

Askalot meets Harvard Courses at edX

[Askalot2edX]

Dokumentácia k tímovému projektