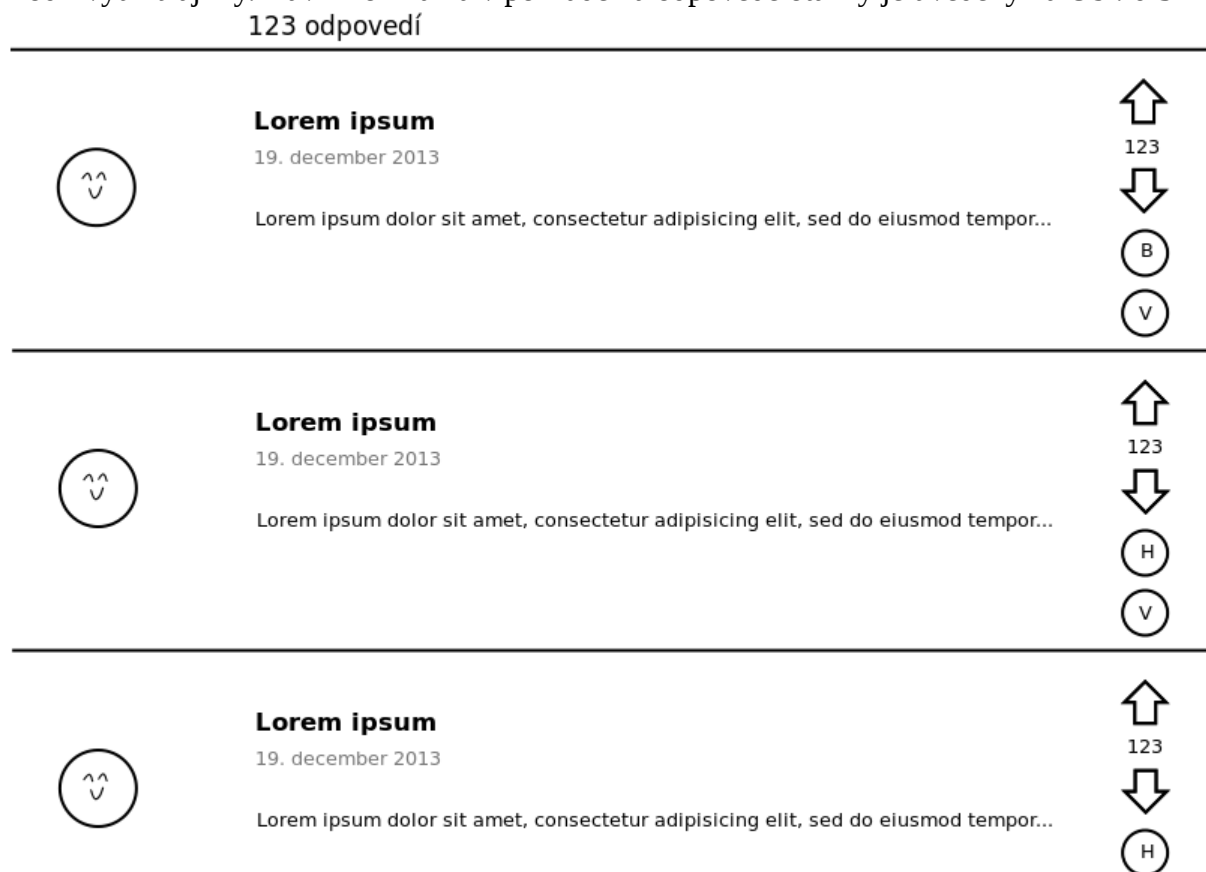
17 Selenium: <https://github.com/jnicklas/capybara>

6.3 Označiť odpoveď ako najlepšiu, pomocnú a overenú

Ako autor otázky chcem označiť najlepšiu odpoveď a rôzne pomocné odpovede. Ako učiteľ chcem označiť ľubovoľnú odpoveď ako mnou overenú. Ako používateľ chcem pri odpovediach k prehliadanej otázke vidieť najlepšiu, pomocné a overené odpovede, ak ich už niekto označil.

6.3.1 Analýza

Existujúce systémy poskytujú rôzne možnosti pre označenie odpovedí. Spravidla však ide o označenie najlepšej alebo pomocných odpovedí. V prípade edukačných systémov alebo systémov v ktorých vystupuje určitá autorita (učiteľ) má zmysel označovať odpovede ako overené (v zmysle správnosti) touto autoritou. Známe systémy používajú rôzne prvky pre označenie odpovedí. Pre označenie najlepšej odpovede je to zvyčajne zelená zaškrtnávací značka, pre pomocné napr. žltá žiarovka a pre overené biela zaškrtnávací značka na modrom podklade. Podobnú sémantiku sme sa rozhodli využiť aj my. Návrh rozhrania v pohľade na odpovede otázky je uvedený na Obr. 6-3



Obr. 6-3: Návrh rozhrania pre označenie odpovedí.

6.3.2 Návrh

Tlačidlá pre označenie odpovede (najlepšej *B*, pomocnej *H* a overenej *V*) sme sa rozhodli umiestniť

pod hlasovanie v stĺpci pod seba, pričom tie sa zobrazia ako vykonateľné akcie v prípade, že má používateľ dostatočné práva na označenie odpovede, inak sa zobrazia iba ako ikony. Súčasne sa napríklad nepovolí (a ani nezobrazí) označenie odpovede ako pomocnej ak už bola označená ako najlepšia.

6.3.3 Implementácia

Pre reprezentáciu obľúbených odpovedí v databáze sme vytvorili modely `Labeling` a `Label`. `Label` reprezentuje samotnú značku a `Labeling` označenie, t.j. väzbu medzi odpoveďou, značkou a používateľom (autorom značky).

Riadenie odpovedí sme rozšírili o akciu `label`. Akcia označenia odpovede požaduje na vstupe jediný parameter `value`, ktorý obsahuje hodnotu značky. Samotná akcia sa realizuje asynchrónne pomocou atribútu `data-remote`. Po dokončení akcie sa spustí spätné JavaScript volanie, ktoré nahradí ovplyvnené elementy (časti odpovedí) novým obsahom. Preto nie je potrebné, aby sa celá stránka generovala pri tak jednoduchej akcii ako označenie odpovede. Pri každom označení odpovede sa vykoná aj príslušná autorizácia a preverí sa či má používateľ dostatok práv na to aby mohol akciu vykonať.

6.3.4 Testovanie

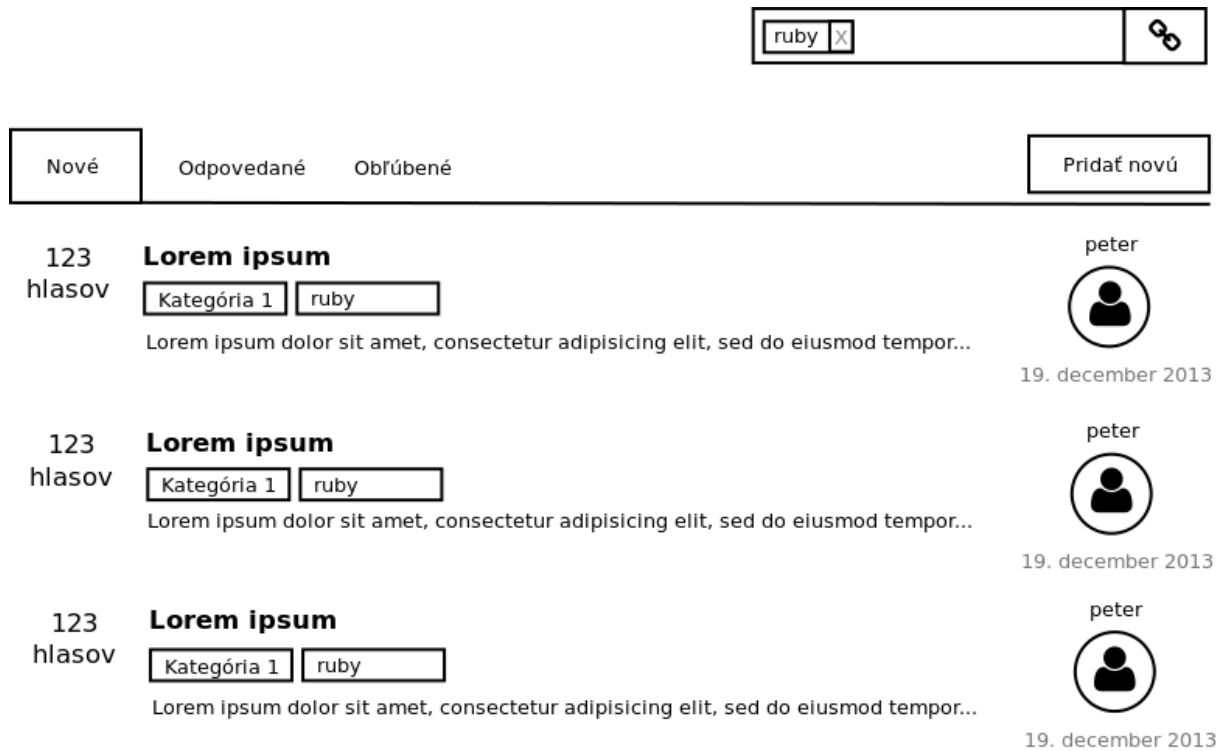
Pre označenie odpovede sme v modeli implementovali viaceré metódy, z ktorých spomenieme hlavne `toggle_labeling_by!`, `labeled_with` a `labeled_by?`. Pre spomenuté metódy sme vytvorili jednotkové testy v modeli `Answer`. Funkcionalitu označenia odpovede sme otestovali pomocou akceptačného testu, v ktorom sme pokryli všetky najdôležitejšie prípady aj s ohľadom na autorizáciu používateľa v danom kontexte.

6.4 Filtrovanie otázok podľa značiek

Ako používateľ, chcem filtrovať otázky podľa značiek (angl. tagov), aby som našiel otázky, ktoré ma zaujímajú.

6.4.1 Analýza

Pre podporu tejto funkcionality rozšírime pohľad na zoznam otázok o možnosť pridávania značiek ako filtra pre zoznam otázok. Filtru otázok vyhovujú len také otázky, ktoré obsahujú všetky značky špecifikované používateľom ako filter. Filtrovanie sa prispôbuje aktuálne zvolenej záložke otázok.



Obr. 6-4: Návrh rozhrania pre filtrovanie otázok

6.4.2 Návrh

Identifikovali sme, že veľká časť funkcionality pre výber značky do filtra je podobná s pohľadom pre vytvorenie novej odpovede. Preto sme sa rozhodli pridať túto rovnaké pole do rozhrania zoznamu otázok. Filter pre rozhranie bude brať do úvahy len otázky pre zvolenú záložku. Po kliknutí na značku uvedenú v zozname otázok sa značka pridá do filtrovacieho poľa. Vedľa vyhľadávacieho poľa je tlačidlo pre prekopírovanie aktuálneho dopytu do schránky (angl. clipboard).

6.4.3 Implementácia

Funkcionalitu filtrovania sme podporili pomocou asynchrónnych AJAX volaní na server. Pre perzistenciu parametrov ako záložka, strana a aktuálne zvolené značky v rámci URL adresy sme použili funkcionality HTML5 `pushState`¹⁸ pre prácu s históriou webového prehliadača. Pre automatické dopĺňanie značiek do filtrovacieho poľa sme použili knižnicu `select2`¹⁹, ktorú sme použili a prispôbili už v predchádzajúcom šprinte na implementáciu pridávania značiek pri vytváraní otázky. Funkcionalitu pre podporu pridania značky do filtra zo zoznamu otázok sme implementovali s využitím rozhrania knižnice `select2` pre dynamické pridávanie značiek.

18 HTML5 `pushState`: <http://www.w3.org/TR/2012/WD-html5-20121025/single-page.html#dom-history-pushstate>

19 `Select2`: <http://ivaynberg.github.io/select2/>

6.4.4 Testovanie

Filtrovanie sme otestovali pomocou rozhrania Selenium²⁰ pre interakciu s webovým prehliadačom. Vytvorili sme akceptačné testy pre viaceré prípady interakcie s filtrom pre značky. Prvý prípad berie do úvahy len filtrovanie podľa jednej značky. V rámci testu sme overili, či korektne fungujú aj asynchrónne stránkovanie otázok. Ostatné testy sa zamerali na filtrovanie podľa viacerých značiek a značiek zo zoznamu otázok, pričom sme kontrolovali aj korektné zobrazenie v prípade nenájdenných výsledkov.

²⁰ Selenium: <https://github.com/jnicklas/capybara>

7 Šprint 5 – Ultralisk

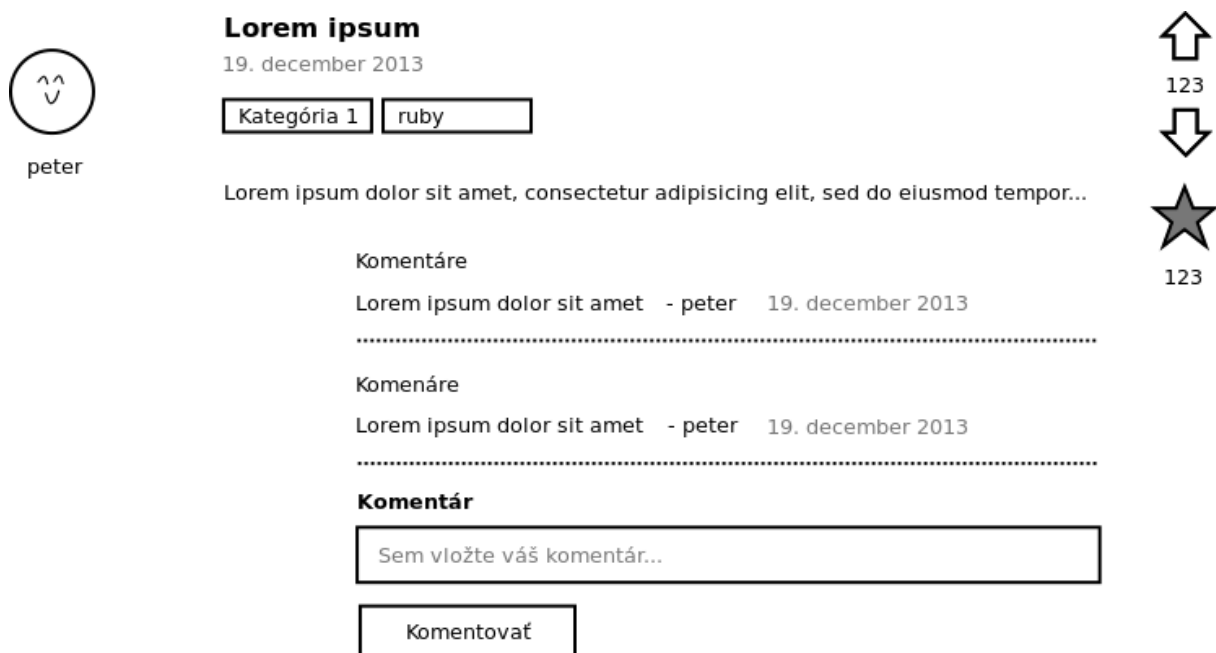
Posledný, piaty šprint zimného semestra, trval jeden týždeň a jeho náplňou bolo doladiť vzniknuté chyby, pridať možnosť filtrovania otázok, upraviť a doplniť niektoré funkcionality podľa požiadaviek, ktoré sme dostali od budúcich používateľov nášho systému.

7.1 Vloženie komentáru pre otázku a odpoveď

Ako používateľ chcem mať možnosť pridať komentár k otázke aj odpovedi, aby som bol schopný spýtať sa na detail, upozorniť na nedostatok alebo inak explicitne reagovať.

7.1.1 Analýza

Pre podporu tejto funkcionality dávame možnosť napísať komentár priamo pod otázku alebo odpoveď, ku ktorej sa má viazať. Na tomto mieste sa potom komentár aj zobrazuje, resp. sa tu nachádza zoznam komentárov zoradený chronologicky.



Obr. 7-1: Zobrazenie a vloženie komentáru

7.1.2 Návrh

Identifikovali sme, že model komentáru pre otázku aj odpoveď je identický, preto je preň možné použiť jednu tabuľku v databázovom modeli. Pre správne naviazanie konkrétneho komentáru k otázke alebo odpovedi sa preto použije polymorfická asociácia.

7.1.3 Implementácia

Pre vloženie komentáru sme vytvorili model `Comment`, ktorý obsahuje polymorfickú asociáciu `commentable`. Používateľ môže komentovať otázku samotnú, ale aj jednotlivé odpovede. Preto sme sa rozhodli na úrovni `routes` implementovať `concern`, ktorý nám umožní polymorfickú asociáciu pre otázky a odpovede. Na vytvorenie komentáru nám slúži `comment_controller`, v ktorom metóda `create` slúži na samotné vytvorenie a uloženie komentáru do databázy. Pomocou metódy `find_commentable` v `controller` zistíme pre aký typ objektu vytvárame komentár.

7.1.4 Testovanie

Pre vytvorenie komentáru sme vytvorili testy, v ktorých sme testovali úspešné a neúspešné vytvorenie komentára. Zobrazenie komentára sme otestovali na vlastných dátach. Sledovali sme zobrazenie jednotlivých komentárov, ich polohu a zarovnanie.

7.2 Zobrazenie počtu hlasov, odpovedí a zobrazení pri otázke

Ako používateľ chcem vedieť koľko má otázka hlasov, odpovedí a zobrazení.

7.2.1 Analýza

Počet hlasov je vyriešený v hlasovaní. Z neho sa použije už existujúca metóda. Taktiež počet odpovedí nám poskytuje dátový model. Jediná nová funkcionálna bude počet zobrazení. Zaznamenávať sa bude každé zobrazenie otázky. Zobrazovať ale bude len počet používateľov ktorý danú otázku videli.

7.2.2 Návrh

Zobrazenia sa bude zaznamenávať nová dátová entita `View`. Bude obsahovať len čas svojho vytvorenia a reláciu na používateľa a otázku. Keďže v budúcnosti sa môže vyskytnúť viacero entít ktoré budú zobraziteľné tak sa funkcionálna implementuje v samostatnom `concerne Viewable`. Tak bude možné v budúcnosti ľahko implementovať pre ďalšie entity.

7.2.3 Implementácia

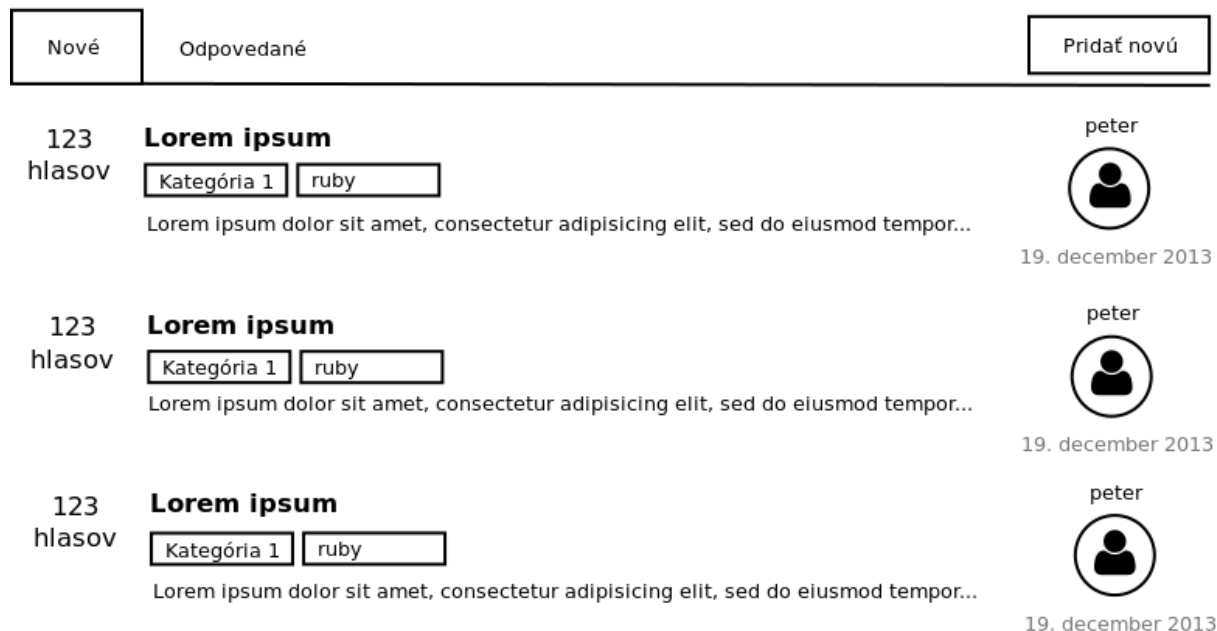
Zaznamenávanie zobrazenia otázky je vložené do metódy `show` v triede `QuestionController`. Počet zobrazení vracia metóda `total_views` z triedy `Viewable`.

7.3 Zobrazenie zodpovedaných otázok

Ako používateľ chcem vidieť všetky otázky, ktoré už majú odpoveď.

7.3.1 Analýza

Pohľad na zoznam otázok doplníme o navigáciu pomocou záložiek, ktoré budú obsahovať zvlášť nové a zvlášť odpovedané otázky, čo umožní používateľovi prehľadným spôsobom prechádzať cez otázky. Návrh môžeme vidieť na nasledovnom obrázku.



Obr. 7-2: Návrh zobrazenia pre odpovedané otázky

7.3.2 Návrh

Táto funkcionálnosť je doplnenie už existujúcej, ktorá vytvára zoznam otázok avšak zobrazuje zatiaľ len nové otázky. Implementovanie záložiek je pomocou *JavaScript*. Nasledovné vypisovanie zodpovedaných otázok prebieha pomocou sledovania v databáze, či je konkrétnej otázke priradená odpoveď označená ako najlepšia.

7.3.3 Implementácia

Implementácia je veľmi podobná ako pri nových otázkach. Využívajú sa rovnaké postupy. Doplnila sa len o prepojenie tabuliek a na základe existencie výskytu v tabuľke s odpoveďami sa otázka zaradí do zodpovedaných.

7.3.4 Testovanie

Testovanie rovnaké ako pri používateľskom príbehu zobrazenie nových otázok pomocou akceptačných testov. Kontroluje sa výpis zodpovedaných kde sa hľadá nadpis a kategória otázky.

7.4 Kategória ako pomenovaná množina značiek

Model `Category` bol doplnený o atribút `tags`. Tento atribút obsahuje pole textových reťazcov určujúcich značky pre kategóriu, ktoré sú pri vytváraní otázky s touto kategóriou pridané k ostatným značkám otázky. Túto funkcionality zabezpečuje metóda `add_tags` doplnená do modelu `Question`, ktorá je zavolaná pred vytvorením otázky.

Testy pre model `Question` boli doplnené o testy pre metódu `add_tags`.

8 Celkový pohľad po prvý kontrolný bod

Za prvé tri šprinty sa nám podarilo vytvoriť prototyp komunitného systému otázok a odpovedí, ktorý obsahuje nasledovné funkcionality:

- Autorizácia – V systéme existuje viac druhov používateľov, preto boli vytvorené rôzne právomoci pre rôzne skupiny.
- Lokalizácia – Náš systém bude obsahovať dve jazykové verzie. Implementované sú funkcie pre preklady do iného jazyku. Zatiaľ sú vytvorené len slovenské preklady.
- Prihlásenie – Do nášho systému je možné sa registrovať priamo ako nový používateľ alebo použiť prihlasovacie údaje z IS.
- Profil používateľa – Po úspešnom prihlásení sa do nášho systému si môže používateľ zobrazit' svoj profil. Okrem zobrazenia môže upraviť aj jednotlivé položky ako email alebo prezývku. Niektoré zmeny nie sú povolené za základe skupiny do akej používateľ patrí. Môže si nastavovať úroveň súkromia alebo pridávať odkazy na svoj profil na sociálnych sieťach.
- Vloženie novej otázky – Implementované je aj pridávanie otázok spolu s kategóriami a značkami.
- Zaznamenávanie udalostí – Nami vytvorený systém dokáže zaznamenávať aktivity používateľa, na základe ktorých bude možné vytvárať štatistiky alebo odporúčania.
- Zobrazenie otázky – Otázky sa zobrazujú v zozname zoradené podľa dátumu ich pridania. Spolu s nimi sa zobrazia aj značky, kategórie, dátum vytvorenia a autor otázky. Po kliknutí na konkrétnu otázku sa zobrazí celá otázka aj s odpoveďami.

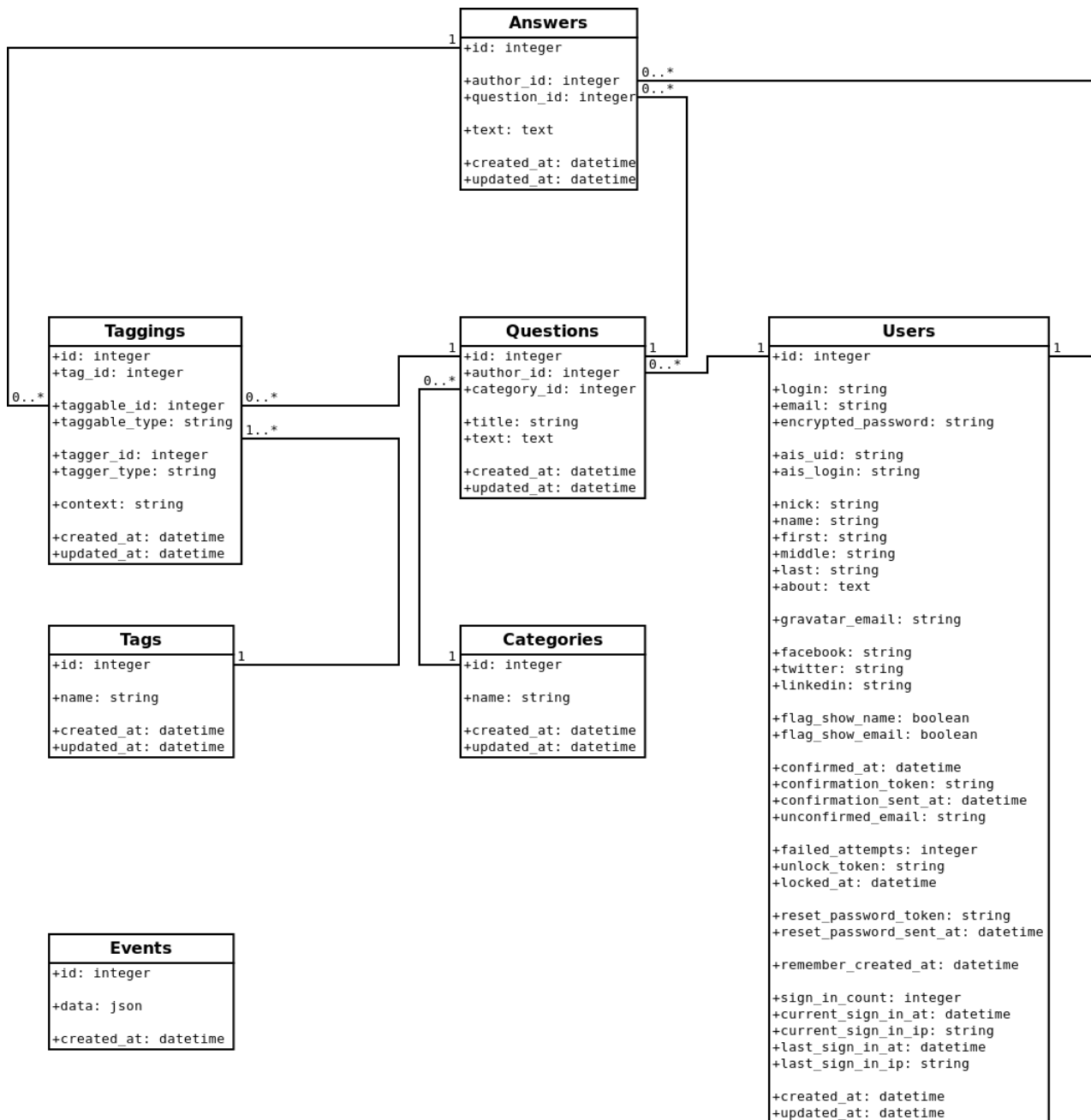
8.1 Architektúra

Z architektonického hľadiska ide zatiaľ o klasickú webovú aplikáciu založenú na návrhovom vzore MVC – modelu, pohľadu a kontrolóra (angl. model-view-controller). Na databázovej vrstve aplikácia používa relačnú databázu PostgreSQL 9, na aplikačnej vrstve sú to rámce Rails 4 (webový rámec MVC) a rámec Bootstrap 3 (rámec pre tvorbu používateľského rozhrania).

Okrem spomenutých rámcov je v projekte použité aj značné množstvo knižníc na rôznych úrovniach architektúry, napr. pri testovaní, asociovaní entít so značkami alebo autentifikácii používateľa. V tejto fáze prototypovania aplikácie je cieľom znovupoužiť čo najviac existujúcich riešení (knižníc), v neskorších fázach naopak podľa potreby implementovať vlastné riešenia.

8.2 Dátový model

Na Obr. 8-1 je znázornený vytvorený dátový model, ktorý obsahuje zatiaľ všetky použité entity.



Obr. 8-1: Dátový model

9 Celkový pohľad po druhý kontrolný bod

Od prvého kontrolného bodu sme do webového komunitného systému otázok a odpovedí doplnili nasledovnú funkcionálnosť:

- Vkládanie komentárov pre otázku – Používateľ môže okrem odpovede na otázku pridávať aj komentár k odpovedi.
- Hlasovanie – Pre každú otázku aj odpoveď je možnosť hlasovať pre každého používateľa a tým vyzdvihnúť odpovede alebo otázky.
- Značenia odpovedí – Odpovede môžu byť označené viacerými symbolmi. Prvý je označenie odpovede ako najlepšia. Toto značenie pridáva používateľ, ktorý otázku vytvoril. Takisto pridáva označenie pomocnej odpovede.
- Rola používateľa – V našom systéme sa predpokladá viac druhov používateľov. Zatiaľ identifikovaní sú študent a učiteľ. Každý z nich má iné práva.
- Oblíbené otázky – V systéme je možné pre každého používateľa vytvoriť si vlastný zoznam obľúbených otázok, ku ktorým sa vie jednoducho a rýchlo dostať.
- Štatistiky otázok – V zozname otázok sa pri každej otázke zobrazuje aj počet videní otázky, počet odpovedí a počet komentárov.
- Filtre – Pre rýchle vyhľadávanie otázok slúži filter, ktorý sa nachádza v zozname otázok. Kliknutím na značku alebo dopísaním značky priamo do poľa sa automaticky zobrazia len otázky vyhovujúce kritériám.

9.1 Architektúra

Architektúra systému nezaznamenala žiadne výrazné zmeny oproti prvému kontrolnému bodu. Na druhej strane však došlo k výraznému rozšíreniu dátového modelu dôsledkom implementácie novej funkcionality systému.

Z pohľadu modularizácie samotnej aplikácie na úrovni zdrojového kódu došlo k značnému refaktorovaniu najmä tried kontrolérov a modelov, resp. pomocníkov a pohľadov. Spoločná funkcionality bola vytiahnutá do tzv. *concerns* a *scopes*, resp. pomocných metód a *partial views*. Tento architektonický zásah významne prispel k aplikovaniu DRY princípu (angl. Do not Repeat Yourself, t.j. odstránení duplicitného alebo príliš podobného zdrojového kódu), znovupoužitelnosti a lepšej prehľadnosti zdrojového kódu.

10 Modul aktivita a nasledovanie

Modul aktivít a nasledovanie zahrňuje zaznamenávanie všetkej aktivity používateľov v systéme. Následne je možné zobrazit' aktivitu používateľov ktorých nasledujete.

10.1 Analýza

V systéme už existuje jeden modul ktorý zaznamenáva čo sa v systéme stane. Tento je ale založený na udalostiach a je až príliš jemný a aj formát nie je vhodný. Preto je treba vyvinúť druhý modul ktorý bude zachytávať vybranú aktivitu v systéme vo vhodnom formáte. Využije sa pritom modul notifikácií z ktorého sa budu preberať akcie. Tieto aktivity sa budú zobrazovať v globálnom zázname aktivít.

Používateľ si môže nastaviť že bude sledovať iného používateľa. Začnú mu potom chodiť notifikácie o jeho akciách ktoré si potom môže pozrieť v zázname aktivít nasledovaných používateľov.

10.2 Návrh

Všetka aktivita sa bude ukladať do modelu `Activity`. Táto entita bude obsahovať nasledovné atribúty: `initiator`, `resource`, `action`, `anonymous`. Tieto atribúty predstavujú kto danú aktivitu vykonal, predmet danej aktivity, akcia ktorá sa vykonala a či to bola akcia týkajúca sa anonymnej otázky. Akcia môže byť vytvorenie, úprava, vymazanie a spomenutie. Bude sa sa zaznamenávať aktivita týkajúca sa nasledovných tried: `Answers`, `Comments`, `Evaluations`, `Favorites`, `Followings`, `Labelings`, `Questions`, `Taggings`, `Views`, `Votes`, `Watchings`. Zobrazovať sa budú ale len nasledovné: `Answer`, `Comment`, `Evaluation`, `Labeling`, `Question`.

Nasledovanie sa bude ukladať do modelu `Following`. Táto entita bude mať dve asociácie na model `User` a to `follower` a `followee`. Prvý atribút je používateľ ktorý nasleduje a druhý je používateľ ktorého nasleduje. Pridanie nasledovania sa bude dať vykonať v zozname používateľov alebo v profile používateľa.

10.3 Implementácia

V module notifikácia je služba do ktorej sa vloží trieda `Activities::Feeder` ktorá počúva na prichádzajúce akcie. Z týchto akcií vytvorí novú aktivitu ktorá sa vloží do databázy. Pre zobrazenie aktivít je implementovaný `ActivityController` ktorý ma na starosti získať dva zoznamy aktivít. Zobrazenie iba vybraných aktivít je zabezpečené cez `default_scope` ktorá filtruje aktivity podľa typu `resource`. Okrem toho sa filtrujú iba aktivity ktoré majú nastavený `anonymous` na `false`.

Prvým zoznamom je globálny zoznam aktivít. Tento obsahuje aktivitu všetkých a používateľov v systéme. Používa sa špecializovaný scope `:global` ktorý ruší filtrovanie cez `anonymous`. Zobrazia sa teda aj anonymné akcie. Ako iniciátor sa zobrazí anonymný používateľ. Druhý zoznam obsahuje aktivitu používateľov ktorých nasleduje aktuálne prihlásený používateľ. Tento zoznam neobsahuje anonymnú aktivitu. Personálny zoznam aktivít je aj v profile používateľa kde sa zobrazuje aktivita daného používateľa.

Pridanie sledovania implementuje metóda `follow` v `UserController`. V tej sa zavolá metóda `toggle_follow_by!` z concern `Followable`.

10.4 Testovanie

Aktivita je testovaná v nasledujúcich testoch:

- Aktivita keď sa spýta otázku.
- Aktivita keď sa odpovie otázka.
- Aktivita keď sa edituje otázka.

Nasledovanie je testované v nasledovných testoch:

- Nasledovanie používateľa z profilovej stránky.
- Zrušenie nasledovanie používateľa.
- Pridať nasledovanie zo zoznamu používateľov.
- Zrušiť nasledovanie zo zoznamu používateľov.
- Nasledovanie keď sa sa kontroluje zoznam nasledovníkov.
- Nasledovanie keď sa sa kontroluje zoznam nasledovaných.

11 Modul administrácia, sli.do, označovanie správnych odpovedí, štatistiky

Tento modul sa zaoberá funkciami prístupnými rolami (používateľmi) s špecifickými právami. Modul je rozdelený na 4 časti:

- administrácia,
- sli.do,
- štatistiky,
- označovanie správnych odpovedí.

11.1 Analýza

Zo spomínaných častí tohoto modulu je najpopisnejšia administrácia, nakoľko sa zaoberá práve rozdeľovaním práv medzi jednotlivými rolami.

Je vytvorená podpora pre viaceré roly na jedného používateľa. Jednotlivé roly sú zasa viazané na konkrétnu kategóriu, čiže používateľ môže mať rôzne roly v rôznych kategóriách. V systéme Askalot rozoznávame tieto roly:

- študent
- učiteľ
- administrátor

Medzi nimi existuje hierarchické usporiadanie, kde najmenšie práva má študent a najväčšie administrátor. Platí pri tom, že každá vyššia rola obsahuje práva tej nižšej.

sli.do je externý systém pre vytváranie otázok. Tento systém je integrovaný s Askalotom v zmysle archivovania týchto otázok do databázy. Pre tieto potreby je vytvorený špeciálny používateľ sli.do. Od ostatných používateľov sa líši najmä v tom, že za ním nevystupuje fyzická osoba a odpoveď na jeho otázku poskytnutú učiteľom je priamo označená ako najlepšia.

Označovanie najlepšej odpovede je v kompetencii autora otázky, ktorý ma takto označiť len jednu odpoveď. Výnimku tvorí práve používateľ sli.do (viď vyššie). Autor môže svoje rozhodnutie kedykoľvek zmeniť. Táto odpoveď je zobrazovaná v zozname odpovedí otázky najvyššie.

Štatistiky poskytujú súhrnné informácie o vyvíjanej aktivite v systéme. Logicky ich môžeme deliť

na informácie o:

- všetkých používateľoch (súhrnné),
- otázkach,
- odpovediach,
- jednotlivých používateľoch.

Dôležitým prvkom štatistík je aj filtrovanie podľa značiek a dátumu (dátumového intervalu).

11.2 Návrh

V modeli rolí je kľúčovou práve entita `Roles`. V nej sú ukladané jednotlivé roly, ktoré systém rozpoznáva. Ich návaznosť na kategórie a na používateľa je realizovaná prostredníctvom väzobnej entity `Assignments`. Pre definíciu konkrétnych práv jednotlivých používateľov je použitá knižnica `CanCan`.

Pri aktuálne prebiehajúcej `slido` akcií vznikne záznam v entite `SlidoEvents` je zobrazená informácia o aktuálne prebiehajúcej `slido` akcií v zozname otázok. Kategórie, ktoré spolupracujú so systémom `slido`, majú pre tento systém vlastné používateľské meno (`slido_username`), čím je naznačené ich spojenie.

Označenia správnych odpovedí sú ukladané v entite `Labels`. Tieto označenia môže vytvárať len autor odpovede, na ktorú sú tieto označenia cez väzobnú entitu `Labelings` napájané.

Štatistiky predstavujú jednoduchý index, kde sú agregované informácie o jednotlivých subjektoch systému, nie je preto potrebný žiadny širší model.

11.3 Implementácia

Práva jednotlivých rol sú určované za pomoci autorizačnej knižnice `CanCan`, pričom konkrétne sú implementované v triede `Ability`.

Pre roly sú takisto dôležité metódy v triede `User`, konkrétne `role?`, ktorá vracia, či existuje kategória, pre ktorú má tento používateľ danú rolu; a `assignment?`, ktorá vracia akú rolu má používateľ pre danú kategóriu.

Pre všetky kategórie, ktoré majú definovaný `slido_username`, sa `scraper.rb` uložený v adresári `lib/slido` pozrie, či momentálne neprebíha `slido` akcia. Pokiaľ áno, je uložená v `SlidoEvents`.

Pokiaľ práve prebieha táto akcia a je uložená v databáze, je o nej zobrazená informačná flash správa, čo zabezpečuje `flash.rb` v `app/controllers/slido`.

Označovanie odpovedí zabezpečuje `answers_controller`, konkrétne jeho metóda `label`.

Štatistiky využívajú podporu matematických operácií, ktoré sú implementované v `statistic_helper.rb`.

11.4 Testovanie

Modul je z časti pokrytý akceptačnými testami. Jednotlivé testy zabezpečujú nasledovné scenáre:

- Označenie odpovede ako najlepšej.
- Odstránenie označenia odpovede ako najlepšej.
- Ukázanie notifikácie o aktuálnom priebehu `sli.do` udalosti.
- Ukázanie novej notifikácie o aktuálnom priebehu `sli.do` udalosti pri automatickom obnovení stránky.

Akceptačnými testami nie je pokrytý model rôl. Takisto nie sú k dispozícii akceptačné testy pre štatistiky.

12 Modul úprava, mazanie a anonymné otázky

Modul umožňuje upraviť obsah otázok odpovedí a komentárov potom čo už boli odoslané. Umožní sa tým opraviť chyby, doplniť, rozšíriť alebo spresniť otázku, odpoveď alebo komentár.

Taktiež podporuje zmazanie otázky, odpovede, komentára, hlasovania, hodnotenia, označenia, nasledovania, oštitkovanie, sledovania a obľúbenia.

Obsahuje tiež pýtanie sa anonymných otázok. Zabezpečuje pridanie otázky ako anonymnej a zobrazenie anonymného používateľa pri zobrazení autora otázky.

12.1 Analýza

Editovať sa budú dať všetky príspevky používateľov. Teda otázky, odpovede, komentáre a hodnotenie. Pri editovaní obsahuje je potrebné uložiť predchádzajúcu verziu príspevku. Používateľ bude môcť editovať len príspevky ktorých je sám autorom. Administrátor bude môcť editovať všetky príspevky.

Pri používaní systému je potrebné odstrániť zo systému rôzne entity. Napríklad používateľ prestane sledovať otázku. Treba vtedy odstrániť entitu ktorá označuje že sledoval danú otázku. Alebo chce odstrániť svoju odpoveď. Pri zmazení sa ale daná entita neodstráni natrvalo. Iba sa označí ako zmazená a prestane sa tak zobrazovať. Je potrebné sa taktiež postarať aby sa zmazali aj všetky entity naviazané na zmazenú entitu. Teda napríklad v prípade otázky zmazať všetky odpovede, komentáre k tejto otázke, hlasovania.

Zmazanie otázky má byť pre používateľa možné len ak k nej neexistujú odpovede, komentáre a hodnotenie. Podobne odpoveď má byť možné zmazať iba ak k nej nie sú žiadne komentáre.

Niektorí používatelia sa môžu cítiť lepšie, ak nie je vidno, že sa danú otázku pýtajú oni. Ak sa môžu spýtať anonymne, umožní to, aby sa spýtali aj na menej príjemné otázky.

12.2 Návrh

Predchádzajúce verzie príspevkov sa ukladajú do samostatných modelov. Názvy modelov sú `Answer::Revision`, `Comment::Revision`, `Question::Revision`. Pri editovaní sa vytvorí nová revízia príspevku. Pri vytvorení sa taktiež nastaví príznak že bol daný príspevok už upravovaný. Pri zobrazení príspevku sa v prípade že bol upravovaný zobrazí čas poslednej úpravy spolu s používateľom ktorý danú zmenu vykonal.

Do každej entity ktorú je možné zmazať sa pridávajú nasledovné atribúty: `deleted`, `deleted_at`, `deletor_id`. Obsahujú či bola daná entita zmazená, kedy bola zmazená a kým bola zmazená.

To, že je otázka anonymná zabezpečí boolean atribút v tabuľke s otázkami. Okrem toho je potrebné zabezpečiť aby sa aktivita súvisiaca s anonymnou otázkou neobjavila v personálnom logu aktivity a upozornení. Taktiež aj v profile používateľa je zoznam otázok v ktorých sa anonymné otázky nesmú objaviť. Pri upozorneniach je potrebné zabezpečiť aby sa ako iniciátor objavil anonym. V globálnom logu aktivít sa taktiež musí zobrazovať anonym ako iniciátor aktivity a v logu nasledovaných používateľov sa neobjaví vôbec.

12.3 Implementácia

Na podporu úpravy v kontroléroch je vytvorený samostatný concern `Editables::Update`. Tento je vložený do kontrolérov entít pri ktorých sa vykonáva úprava. Konkrétne do `AnswerController`, `CommentController`, `QuestionController`. Kontrolér `Editables::Update` definuje metódu `update` v ktorej sa získa trieda entity pre ktorú sa ma vytvoriť nová revízia. Nakoniec sa zavolá metóda `create_revision!` ktorá je definovaná pre každú z `Revision` tried.

Mazanie je implementované v concern `Deletables`. Definuje nasledovné metódy:

- `deleted_or_new!` - v tejto metóde sa vráti už existujúca zmazaná entita alebo sa vytvorí nová.
- `mark_as_deleted_by!` - táto metóda označí danú entitu za zmazanú. Taktiež rekurzívne označí všetky entity ktoré závisia na entite taktiež za zmazané.
- `toggle_deleted_by!` - prepne zmazaný stav na opačný
- `mark_as_deleted_recursive!` - rekurzívne zmaže akékoľvek entity ktoré majú nastavenú asociáciu `:has_many` a obsahuje atribút `deleted`.

Okrem toho sa pri mazaní aktualizujú počítadla v ktorých je uložený počet asociovaných entít. To zabezpečuje metóda `update_counter_caches!` v ktorej sa každá `:belongs_to` asociácia s nastaveným počítadlom automaticky aktualizuje aby číslo v počítadle reflektovalo reálny počet.

Pridali sme boolean atribút k otázke že je anonymná. Pri zobrazovaní sa volajú pomocné funkcie na zobrazenie mena a avataru používateľa. Tieto sa upravili tak aby pri anonymnej otázke vykreslili anonymného používateľa. Týmto je zabezpečené že sa anonymná zobrazí ako anonymná. V profile používateľa je zoznam otázok ktoré daný používateľ položil kde sa anonymné otázky nesmú objaviť. Toto je zabezpečené pomocou `where(anonymous: false)` klauzuly pri dopytovaní sa na otázky v `UserController`. Používateľ si môže zobrazit' svoje vlastné anonymné otázky v samostatnej záložke. Pri upozorneniach a aktivitách sa narazilo na komplikáciu s polymorfickými asociáciami. Keďže anonymný príznak majú len otázky tak to ostatným resource chýba. Vyriešili sme to pridaním anonymného príznaku do aktivít aj notifikácii. Tento príznak je potom štandardne `false` a len pri anonymných otázkach sa nastaví na `true`. Pre aktivity sú potom pripravené scopes

ktoré vracajú všetku aktivitu používateľa ak si prezerá svoj profil alebo odfiltrovanú anonymnú aktivitu ak si pozerá profil iného používateľa.

12.4 Testovanie

Editovanie je testované v nasledujúcich testoch:

- Editovanie keď otázka nemá hodnotenie – môže editovať otázku.
- Editovanie keď otázka nemá hodnotenie – môže editovať odpoveď.
- Editovanie keď otázka nemá hodnotenie – môže editovať komentár.
- Editovanie keď otázka má hodnotenie – používateľ nemôže editovať otázku.
- Editovanie keď otázka má hodnotenie – administrátor môže editovať otázku.
- Editovanie keď je prihlásený ako iný používateľ – nemôže editovať.

Mazanie je testované v nasledujúcich testoch:

- Mazanie keď otázka nemá odpovede a komentáre – môže zmazať otázku
- Mazanie keď odpoveď má odpoveď bez komentára – môže zmazať odpoveď
- Mazanie keď otázka má odpoveď s komentárom – môže zmazať komentár k otázke
- Mazanie keď otázka má odpoveď s komentárom – môže zmazať komentár k odpovedi
- Mazanie keď je prihlásený ako autor otázky a má odpoveď s komentármi – nemôže zmazať otázku
- Mazanie keď je prihlásený ako autor otázky a má odpoveď s komentármi – nemôže zmazať odpoveď
- Mazanie keď je prihlásený ako autor otázky a má odpoveď s komentármi – nemôže zmazať komentár
- Mazanie keď otázka má odpoveď a komentáre a používateľ nemá oprávnenia – nemôže zmazať otázku
- Mazanie keď otázka má odpoveď a komentáre a používateľ nemá oprávnenia – nemôže zmazať odpoveď

- Mazanie keď otázka má odpoveď a komentáre a používateľ nemá oprávnenia – nemôže zmazať komentár
- Mazanie keď otázka má odpoveď a komentáre a používateľ je administrátor – môže zmazať otázku
- Mazanie keď otázka má odpoveď a komentáre a používateľ je administrátor – môže zmazať odpoveď
- Mazanie keď otázka má odpoveď a komentáre a používateľ je administrátor – môže zmazať komentár

Anonymné otázky sú testované v nasledujúcich testoch:

- Pridanie novej otázky anonymne
- Zobrazenie profilu keď je prihlásený ako autor anonymnej otázky
- Zobrazenie profilu keď je prihlásený ako iný používateľ

Okrem toho bolo aj ručne kontrolované aby sa otázka položená anonymne tak aj zobrazovala. Taktiež sa skontrolovalo že sa neobjavila anonymná aktivita v personálnom logu aktivít a v logu sledovaných používateľov a či je v globálnom logu sa zobrazuje ako iniciátor anonym.

13 Modul filtrovanie a vyhľadavanie

Modul slúži na filtrovanie a ľahšiu orientáciu (aj vyhľadavanie) v otázkach, odpovediach, komentároch, kategóriách, značkách, používateľoch. Je rozdelený na dve časti:

- Filter – filtrovať je možné buď podľa záložiek v zozname danej entity, alebo podľa značiek priradených danej entite.
- Fulltextové vyhľadavanie – vyhľadavanie v otázkach, odpovediach, komentároch, kategóriách, značkách a používateľoch pomocou fulltextových dopytov.

13.1 Analýza

Filtrovanie aj vyhľadavanie je nad viacerými entitami (otázky, odpovede, komentáre, kategórie, značky, používatelia). Ďalej v texte sa na ne budeme všeobecne odkazovať ako na entitu.

Pri zobrazení zoznamu otázok je bez ich následného filtrovania náročné vyhľadať otázky, ktoré používateľa zaujímajú. Preto je potrebné otázky filtrovať a vedieť v nich vyhľadať. Tento modul rieši problém navigácie v otázkach.

Otázky by mali byť filtrované podľa rôznych pohľadov na ne. Identifikovali sme 5 skupín na ktoré sú otázky v zozname rozdelené:

- Aktuálne – všetky otázky, zoradené v poradí podľa poslednej interakcie s otázkou.
- Bez odpovede – otázky, ktoré nemajú žiadnu odpoveď.
- Odpovedané – otázky, ku ktorým je pridaná aspoň jedna odpoveď.
- Vyriešené – otázky, ktoré majú zvolenú najlepšiu odpoveď.
- Obľúbené – otázky, ktoré má aspoň jeden používateľ zaradené medzi svoje obľúbené otázky.

Rozdelenie do skupín je realizované pomocou záložiek, v ktorých sa zobrazujú aj počty otázok v jednotlivých skupinách. Otázky v záložke *Aktuálne* sú zoradené podľa času poslednej interakcie s nimi (parameter `touched_at`). Otázky v ostatných záložkách sú zoradené podľa počtu hlasov, ktoré otázka dostala.

Okrem týchto piatich skupín je možné otázky filtrovať podľa značiek (angl. tags). Filtrovanie podľa značiek je realizované pridávaním jednotlivých značiek do formulára umiestneného navrchu zoznamu. Otázky sa následne filtrujú podľa aktuálneho výberu. Značky je do formulára možné pridávať ručne, alebo kliknutím na značky pri otázkach.

Fulltextové vyhľadávanie je tiež realizované pomocou formulára, do ktorého používateľ píše svoj dopyt. Po odoslaní formulára sa používateľovi zobrazí zoznam otázok vyhovujúcich dopytu v poradí podľa relevancie. Toto poradie vráti Elasticsearch po spracovaní dopytu. Otázky sú zobrazené iba v jednej záložke, nie sú delené na 5 skupín ako pri filtrovaní.

Formuláre pre filtrovanie otázok a fulltextové vyhľadávanie sa v používateľskom rozhraní javia ako jeden formulár v ktorom používateľ prepína funkcionality.

Kategórie je možné filtrovať podľa názvu kategórie. Nachádza sa tu formulár na fulltextové vyhľadávanie.

Zoznam značiek je možné filtrovať podľa mena značky, rovnako ako kategórie, alebo podľa troch rôznych pohľadov na ne. Tieto pohľady sú rozdelené do záložiek:

- Všetky – obsahuje všetky značky, ktoré sa v systéme nachádzajú.
- Nové – obsahuje značky, ktoré boli pridané za posledný mesiac.
- Populárne – obsahuje značky, ktoré sú v systéme aktuálne najpopulárnejšie, je v nich najviac aktivity. Zobrazuje sa tu 30 značiek s najväčšou aktivitou.

Používateľov je tiež možné filtrovať podľa ich prezývky alebo pomocou záložiek. Tu sú prítomné dve záložky:

- Všetci – všetci používatelia nachádzajúci sa v systéme.
- Noví – používatelia, ktorí sa zaregistrovali za posledný mesiac.

13.2 Návrh

Pri filtrovaní zoznamu jednotlivých entít nie je žiadúce znova generovať celú stránku so zoznamom otázok. Preto pri zmene filtra využívame na aktualizáciu dát spätné JavaScript volania – AJAX, za pomoci knižnice jQuery.

Pre implementáciu fulltextového vyhľadávania sme sa rozhodli použiť vyhľadávací stroj Elasticsearch²¹. Je to open source distribuovaný vyhľadávací stroj pracujúci v reálnom čase. Pri inicializácii sa do Elasticsearch naimportujú dáta vo formáte JSON, pretože Elasticsearch databáza nemá pevnú schému.

13.3 Implementácia

Na vyhľadávanie aj filtrovanie otázok je v pohľade na zoznam otázok formulár pomocou ktorého je

21 <http://www.elasticsearch.org/>

možné tieto akcie vykonávať. V používateľskom rozhraní sa tieto formuláre javia ako jeden. V skutočnosti sú to dva formuláre, medzi ktorými sa môže používateľ prepínať pomocou tlačidiel ovládaných JavaScript-om. Toto prepínanie medzi formulármi je implementované za použitia elementov a metód triedy Collapse z rámca Bootstrap.

Filtrovanie otázok sa vykonáva dvoma spôsobmi. Jedným je rozdelenie na 5 skupín podľa určitých vlastností otázok. Toto rozdelenie je realizované v modeli `Question` pomocou vopred zadaných `scope`. Pomocou operátora `scope` je možné definovať metódu pre často používané dopyty do databázy. Záznamy sa následne vyberú na základe takto definovaných podmienok. V modeli `Question` je zadaných 5 takýchto metód pre vrátenie otázok podľa určitých kritérií. Názvy týchto metód sú samo vysvetľujúce: `recent`, `unanswered`, `answered`, `solved`, `favored`.

Filtrovanie podľa značiek je v pohľade na zoznam otázok implementované ako formulár, v ktorom sa pri písaní značky automaticky dopĺňajú. Akcia, ktorá robí dopyt do databázy na získanie existujúcich značiek využíva atribút `data-remote`. Automatické dopĺňanie značiek využíva asynchrónne JavaScript volanie, ktoré komunikuje so serverom a mení aktuálne dopĺňané značky na základe vrátených výsledkov. Po odoslaní formulára (doplnenie novej značky spustí túto akciu) sa načítajú otázky vyhovujúce filtru pomocou spätného JavaScript volania. Nie je preto potrebné znova načítať celú stránku so zoznamom otázok, stačí iba aktualizovať zoznam otázok.

Filtrovanie kategórií je možné podľa názvu kategórie. Vyhľadávanie v názvoch je realizované pomocou vyhľadávacieho stroja Elasticsearch. V pohľade na zoznam kategórií je prítomný formulár do ktorého môže používateľ písať meno kategórie.

Filtrovanie značiek je v pohľade na zoznam značiek implementované pomocou záložiek. Tieto záložky obsahujú zoznamy značiek filtrované podľa určitých kritérií (všetky, nové, populárne). Dopyty do databázy pre nové a aktuálne značky sú definované ako metódy v modeli `Tag` pomocou operátora `scope` (`recent`, `popular`). Rovnako ako pri kategóriách, filtrovanie je možné aj podľa názvu značky.

Filtrovanie používateľov je implementované analogicky. Skupiny sú pri používateľoch dve – všetci a noví (metóda `recent` definovaná pomocou `scope`). V zozname používateľov je tiež možné vyhľadávať podľa prezývky používateľa. Toto vyhľadávanie je implementované pomocou stroja Elasticsearch.

Elasticsearch

Nad strojom Elasticsearch sme vytvorili obalovaciu knižnicu `Probe`. Táto knižnica poskytuje rozhranie na importovanie otázok do stroja Elasticsearch. Otázky sa zaindexujú a potom nad nimi pracuje vyhľadávací stroj. Knižnica tiež poskytuje vlastné rozhranie na tvorbu indexu pre model, vyhľadávanie v naimportovaných dátach.

Po odoslaní formulára s používateľovou požiadavkou na fulltextové vyhľadávanie sa táto

požiadavka odošle na server pomocou spätného JavaScript volania. Dáta sú teda z databázy získané asynchrónne. Po odoslaní požiadavky na fulltextové vyhľadávanie sa na strane servera najskôr vyhľadajú otázky vyhovujúce požiadavke používateľa pomocou stroja Elasticsearch. Výsledky ktoré vráti sú zoradené podľa relevancie k používateľovej požiadavke. Elasticsearch má interne implementované rôzne algoritmy na určovanie podobnosti textu s používateľovou požiadavkou. Tieto záznamy sú následne získané z relačnej databázy v takom poradí id aké vrátil Elasticsearch. Toto zabezpečuje funkcia `array_idx`, ktorá záznamy v relačnej databáze zoradí podľa vopred daného zoznamu id.

```
CREATE OR REPLACE FUNCTION array_idx(anyarray, anyelement)
  RETURNS int AS
  $$
  SELECT i FROM (
    SELECT generate_series(array_lower($1,1),array_upper($1,1))
      ) g(i)
  WHERE $1[i] = $2
  LIMIT 1;
  $$ LANGUAGE sql IMMUTABLE;
```

Funkcionalita, ktorá rozširuje model o možnosť použitia funkcie `array_idx` je implementovaná v concern `Orderable`.

Model, ktorý chce používať rozhranie knižnice `Probe`, musí zahŕňať concern `Searchable`. Tento concern implementuje databázové spätné volanie na import každej novej otázky do vyhľadávacieho stroja Elasticsearch. Implementuje tiež metódu `search` na vykonanie dopytu do Elasticsearch.

```
module Searchable
  extend ActiveSupport::Concern

  included do
    include Orderable
    include Probe

    after_save do
      self.class.probe.index.import(self)
    end

    after_destroy do
      self.class.probe.index.import(self, action: :delete)
    end
  end

  module ClassMethods
    def search(query = {})
      size = query.delete(:per_page) || 20
      from = (query.delete(:page) || 0) * size
    end
  end
end
```



```
    results = probe.search(query.reverse_merge(from: from, size: size))
      self.order_by(id: results.map(&:id))
    end
  end
end
```

Každý model, ktorý chce obsahovať funkčnosť implementovanú v Searchable musí zahŕňať concern, ktorý má názov `{Model}::Searchable`. Tento concern je potrebný pre zadenovanie Elasticsearch indexu pre daný model.

13.4 Testovanie

Modul je pokrytý testami na úrovni akceptačných testov. Príslušné modely, ktoré sú filtrované a nad ktorými sa vyhľadáva (`Question`, `Answer`, `Comment`, `Category`, `Tag`, `User`), sú pokryté taktiež jednotkovými testami. Akceptačné testy pre filtrovanie otázok pokrývajú nasledovné scenáre:

- filtrovanie otázok podľa jednej značky,
- filtrovanie otázok podľa viacerých značiek,
- filtrovanie otázok podľa značiek pridaných kliknutím na značku v otázke,
- filtrovanie otázok podľa značiek kategórie,
- zachovanie filtra podľa značiek pri zmene záložiek,
- správne filtrovanie otázok pri navigácii v histórii.

Fulltextové vyhľadávanie nie je pokryté akceptačnými testami, iba jednotlivé triedy a ich metódy sú pokryté akceptačnými testami. Hlavnou zložkou je knižnica `Probe`. Testuje sa import dát v dvoch základných módoch – po jednom zázname, alebo v dávkach. Otestované sú tiež triedy na vytvorenie indexu, vyhľadávanie a mapovanie záznamov. Tieto jednotkové testy pokrývajú kľúčovú funkčnosť knižnice `Probe`.

14 Modul zoznamy, zmeny v systéme, úvodná strana

Tento modul obsahuje opis všetkých indexov nachádzajúcich sa v projekte Askalot. Ďalej opisuje úvodnú stranu a popis zmien v systéme. Všetky tieto prvky majú prevažne informatívny charakter.

14.1 Analýza

Náš systém obsahuje niekoľko zoznamov. Sú to zoznamy nad používateľmi, otázkami, kategóriami, zoznamom zmien, značkami a aktivitami. Rovnako informatívny charakter majú aj úvodná strana. Všetky tieto funkcionality sú nevyhnutné pre používateľsky prívetivé prostredie.

Ako prvé čo používateľ uvidí v systéme je úvodná obrazovka. Na úvodnej obrazovke sú tlačidlá odkazujúce na najčastejšie vykonávané aktivity. Ďalej tu môže používateľ vidieť niektoré už položené otázky. Táto stránka slúži na rýchle presmerovanie používateľa na činnosť, ktorú chce vykonávať.

Zoznam otázok je rozdelený do viacerých záložiek podľa toho ako sú filtrované a zoradené. Popis týchto filtrov je bližšie opísaný v module filtrovanie. Celý zoznam otázok sa priebežne aktualizuje. V zozname otázok sa okrem výpisu jednotlivých atribútov o otázke vypisujú aj nasledovné informácie súvisiace s otázkou:

- Stav hlasovania pre otázku – Predstavuje aký je aktuálny súčet hlasov (negatívne + pozitívne)
- Počet odpovedí na otázku – Koľko odpovedí obsahuje otázka. V prípade, že otázka nemá ešte odpoveď, je červenou farbou.
- Počet zobrazení otázky – Koľko unikátnych používateľov si otázku pozrelo.
- Informácie o používateľovi, ktorý položil otázku – V prípade, že otázka nie je anonymná tak sa pri otázke zobrazí prezývka používateľa, jeho gravatar a dátum kedy bola otázka pridaná.

Okrem toho zoznam otázok obsahuje aj nasledovné funkcionality:

- Pridať novú otázku – Nad zoznamom sa nachádza tlačidlo na pridanie otázky.
- Vyhľadávanie – Na tejto stránke sa nachádza textové pole, do ktorého je možné napísať hľadaný výraz a zobrazia sa len otázky, ktoré ho obsahujú.

Používateľ tak má prehľad o stave otázky ktorá ho zaujíma.

Ďalším zoznamom je zoznam kategórií, ktorý obsahuje všetky kategórie v systéme. Kategórie sú zoradené podľa skupín. Pre každú kategóriu vidíme koľko otázok a odpovedí obsahuje. Pomocou tlačidla so symbolom oka je možné sledovať danú kategóriu.

Zoznam značiek funguje rovnako ako zoznam kategórií. Obsahuje len viac záložiek, pre lepšiu prehľadnosť.

Podstatnou súčasťou je zoznam používateľov. Ten je rozdelený do dvoch záložiek. Prvá obsahuje všetkých používateľov a druhá len nových. V zozname používateľov je zobrazený gravatar používateľa a jeho prezývka. Okrem toho je pri prezývke symbol na nasledovanie daného používateľa.

Pre informovanie používateľov o zmenách v systéme je vytvorený zoznam zmien v systéme. Ten obsahuje históriu zmien v systéme. Pre každý nový záznam sa zobrazuje jeho verzia prípadne názov a popisný text. Používateľ si tak ľahko môže pozrieť aké nové funkcionality pribudli v systéme. Verzia odkazuje priamo na Github. Na boku stránky je zoznam všetkých zmien systému pomocou, ktorých je možná rýchla navigácia.

14.2 Návrh

Pri jednotlivých zoznamoch je potrebné aby sme zobrazovali aj údaje z iných entít. Preto je potrebné mať prepojenia medzi týmito entitami. V niektorých prípadoch je nutné vytvárať aj väzobné entity. Príklad takejto entity môžeme vidieť pri otázkach a značkách. Tu nám vzniká prepojená trojica Questions-Taggings-Tags.

Vo väčšine zoznamov sa nachádza stránkovanie, ktoré je implementované pomocou `Kaminari`. Obsahuje všetky potrebné funkcie na presúvanie sa v stránkovaní ako je `next_page`, `last_page` a ďalšie.

14.3 Implementácia

V zozname otázok môžeme zobrazovať jednotlivé záložky, ktorých obsah je generovaný na základe definovaného operátora `scope`. Pomocou nich sa filtrujú len požadované informácie a zoradené podľa potreby.

Množstvo zoznamov využíva `kaminari` na stránkovanie. V ňom je zadaná navigácia stránkovania. Stránkovanie vďaka tomu umožňuje prehľadné zobrazenie. Pomocou `@questions = @questions.page(params[:page]).per(20)` je zadané aby jedna stránka obsahovala len 20 položiek. Tento počet môže byť pre každý zoznam rôzny. Pre otázky sú vytvorené záložky, ktoré určujú obsah. Každá z nich je generovaná pomocou operátora `scope`.

- `scope :recent, lambda { order(touched_at: :desc) }`

- `scope :unanswered, lambda { includes(:answers).where(answers: { question_id: nil }) }`
- `scope :answered, lambda { joins(:answers).uniq }`
- `scope :solved, lambda { joins(:answers).merge(best_answers).uniq }`

Recent generuje všetky otázky zoradené podľa atribútu `touched_at`, ktorý reprezentuje poslednú úpravu. Unanswered generuje otázky, ktoré ešte neboli odpovedané. Answered generujú otázky, ktoré obsahujú odpoveď. Solved generuje otázky, ktoré majú minimálne jednu odpoveď a označenie jednej odpovede ako najlepšej.

Pri zozname otázok je nastavený maximálny počet znakov, ktoré sa zobrazia. Napríklad pre text otázky je nastavený limit 250 znakov.

Pre vypisovanie kategórie a značiek sú položky vypisované pomocou `group` v skupinách po riadkoch, čím vznikajú tri stĺpce. Ešte predtým sa kategórie rozdelia do skupín na základe významu. To je vykonávané pomocou prefixu. Vďaka tomu dokážeme na základe zhody rozdeliť kategórie do skupín.

```
def group_categories_by_prefix(categories)
  mixed = []
  groups = categories.inject(Hash.new) do |groups, category|
    if match = category.name.match(/[A-Z]{2,}\s/)
      (groups[match[0].strip.downcase.to_sym] ||= []) << category
    else
      mixed << category
    end
  end

  groups
end

groups.sort_by { |key, _| key }
groups[:mixed] = mixed
groups
end
```

Pre zoznam kategórií a zoznam značiek sú jednotlivé položky vypísané ako odkazy. Po kliknutí na tieto odkazy sme presmerovaní priamo na zoznam otázok kde je použitý konkrétna značka alebo kategória ako atribút pri vyhľadávaní. To je spôsobené funkciou `Link_to_tag`, ktorá sa nachádza v `tags_helper.rb`.

```
def link_to_tag(tag, options = {})
  link_to tag.name, questions_path(tags: tag.name)
end
```

Zoznam používateľov obsahuje výpis všetkých používateľov systému abecedne zoradených podľa prezývky, alebo abecedne zoradených nových používateľov. Za nových používateľov sa považujú taký používatelia, ktorý sa do systému zaregistrovali posledný mesiac.

Na úvodnej stránke sú umiestnené linky na rýchle presmerovanie na najčastejšie aktivity v systéme. Ďalej úvodná stránka zobrazuje časť obsahu zoznamu otázok. Zobrazené sú v troch kategóriach. Prvá zobrazuje nové otázky. Druhá vyriešené a posledná obľúbené otázky.

```
Count = 4
```

```
Question.order(created_at: :desc).limit(count)
Question.solved.random.limit(count)
Question.favored.random.limit(count)
```

Ako vidíme v časti kódu všetky otázky majú nastavený počet na štyri aby nedošlo k zbytočnému zahlteniu úvodnej strany. Pričom prvé je zoradenie podľa času vytvorenia, druhá obsahuje vyriešené otázky a posledná obľúbené.

Z úvodnej strany sa vieme dostať k zobrazeniu zoznamu zmien v systéme. Ten je reprezentovaný verziou, ktorá nás presmeruje priamo na Github. Obsahuje aj názov zmeny a jej popis, ktoré však nie sú povinným atribútom. Celý zoznam zmien je usporiadaný podľa verzie. Zobrazenie zmien v systéme zahŕňa v sebe markdown.

14.4 Testovanie

Tento modul je pokrytý akceptačnými testami. Napríklad pre zoznam otázok boli vytvorené testy pre nasledovne scenáre:

- Zobrazenie zodpovedaných otázok.
- Zobrazenie obľúbených otázok.
- Zobrazenie nezodpovedaných otázok
- Zobrazenie vyriešených otázok

Na tieto účely boli v rámci testov vytvorené otázky, ktoré obsahovali riešenia, odpovede alebo ohodnotenia. Takisto sa testovali aj prázdne otázky bez odpovedí a ohodnotení.

Nad kategóriami boli vytvorené testy zamerané na:

- Kontrolu unikátnej hodnoty.
- Počet výskytu.
- Správne zobrazenia počtov po vymazaní otázky.

Keďže úvodná strana systému je jednoduchá a neposkytuje zložitejšiu funkcionálnosť, bola testovaná len ručne.

15 Modul Markdown, pomoc v systéme

Tento modul opisuje pomocné prvky v používateľskom rozhraní a prvky zlepšujúce skúsenosť používateľa so systémom. Modul obsahuje Markdown (jazyk pre zápis štruktúrovaného textu) s podporou pre Gemojis (emotikony). V tomto module je obsiahnutá aj pomoc pre celý systém.

15.1 Analýza

Skúsenosť používateľa s našim systémom je lepšia, ak môže otázky a odpovede písať ako štruktúrovaný text. Štruktúrovaným textom je jednoduchšie plne vyjadriť svoje myšlienky a takýto text sa následne aj jednoduchšie číta pri odpovedaní na otázku, resp. ďalšom štúdiu. Preto sme do nášho systému pridali možnosť zadávať štruktúrovaný text pomocou jazyka Markdown.

Pri písaní textu otázky a odpovede je možné použiť Markdown a zvoliť si náhľad na vygenerovaný text. Prepínanie medzi písaním textu a náhľadom je realizované cez záložky. V Markdowne pre otázky a odpovede je možné využívať formátovanie textu, referencovanie používateľov a otázok, vkladať bloky kódu, pridávanie emotikonov. Pre komentáre, je však možné použiť iba podmnožinu Markdownu, kde je možné referencovať používateľov a otázky, prípadne vkladať jednoriadkové kusy kódu.

Pre používateľa sme tiež vytvorili pomoc v systéme. Táto pomoc je koncipovaná ako množina scenárov, ktoré môžu pri používaní systému nastať, spolu s krokmi riešenia. Text je doplnený o obrazovky zo systému.

15.2 Návrh

Pre Markdown sme sa rozhodli použiť knižnicu `Redcarpet`, ktorá obsahuje procesory na spracovávanie Markdownu do HTML kódu. Nad touto knižnicou sme ešte vytvorili obalovaciu knižnicu `Redcurtain`.

Ďalej sme použili knižnicu `pygments.rb`, ktorá je obalovacia knižnica nad knižnicou na zvýrazňovanie syntaxe kódu v jazyku Python.

Pre emotikony sme použili knižnicu `gemoji`, ktorá obsahuje pomenované obrázky, ktoré sa dajú vkladať do textu.

15.3 Implementácia

Markdown

Pri implementácii Markdownu bol použitý Github Markdown, ktorý sme upravili pre naše potreby.

Nad knižnicou Redcarpet sme vytvorili knižnicu Redcurtain, ktorá ju obaľuje a dáva nám rozhranie pre prácu s Markdownom. Knižnica má vlastné renderery pre Gemoji, Pygments, Redcarpet a na linkovanie otázok a používateľov. Renderer skladajúci sa z týchto prvkov je následne použitý ako renderer pre Markdown v systéme.

V Markdowne je možnosť odkazovať sa na ostatných používateľov. Po napísaní znaku '@' a písaní prezývky sa začnú automaticky dopĺňať prezývky ostatných používateľov. Markdown poskytuje aj možnosť odkazovania sa na konkrétnu otázku. Po napísaní znaku '#' sa začnú automaticky dopĺňať nadpisy existujúcich otázok. Takáto referencia je následne spracovaná Markdown processorom ako odkaz na id tejto otázky. V texte je tiež možné pridávať Gemojis na vyjadrenie svojich emócií. Emotikony sa začnú dopĺňať po napísaní znaku ':'.

Markdown je v systéme použitý na viacerých miestach. Všade kde sa dá písať text do formulára, je možné použiť nejakú podmnožinu Markdownu, nie všade sú prístupné všetky jeho možnosti. Pri otázkach a odpovediach je prístupný celý Markdown. Pri komentároch je prístupná iba jeho podmnožina a to odkazovanie sa na používateľov, otázky a písanie jednoriadkových kúskov zdrojového kódu.

Markdown sa dá použiť aj pri vyplňaní časti about v profile používateľa.

Pomoc pre používateľa

Pomoc pre používateľa je v systéme prístupná v dvoch podobách. Pod adresou /help je prístupná pomoc pre celý systém s obrazovkami zo systému. Táto pomoc je koncipovaná ako riešenie možných scenárov, ktoré môžu nastať. Technicky je riešená ako statická stránka.

Stručnejšia verzia pomoci pre systém je používateľom ponúknutá pri vytváraní novej otázky. V pohľade na otázku je pomoc umiestnená v pravej polovici obrazovky. Nachádzajú sa v nej tipy ako položiť otázku čo najlepšie a ako čo najlepšie vyplniť jednotlivé polia.

Pohľad na pomoc pre systém je rozdelený na viacero častí pomocou partials. Každý scenár je popísaný zvlášť v samostatnom súbore a text je písaný pomocou Markdownu. Lokalizácia pri týchto súboroch nie je riešená cez konfiguračný súbor v config/locales, ale pomocou partials v rôznych jazykových mutáciách. Rámec Rails podporuje aj tento spôsob lokalizácie. ActionController je schopný generovať rôzne pohľady v závislosti od lokalizačnej premennej. Stačí, ak súbory s pohľadmi obsahujú vo svojom názve danú lokalizačnú premennú. Vzor pre meno súboru je [filename].[locale].html.erb

15.4 Testovanie

Pre Markdown sú akceptačnými testami otestované nasledovné scenáre:

- Markdown pri otázke zobrazuje náhľad na otázku

- Markdown pri otázke zobrazuje emotikony
- Markdown pri otázke odkazuje na používateľa
- Markdown pri otázke odkazuje na otázku
- Markdown pri otázke, pri vložení linky na otázku odkazuje na otázku
- Markdown pri otázke pri vložení odkazu na používateľa embeduje používateľa
- Markdown pri odpovedi zobrazuje náhľad na odpoveď
- Markdown pri ukázaní otázky spracováva text v Markdowne
- Markdown pri odpovedi referencuje otázku a používateľa
- Markdown pri komentároch renderuje iba odkazy a spomenutia používateľov.

Jednotkovými testami sme tiež otestovali knižnicu `Redcurtain`, ktorá je obalovačom nad knižnicou `Redcarpet` pre generovanie textu v Markdowne.

16 Modul notifikácií a sledovania

Modul notifikácií predstavuje jednoduchý spôsob informovania používateľov o aktuálnom dianí v ich otázkach. Okrem prehľadných notifikácií modul umožňuje si vybrať, ktoré oblasti systému chce používateľ sledovať.

16.1 Analýza

Často sa používatelia prihlasujú do systému náhodne. Niekedy sa do systému neprihlásia aj niekoľko dní. Napriek tomu by chceli mať prehľad čo sa v ich otázke udialo bez toho aby svoju otázku museli vyhľadávať v zozname otázok a zisťovať zmeny sami. Typickým scenárom je, že používatelia chcú vedieť či niekto neodpovedal na ich otázku alebo komentoval ich odpoveď. Práve na tento účel slúži modul notifikácií.

Niektorí používatelia, hlavne učitelia, chcú mať prehľad o všetkých otázkach, na ktoré by boli schopní odpovedať. Preto okrem vlastnej otázky, prípadne otázky, na ktorú používateľ poskytol odpoveď, by mali mať používatelia možnosť sledovať aj niektoré iné elementy okrem otázok. Identifikovali sme preto tieto elementy, ktoré používatelia sú schopní sledovať:

- Kategória
- Značka (angl. tag)
- Otázka

Samotné otázky môžu byť sledované dvoma spôsobmi – explicitne a implicitne. Explicitný spôsob sledovania znamená, že používateľ bude informovaný o dianí v otázke až po tom čo sám o to požiada. Implicitný spôsob sledovania spočíva v tom, že používateľ po interakcii s otázkou začne byť informovaný o dianí v otázke. Rozhodli sme sa, že sledovanie by malo byť aktivované po týchto interakciách používateľa s otázkou:

- Vloženie novej otázky
- Vloženie novej odpovede (odpovedanie na otázku v systéme)
- Vloženie komentára (komentovanie otázky alebo niektorej jej odpovede)

Vzhľadom na množstvo akcií, ktoré je možné vykonať v jednotlivých elementoch, sme definovali tie najhlavnejšie aktivity, o ktorých by mal byť používateľ informovaný prostredníctvom notifikácií.

- Vloženie novej otázky (iba v prípade sledovania kategórie alebo značky, do ktorej je otázka zaradená).

- Vloženie novej odpovede.
- Vloženie nového komentára (pre otázku alebo odpoveď).
- Úprava otázky.
- Úprava odpovede.
- Úprava komentára.
- Zmazanie otázky.
- Zmazanie odpovede.
- Zmazanie komentára.
- Vyber najlepšej odpovede.

Okrem uvedených aktivít modul notifikácií upozorní používateľa aj v prípade, že ho niekto spomenul (označil) v otázke, odpovedi alebo komentári. Pritom tento element nemusí sledovať a tým pádom podporujeme formu „preposielania“ otázok ďalším používateľom.

16.2 Návrh

Modul notifikácií a sledovania predstavuje zoznam jednotlivých aktivít, o ktorých je používateľ informovaný. Zobrazenia jednotlivých notifikácií sú riešené dvoma spôsobmi. Jedným je jednoduché zobrazenie vysúvacieho okna s notifikáciou. Druhým spôsobom je zobrazit' si notifikácie ako zoznam udalostí.

Pri druhom spôsobe zobrazenia notifikácií sa nachádza aj zoznam sledovaných elementov. Samotné explicitné sledovanie je navrhnuté pomocou jednoduchej meniacej sa ikony. Pre zmenu ikony neobnovujeme celú stránku ale využívame JavaScript volania.

Pri implementácii modulu notifikácií sme sa rozhodli využiť návrhový vzor Publish–subscribe.

16.3 Implementácia

Sledovanie

Samotné nastavenie sledovania pre objekt sa vykonáva dvoma spôsobmi. Jedným je automatické nastavenie sledovania potom čo používateľ vykoná určitú akciu s otázkou. Toto nastavenie je realizované v `AnswersController`, `QuestionsController`, `CommentsController` a `EvaluationsController` zavolaním funkcie `register_watching_for`, ktorá je

implementovaná v module Watchings.

Druhým spôsobom pre sledovanie jednotlivých elementov je možnosť používateľov využiť ikonu oka, pomocou ktorej zapnú alebo vypnú používateľa sledovanie elementu. Samotná implementácia tejto ikony do používateľského rozhrania je realizovaná pomocou modulu watchables. Zmena typu ikony prebieha pomocou JavaScript volania. `$(document).ready ->`

```
$('#<%= @model %>-<%= @watchable.id %>-watching').replaceWith("<%=  
escape_javascript render('watchables/watching', model: @model, watchable:  
@watchable) %>")
```

```
fixes()
```

Akcia kliknutia na ikonu spôsobí spustenie funkcie `toggle_watching_by`, ktorá je implementovaná v module `Watchable` určenom pre modely. Tento modul je obsiahnutý v `Category`, `Tag` a `Question` modeli.

Pre jednoduchší prehľad sledovaných elementov bol implementovaný aj manažment sledovania, kde sú sledované objekty rozdelené podľa jednotlivých typov elementov na otázky, kategórie a značky. Tento manažment umožňuje zrušiť jednotlivé sledovania ako aj zrušiť všetky sledovania naraz.

Notifikácie

Zobrazenie samotných notifikácií je implementované tak, že po prihlásení alebo po vzniknutí novej notifikácie sa táto notifikácia objaví vo vrchnej časti obrazovky vo forme vysúvacieho okna, na ktoré je možné kliknúť. Po kliknutí na notifikáciu je používateľ automaticky presmerovaný na konkrétny element, ktorého sa notifikácia týkala a notifikácia je označená ako prečítaná.

Pre notifikácie bol taktiež implementovaný manažment notifikácií, kde používateľ má možnosť vidieť všetky jeho notifikácie. Tieto notifikácie sú postranným menu rozdelené na nové; teda tie, ktoré používateľ ešte neprezrel a všetky.

Samotný modul notifikácií predstavuje abstraktný modul založený na Publish–subscribe vzore. Na základe toho môžeme architektúru rozširovať o najrôznejšie notifikátory (systémové, mailové atď.).

Celkovo modul notifikácií dokáže registrovať zmeny nad všetkými modelmi - `resource`. Momentálne registrujeme notifikácie nad všetkými `resource`, ktoré používateľ svojou činnosťou môže vytvoriť. Medzi takéto `resource` patrí napríklad `View`, `Vote`, atď. Avšak pre potreby zobrazenia notifikácií je v modeli `Notification` použitý `default_scope`, ktorý túto množinu zdrojov notifikácií obmedzí len na tieto zdroje `Answer`, `Comment`, `Evaluation`, `Favorite`, `Labeling`, `Question`. Zvyšné notifikácie sa ukladajú do databázy pre potreby výskumu.

Pre registráciu notifikácií je v module `Events` implementovaná funkcia `dispatch_event`, ktorú je možné zavolať nad ľubovoľným zdrojom notifikácií.

```
def dispatch_event(action, resource, initiator: current_user, **options)
  Events::Dispatcher.dispatch(action, initiator, resource, options)
end
```

16.4 Testovanie

Modul notifikácií je na úrovni akceptačných testov otestovaný pre nasledovné scenáre:

- Notifikovanie používateľov sledujúcich kategóriu o novej otázke.
- Notifikovanie používateľov sledujúcich značku (tag) o novej otázke.
- Notifikovanie používateľov o pribudnutí novej odpovede v otázke.
- Notifikovanie používateľov o pribudnutí nového komentára k otázke.
- Notifikovanie používateľov o pribudnutí nového komentára k odpovedi.

Sledovanie je pokryté nasledujúcimi akceptačnými testami:

- Nastavenie používateľa ako sledovateľa otázky, ktorú pridal.
- Nastavenie používateľa ako sledovateľa otázky po odkliknutí ikony sledovania.
- Nastavenie používateľa ako sledovateľa otázky po pridaní vlastnej odpovede k otázke.
- Nastavenie používateľa ako sledovateľa otázky po pridaní vlastného komentáru k otázke.
- Nastavenie používateľa ako sledovateľa otázky po pridaní vlastného komentáru k odpovedi.
- Nastavenie používateľa ako sledovateľa značky (tagu) po odkliknutí ikony sledovania.
- Nastavenie používateľa ako sledovateľa kategórie po odkliknutí ikony sledovania.

Jednotkovými testami sú otestované funkcie modelu `Notification`, ktoré zaručujú správne vytvorenie notifikácií. Taktiež jednotkovými testami sú pokryté aj funkcie modulu `Watchable`.

17 Celkový pohľad

V našom prototypu webového komunitného systému otázok a odpovedí identifikujeme nasledovné funkcionality

- Autorizácia – V systéme existuje viac druhov používateľov, preto boli vytvorené rôzne právomoci pre rôzne skupiny.
- Lokalizácia – Náš systém bude obsahovať dve jazykové verzie. Implementované sú funkcie pre preklady do iného jazyku. Zatiaľ sú vytvorené len slovenské preklady.
- Prihlásenie – Do nášho systému je možné sa registrovať priamo ako nový používateľ alebo použiť prihlasovacie údaje z IS.
- Profil používateľa – Po úspešnom prihlásení sa do nášho systému si môže používateľ zobrazit' svoj profil. Okrem zobrazenia môže upraviť aj jednotlivé položky ako email alebo prezývku. Niektoré zmeny nie sú povolené za základe skupiny do akej používateľ patrí. Môže si nastavovať úroveň súkromia alebo pridávať odkazy na svoj profil na sociálnych sieťach.
- Vloženie novej otázky – Implementované je aj pridávanie otázok spolu s kategóriami a značkami.
- Zaznamenávanie udalostí – Nami vytvorený systém dokáže zaznamenávať aktivity používateľa, na základe ktorých bude možné vytvárať štatistiky alebo odporúčanie.
- Zobrazenie otázky – Otázky sa zobrazujú v zozname zoradené podľa dátumu ich pridania. Spolu s nimi sa zobrazia aj značky, kategórie, dátum vytvorenia a autor otázky. Po kliknutí na konkrétnu otázku sa zobrazí celá otázka aj s odpoveďami.
- vkladanie komentárov pre otázku – Používateľ môže okrem odpovede na otázku pridávať aj komentár k odpovedi.
- Hlasovanie – Pre každú otázku aj odpoveď je možnosť hlasovať pre každého používateľa a tým vyzdvihnúť odpovede alebo otázky.
- Označenie odpovede ako najlepšej - Autor otázky môže jednu z odpovedí označiť ako najlepšiu (viď 11 Modul administrácia, sli.do, označovanie správnych odpovedí, štatistiky).
- Rola používateľa – V našom systéme sa predpokladá viac druhov používateľov. Zatiaľ identifikovaní sú študent a učiteľ. Každý z nich má iné práva (viď 11 Modul administrácia, sli.do, označovanie správnych odpovedí, štatistiky).

- **Oblíbené otázky** – V systéme je možné pre každého používateľa vytvoriť si vlastný zoznam obľúbených otázok, ku ktorým sa vie jednoducho a rýchlo dostať.
- **Štatistiky otázok** – V zozname otázok sa pri každej otázke zobrazuje aj počet videní otázky, počet odpovedí a počet komentárov (viď 11 Modul administrácia, sli.do, označovanie správnych odpovedí, štatistiky).
- **Filtre** – Pre rýchle vyhľadávanie otázok slúži filter, ktorý sa nachádza v zozname otázok. Kliknutím na značku alebo dopísaním značky priamo do poľa sa automaticky zobrazia len otázky vyhovujúce kritériám (viď 13 Modul filtrovanie a vyhľadávanie).
- **Aktivita** – Systém zaznamenáva vybranú aktivitu systému vo vhodnom formáte. Táto aktivita môže byť podľa jej zdroja používateľom sledovaná (viď 10 Modul aktivita a nasledovanie).
- **Nasledovanie** – Používateľ si môže nastaviť že bude sledovať iného používateľa. Začnú mu potom chodiť notifikácie o jeho akciách ktoré si potom môže pozrieť v zázname aktivít nasledovaných používateľov (viď 10 Modul aktivita a nasledovanie).
- **Vyhľadávanie** – Je vytvorené vyhľadávanie v otázkach, odpovediach, komentároch, kategóriách, značkách a používateľoch pomocou fulltextových dopytov (viď 13 Modul filtrovanie a vyhľadávanie).
- **sli.do** – Systém spolupracuje s externým systémom sli.do, ktorý dopĺňa množinu otázok (viď 11 Modul administrácia, sli.do, označovanie správnych odpovedí, štatistiky).
- **Úvodná strana** – Prvá stránka s ktorou používateľ prichádza do styku je úvodná strana systému. Na nej sa nachádzajú dôležité odkazy a zobrazenie niektorých otázok (viď 14 Modul zoznamy, zmeny v systéme, úvodná strana).
- **Zoznamy** – Systém obsahuje niekoľko zoznamov nad používateľmi, otázkami, kategóriami, zoznamom zmien, značkami a aktivitami (viď 14 Modul zoznamy, zmeny v systéme, úvodná strana).
- **Markdown** – Pre zlepšenie skúsenosti používateľa so systémom je implementovaný Markdown (jazyk pre zápis štruktúrovaného textu) s podporou pre Gemojis (emotikony; viď 15 Modul markdown, pomoc v systéme).
- **Pomoc v systéme** – Pre používateľa je vytvorená pomoc v systéme. Táto pomoc je koncipovaná ako množina scenárov, ktoré môžu pri používaní systému nastať, spolu s krokmi riešenia (viď 15 Modul markdown, pomoc v systéme).
- **Notifikácie** – Notifikácie predstavujú jednoduchý spôsob informovania používateľov o aktuálnom dianí v ich otázkach (viď 16 Modul notifikácií a sledovania).

- Sledovanie – Sledovanie predstavuje špecifikáciu zaujímavých elementov pre používateľa, o ktorých chce byť informovaný, ak v nich nastane zmena (viď 16 Modul notifikácií a sledovania).
- Úprava – Používateľ má možnosť upravovať ním vytvorené otázky, odpovede, komentáre a hodnotenie. Predchádzajúce verzie sa ukladajú (viď 12 Modul úprava, mazanie a anonymné otázky).
- Mazanie – Systém povoľuje používateľovi mazať jeho otázky, pokiaľ k nim neboli ešte pridané odpovede, komentáre a hodnotenie. Podobne môže mazať odpovede, ak k nim neboli pridané žiadne komentáre (viď 12 Modul úprava, mazanie a anonymné otázky).
- Anonymné otázky – Niektorí používatelia sa môžu cítiť lepšie, ak nie je vidno, že sa danú otázku pýtajú oni. Ak sa môžu spýtať anonymne, umožní to, aby sa spýtali aj na menej príjemné otázky (viď 12 Modul úprava, mazanie a anonymné otázky).
- Zoznam zmien – Pre informáciu o zmenách v systéme slúži zoznam zmien (viď 14 Modul zoznamy, zmeny v systéme, úvodná strana).

17.1 Architektúra

Z architektonického hľadiska ide o klasickú webovú aplikáciu založenú na návrhovom vzore MVC – modelu, pohľadu a kontrolóra (angl. model-view-controller). Na databázovej vrstve aplikácia používa relačnú databázu PostgreSQL 9, na aplikačnej vrstve sú to rámce Rails 4 (webový rámec MVC) a rámec Bootstrap 3 (rámec pre tvorbu používateľského rozhrania).

Okrem spomenutých rámcov je v projekte použité aj značné množstvo knižníc na rôznych úrovniach architektúry, napr. pri testovaní, asociovaní entít so značkami, autentifikácii používateľa alebo zápise štruktúrovaného textu.

18 Čo sme nestihli?

Vďaka dobre stanoveným cieľom náš tím dokázal, skoro všetko čo si určil na začiatku samotného projektu. Nepodarila sa nám však implementácia odznakov. Odznaky mali slúžiť ako motivácia na využívanie nášho systému. Boli by pridelené jednotlivým používateľom na základe ich aktivity.

19 Čo sme sa naučili?

Práca na tímovom projekte mala pozitívny dopad na každého člena. Naše schopnosti sa posunuli ďalej hlavne v rámci diskusie na tímových stretnutiach, kde sa pomocou diskusie o funkcionalite systému dospelo mnohokrát k lepšiemu riešeniu ako na riešenie navrhnuté jednotlivcom. V rámci tímového projektu sme si priblížili aké je to pracovať so skupinou ľudí. Uvedomili sme si, že pri práci v kolektíve sú nutné ústupky a niekedy podriadenie sa väčšine. Taktiež sme si zlepšili komunikačné schopnosti či už medzi sebou ako členmi tímu alebo aj komunikáciu s vedúcim projektu.

Pre viacerých členov tímu znamenala práca na tomto projekte zoznámenie sa s novými technológiami, ktoré si rýchlo osvojili. Počas práci na projekte sme si sa naučili základné ale aj pokročilejšie techniky agilného vývoja. Veríme, že tieto poznatky budeme v budúcnosti aktívne využívať.

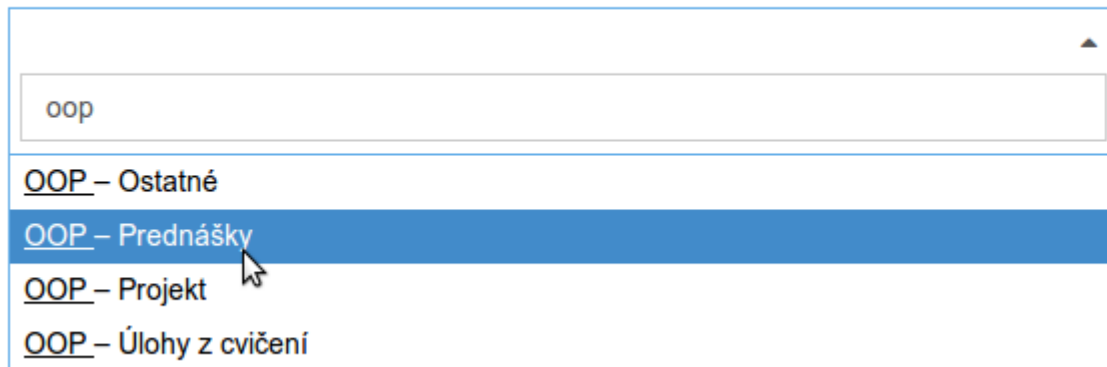
A Používateľská príručka

Náš projekt má za úlohu uľahčiť komunikáciu študentov medzi sebou a medzi učiteľmi a žiakmi. Systém je vytvorený ako komunitný systém otázok a odpovedí, čiže nie je to fórum, ale systém fungujúci na spôsobe otázok a odpovedí. Systém je webová aplikácia, takže jeho spustenie je zadanie stránky vo webovom prehliadači. Následne je nutné sa registrovať, alebo sa prihlásiť pomocou AIS prihlasovacích údajov.

1 Pridanie novej otázky

Pred každým úkonom ktorý chce používateľ vykonať v systéme je potrebné aby bol prihlásený. Po kliknutí na tlačidlo opýtať sa otázku sa otvorí formulár, pomocou ktorého je možné pridať novú otázku do systému.

Kategória




oop

- OOP – Ostatné
- OOP – Prednášky
- OOP – Projekt
- OOP – Úlohy z cvičení

Obr. A-1: Kategória pri pridávaní novej otázky

Prvou položkou, ktoré treba vo formulári novej otázky je kategória. Kategórie slúžia na rámcové zaradenie otázok do okruhov, ktoré už môžu sledovať jednotliví odborníci v daných oblastiach.

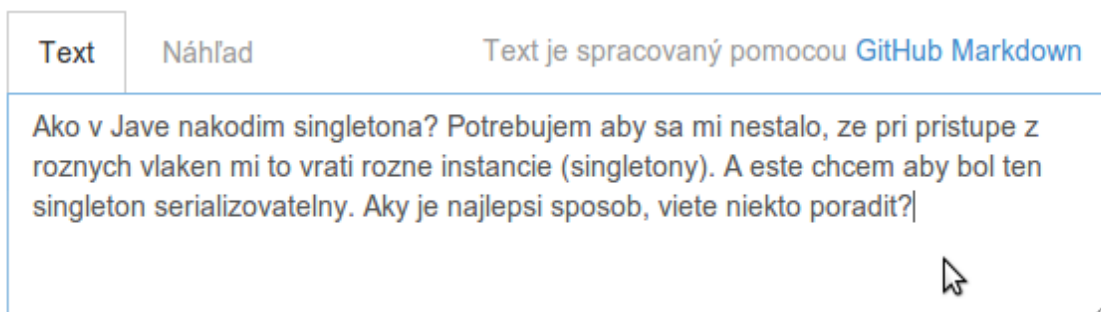
Nadpis



Singleton v Jave

Obr. A-2: Nadpis pri pridávaní otázky

Nadpis otázky má stručne a jasne vystihnúť podstatu toho, na čo sa ide používateľ pýtať. Nadpis by mal byť koncipovaný tak, aby zaujal prípadného odpovedajúceho.



Obr. A-3: Text pri pridávaní otázky

Otázku je potrebné položiť priamo, dobre vysvetliť kontext. Dobrá otázka je taká, ktorá sa priamo pýta na daný problém a obsahuje tiež zhrnutie toho, čo sa používateľ o danej problematike doteraz dozvedel. Pri písaní je možné použiť Markdown pre lepšie štrukturalizovanie textu, alebo Gemojis na vyjadrenie emócií.



Obr. A-4: Značky pri pridávaní otázky

Pre priradenie novej otázky vhodným používateľom (ktorí pravdepodobne budú vedieť odpovedať) je potrebné jej priradiť správne značky. Toto je najdôležitejší krok pre správne zaradenie otázky. Ak používateľ začne značku písať, automaticky sa mu dopĺňajú existujúce značky. Je potrebné, aby sa vyberalo čo najviac už existujúcich značiek (napr. rails vs. ruby-on-rails). Správne značky sú také, ktoré čo najlepšie dopĺňajú význam a kontext otázky. Podľa značiek totižto ostatní používatelia vyhľadávajú otázky ktoré by ich mohli zaujímať, resp. na ktoré by mohli odpovedať.

Aby obsah a vedomosti, ktoré sa pri otázke nahromadili boli znovupoužiteľné, resp. študenti čo hľadajú odpoveď na rovnaký problém vedeli že už bol problém vyriešený, je potrebné po získaní odpovedí na otázku označiť tú najlepšiu. Takto sa v systéme budú postupne zbierať vyriešené problémy.

Pokiaľ sa od minulého roka nezmenili podmienky, tak odpoveď je nie. V prvej časti projektu mas MySQL príkazy písať ty. V druhej mas použiť O/R mapovac. Tá trieda Active Record sice poskytuje niektoré podobné vymoženosti, ale nie je to ešte tak celkom O/R. Dajú sa však stiahnuť knižnice, napr. DataMapper ORM, s ktorou druhú časť zvladnes.

Pridal [piko](#) 28. Február 2014

[Pridať komentár](#)

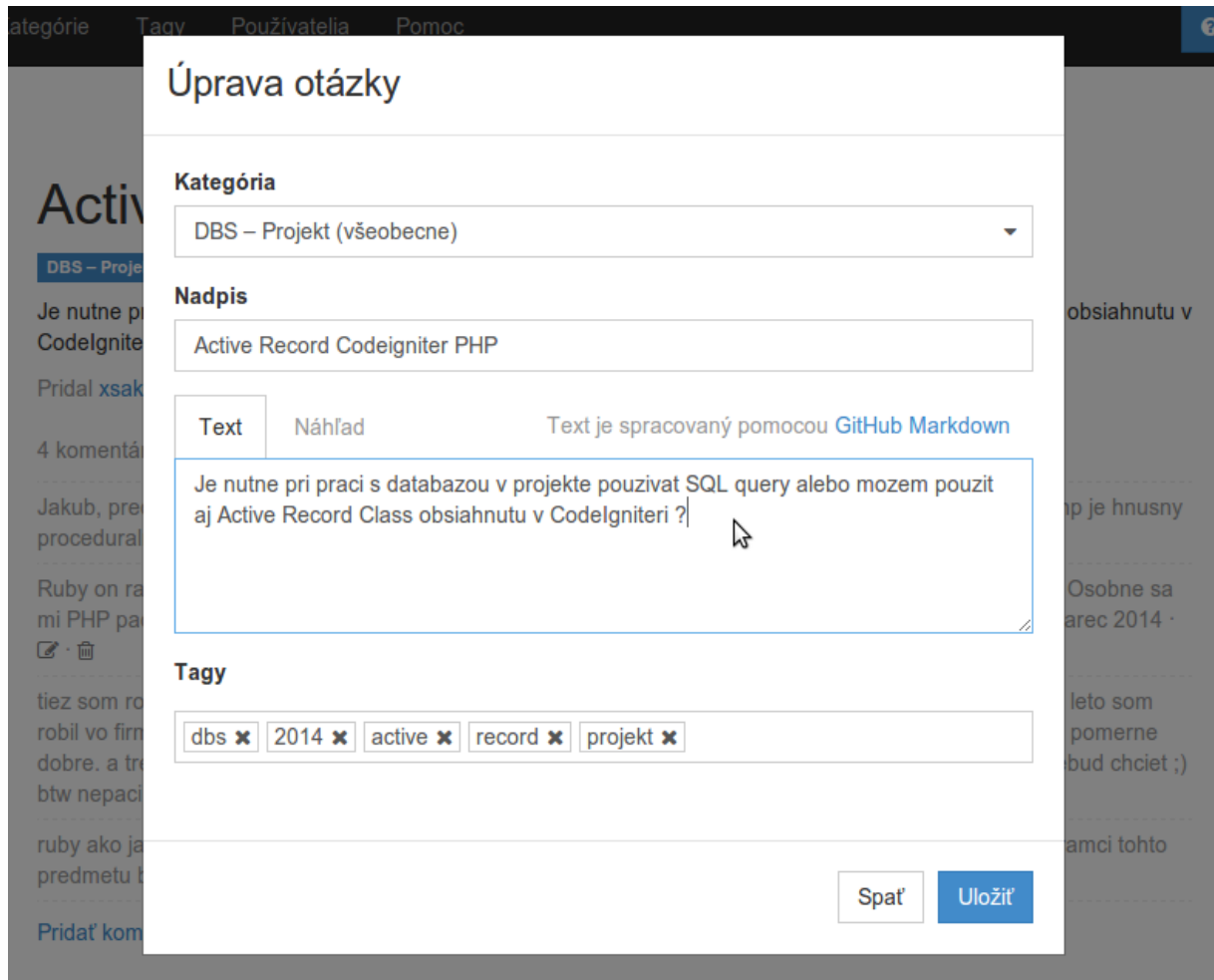
0

Označiť ako najlepšiu

Obr. A-5: Označenie najlepšej odpovede

2 Úprava

Ak sa používateľ pomýlil a chce upraviť svoju otázku, je potrebné kliknúť na ikonu upraviť. Úprava má slúžiť najmä na opravenie preklepov a iných chýb, prípadné doplnenie značiek, či nahradenie nevhodných. Ak používateľ doplní nový text, je vhodné, aby na konci starého textu dopísal slovo edit: a za toto kľúčové slovo doplnil čo chýba. Takto bude zrejmé, čo je v texte nové.



Obr. A-6: Úprava odpovede

3 Hľadanie informácií

Komunitný systém otázok a odpovedí slúži aj ako miesto na vzdelávanie. Vo vyriešených otázkach môže používateľ nájsť množstvo vedomostí a odpovedí na problémy, ktoré rieši.

V systéme je implementované vyhľadávanie podľa značiek. Nachádza sa navrchu zoznamu všetkých otázok. Je možné ho použiť dvoma spôsobmi:

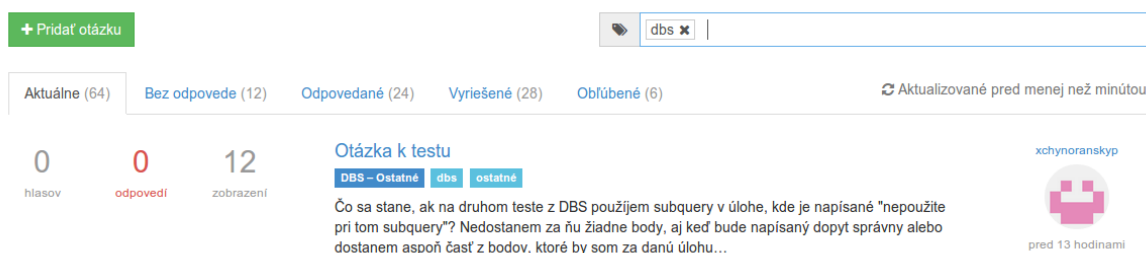
- používateľ začne písať nejakú značku, existujúce značky sa automaticky dopĺňajú a vyberie si tie, ktoré ho zaujímajú,
- klikne na značku alebo kategóriu pri otázkach a dané značky sa automaticky doplnia do vyhľadávania.



Obr. A-7: Vyhľadávanie podľa značiek

V zozname otázok je možné otázky filtrovať viacerými možnosťami:

- aktuálne – všetky otázky, zoradené podľa času poslednej interakcie s otázkou,
- bez odpovede – otázky na ktoré ešte nikto neodpovedal,
- odpovedané – otázky ktoré síce majú odpoveď, ale ešte nebola zvolená najlepšia,
- vyriešené – otázky, ktoré už sú uzavreté – bola zvolená najlepšia odpoveď,
- obľúbené – otázky, ktoré boli označené používateľmi ako obľúbené.



Obr. A-8: Filtrovanie

V systéme môže používateľ sledovať otázky, značky a kategórie ktoré ho zaujímajú. Následne mu budú chodiť notifikácie o dianí v nich. Aby niečo sledoval, je potrebné kliknúť na ikonu sledovania.

Ak používateľa zaujíma aktivita niektorého z používateľov, môže ho nasledovať. Stačí, ak klikne na ikonu nasledovania.

Používateľ má možnosť vyjadriť svoj subjektívny názor na otázku alebo odpoveď pomocou hlasovania. Hlasovanie ovplyvňuje aj ich pozíciu v zozname otázok alebo odpovedí (horšie sú nižšie, lepšie sú zase vyššie v zozname a je teda jednoduchšie sa k nim dostať).

Otázky si používateľ môže označiť aj ako obľúbené. Tieto otázky môže sledovať (chodia mu o nich notifikácie). Vidí tiež počet ľudí, ktorí majú danú otázku ako obľúbenú, je to teda tiež jeden

z ukazovateľov toho, čo je práve aktuálne.

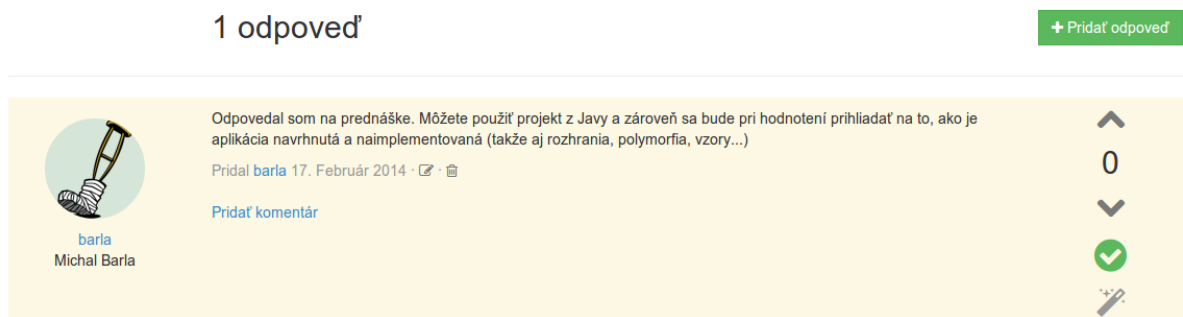


The screenshot shows a forum post by user 'xkrchlik' (Jakub Krchlik) titled 'Lispworks - nefunguje príklad'. The post content includes a code snippet: `(REST '(7 2 14))` and an error message: `Error: In * of ((2 14)) arguments should be of type NUMBER.` The post has one reply and a 'Pridať medzi obľúbené' button. The user's profile picture is a purple 'I' logo.

Obr. A-9: Oblúbenie

Zapojenie učiteľa

Učiteľ v Askalote plní špeciálnu rolu tútora. Aby bolo jasné, ktorý obsah je pridaný učiteľom, sú tieto komentáre a odpovede farebne zvýraznené. Učiteľ tiež môže ohodnocovať otázky a odpovede.



The screenshot shows a forum reply by user 'barla' (Michal Barla) to the post in Figure A-9. The reply text is: 'Odpovedal som na prednáške. Môžete použiť projekt z Javy a zároveň sa bude pri hodnotení prihliadať na to, ako je aplikácia navrhnutá a naimplementovaná (takže aj rozhrania, polymorfia, vzory...)'. The reply has 0 votes and a 'Pridať komentár' button. The user's profile picture is a drawing of a hand holding a pencil.

Obr. A-10: Zapojenie učiteľa

4 Odpovedanie na otázku

Veľmi dôležitým prvkom v systéme sú odpovede, ktoré sú pri detailnom zobrazení otázky zoradené pod ňou. Používateľ môže pridať novú odpoveď pomocou formulára na konci stránky s otázkou (pre rýchlejšiu navigáciu je možné kliknúť na Pridať odpoveď navrchu stránky).

Aby používateľ našiel ľahšie otázky, ktoré sú relevantné k jeho oblasti, môže použiť vyhľadávanie otázok podľa značiek. Taktiež si môže odfiltrovať ešte nezodpovedané, alebo nevyriešené otázky.

Pred pridaním odpovede je potrebné si najprv pozrieť už existujúce odpovede, aby sa neopakovalo, čo už niekto iný napísal. Ak je niektorá odpoveď podľa používateľa dobrá, je vhodné, ak za ňu zahlasuje. Ak je zlá, môže jej dať záporný hlas.

5 Úprava profilu

Profil poskytuje informácie o používateľovi. Je možné v ňom nastavovať viacero vecí.

Používateľ si môže nastaviť prezývku, e-mail na Gravatar (fotografia používateľa), doplniť informácie o sebe. Odporúča sa použiť Markdown.

Profil

Sociálne siete

Súkromie

Účet

Prezývka

jharinek

Gravatar e-mail

j.harinek@gmail.com

Krstné meno

Jozef

Príezvisko

Harinek

O mne Náhľad

Uložiť

Obr. A-11: Úprava profilu

K profilu používateľ je možné nalinkovať jeho profily z iných sociálnych sietí zadaním ich adresy do adekvátnych polí v záložke Sociálne siete.

<ul style="list-style-type: none"> Profil <li style="background-color: #0070C0; color: white; padding: 2px;">Sociálne siete Súkromie Účet 	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; width: 30px;">f</td> <td>http://facebook.com/username</td> </tr> <tr> <td style="text-align: center;">t</td> <td>http://twitter.com/username</td> </tr> <tr> <td style="text-align: center;">g+</td> <td>http://plus.google.com/userid</td> </tr> <tr> <td style="text-align: center;">in</td> <td>http://linkedin.com/in/username</td> </tr> <tr> <td style="text-align: center;"></td> <td>http://stackoverflow.com/users/userid</td> </tr> <tr> <td style="text-align: center;"></td> <td>http://github.com/username</td> </tr> <tr> <td style="text-align: center;"></td> <td>http://bitbucket.org/username</td> </tr> </table> <div style="margin-top: 5px;"> Uložiť Viac </div>	f	http://facebook.com/username	t	http://twitter.com/username	g+	http://plus.google.com/userid	in	http://linkedin.com/in/username		http://stackoverflow.com/users/userid		http://github.com/username		http://bitbucket.org/username
f	http://facebook.com/username														
t	http://twitter.com/username														
g+	http://plus.google.com/userid														
in	http://linkedin.com/in/username														
	http://stackoverflow.com/users/userid														
	http://github.com/username														
	http://bitbucket.org/username														

Obr. A-12: Prepojenie so sociálnymi sieťami

Pokiaľ si používateľ neželá zobrazovať svoje meno alebo e-mail, má možnosť ich skryť v záložke Súkromie.

<ul style="list-style-type: none"> Profil Sociálne siete <li style="background-color: #0070C0; color: white; padding: 2px;">Súkromie Účet 	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; width: fit-content;">Skryť meno ostatným používateľom</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px; width: fit-content;">Skryť e-mail ostatným používateľom ✓</div> <div style="margin-top: 5px;"> Uložiť </div>
---	--

Obr. A-13: Nastavenia súkromia

V záložke Účet môže používateľ zmeniť svoj e-mail asociovaný s účtom, a takisto svoje heslo.

The image shows a user interface for account settings. On the left, there is a vertical menu with four items: 'Profil', 'Sociálne siete', 'Súkromie', and 'Účet'. The 'Účet' item is highlighted with a blue background. To the right of the menu, the 'Účet' section contains the following elements:

- E-mail**: A text input field.
- Heslo** (ponechajte prázdne pokiaľ nechcete zmeniť): A text input field with a light gray background.
- Potvrdenie hesla**: A text input field with a light gray background.
- Aktuálne heslo** (pre potvrdenie potrebujeme Vaše aktuálne heslo): A text input field.
- Uložiť**: A green button.

Obr. A-14: Nastavenia účtu

6 Markdown

V systéme je možnosť použiť Markdown v texte otázok, odpovedí aj komentárov. Implementácia je kompatibilná s GitHub Markdown. Markdown pre komentáre je menšia množina značiek, je možné napr. pridávať značky pre linky a značky pre zdrojový kód v jednom riadku.

Na otázky sa dá odkazovať po napísaní # a identifikátora otázky (napr. #1 odkazuje na otázku s id = 1). Po napísaní identifikátora sa automaticky doplní názov otázky. Ponúkané je aj automatické doplnenie podľa názvu otázky hneď po napísaní #.

V Markdown textoch je možné odkazovať sa na iných používateľov. Po napísaní @ je potrebné začať písať prezývku odkazovaného používateľa. Pomocou je automatické dopĺňanie.

V Markdown textoch je možné pridávať aj ikony Gemoji. Po napísaní : sa zobrazí ponuka Gemojis, vyhľadáva sa v nich podľa názvu.

B Inštalačná príručka

Pre správne fungovanie aplikácie je potrebné mať v systéme nainštalované:

- Ruby 2.1
- Rails 4.1
- PostgreSQL 9.3
- Elasticsearch 1.1

Zdrojový kód projektu je prístupný v repozitári v sieti GitHub. Je teda potrebné najprv si naklonovať tento repozitár na svoj lokálny stroj. To sa vykoná pomocou príkazu:

```
git clone git@github.com:teamnaruby/askalot.git
```

Potom je potrebné navigovať sa do adresára v ktorom sa nachádza lokálna kópia projektu – cd askalot. Samotná inštalácia sa spustí príkazom `bundle install`. Týmto sa nainštalujú všetky potrebné knižnice.

Následne je potrebné skopírovať a upraviť konfiguračné súbory:

```
cp config/configuration.{yaml.example, yaml}
cp config/database.{yaml.example, yaml}
cp config/newrelic.{yaml.example, yaml}
```

Po nastavení aplikácie je potrebné vytvoriť databázu:

```
RAILS_ENV=development rake db:create
RAILS_ENV=development rake db:migrate
```

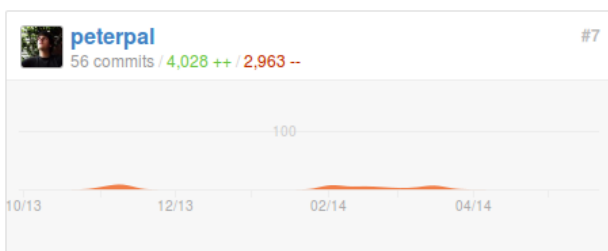
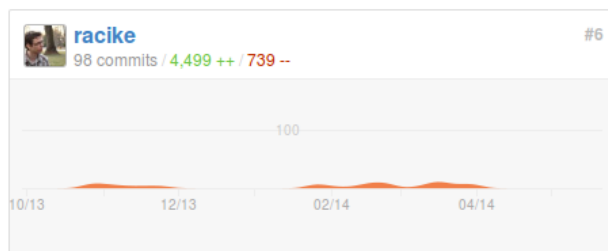
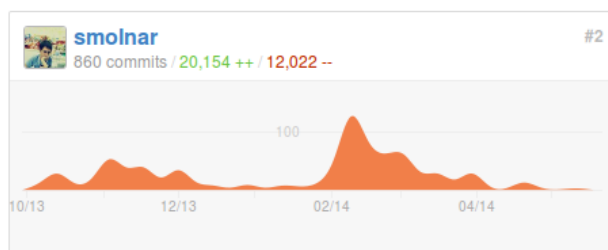
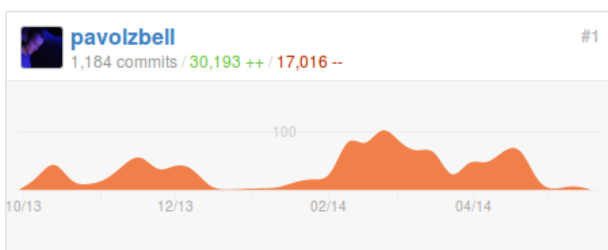
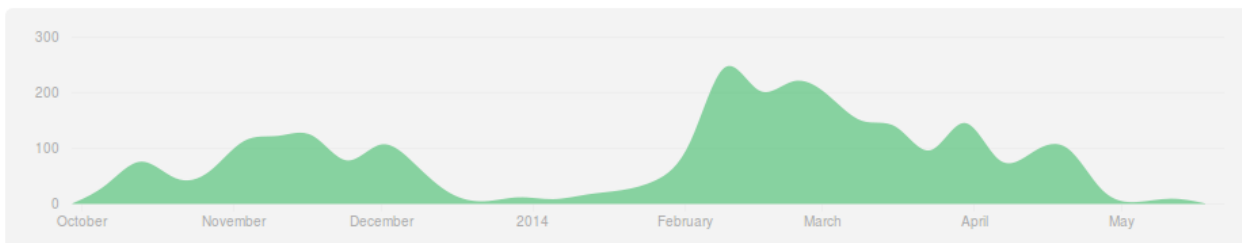
C Štatistiky nad repozitármi projektu

Vybrané štatistiky nad repozitármi projektu *askalot* (zdrojový kód aplikácie) a *askalot-assets* (technické a iné dokumenty) ku dňu 22. máj 2014.

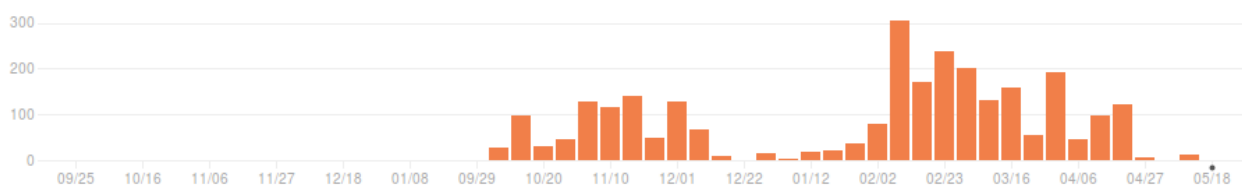
September 29th 2013 - May 22nd 2014

Contributions to beta, excluding merge commits

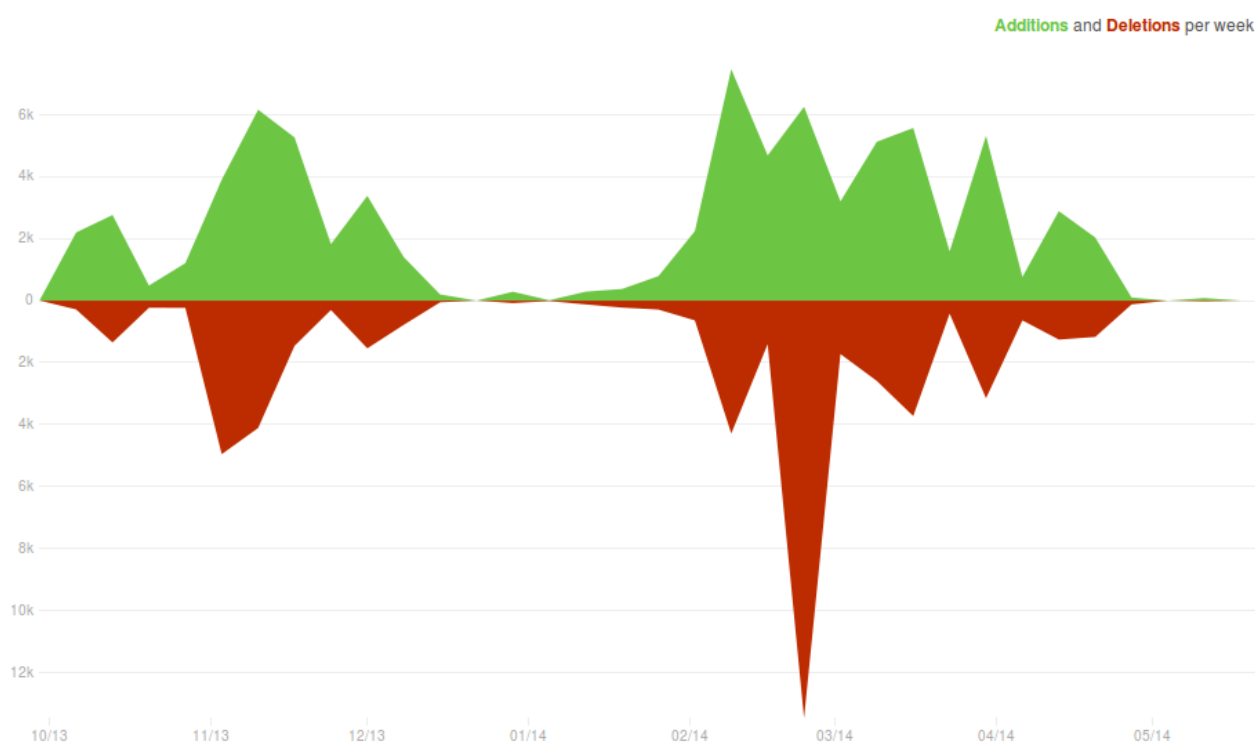
Contribution type: **Commits** ▾



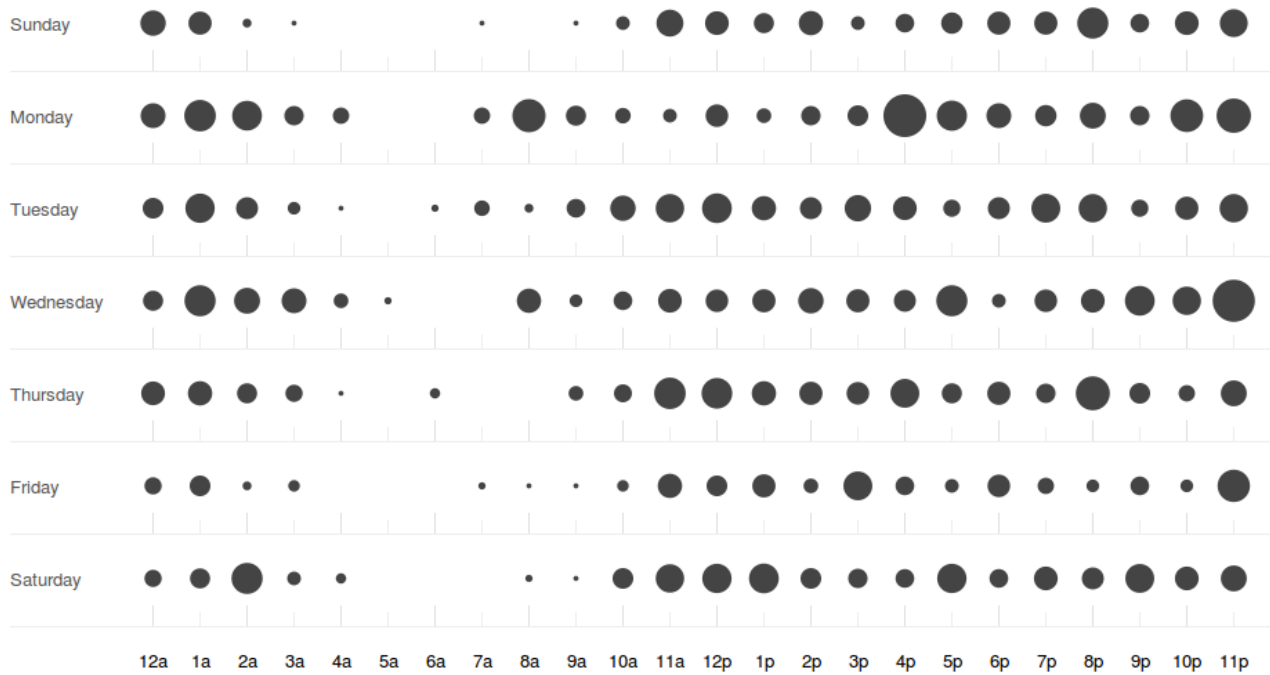
Obr. C-1: Štatistika prispievateľov pre repozitár *askalot*



Obr. C-2: Odobrdania podľa týždňov pre repozitár askalot



Obr. C-3: Zmeny zdrojového kódu pre repozitár askalot

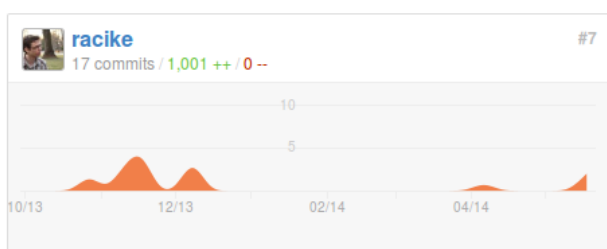
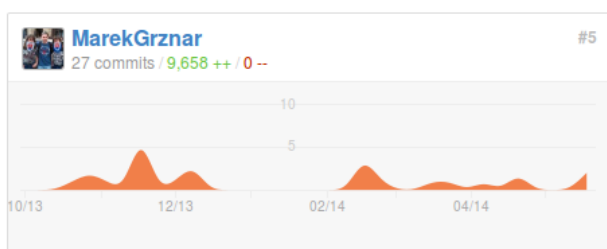
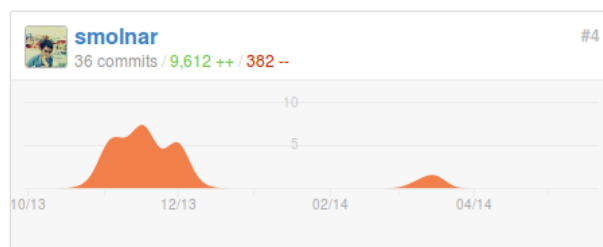
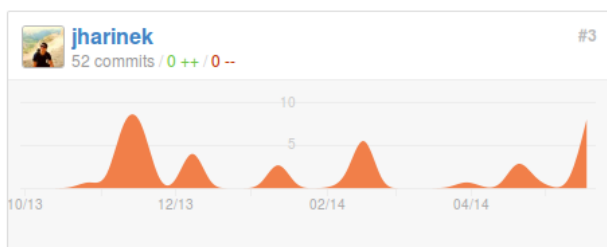
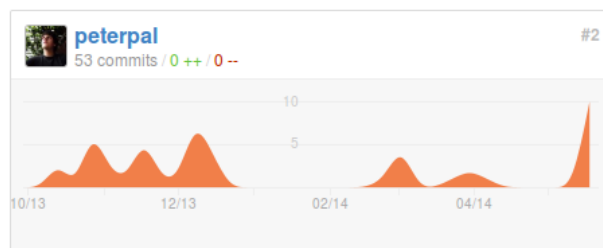
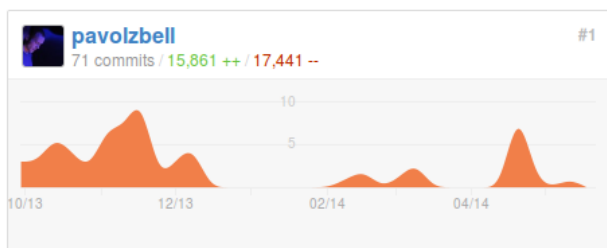
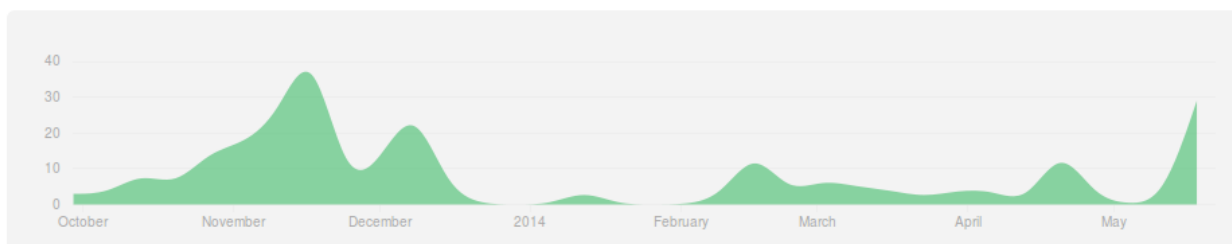


Obr. C-4: Frekvencia odovzdaní podľa hodín pre repozitár askalot

September 29th 2013 - May 22nd 2014

Contributions to master, excluding merge commits

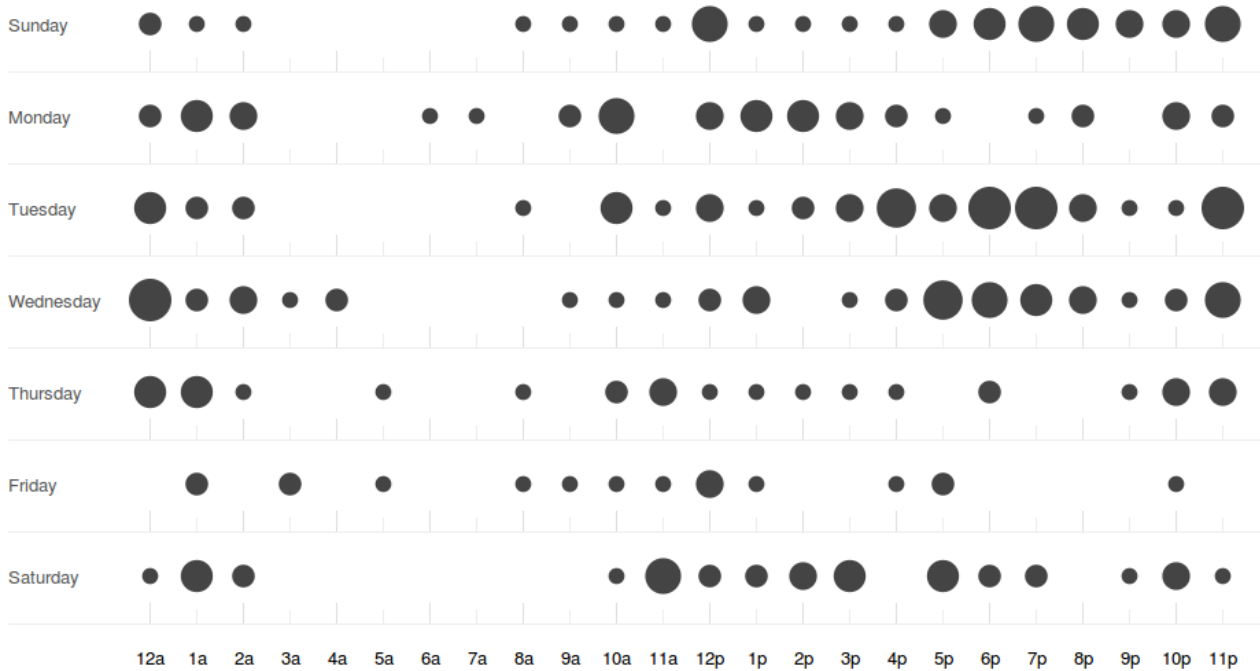
Contribution type: **Commits**



Obr. C-5: Štatistika prispievateľov pre repozitár askalot-assets



Obr. C-6: Odovzдания podľa týždňov pre repozitár askalot-assets



Obr. C-7: Frekvencia odovzdaní podľa hodín pre repozitár askalot-assets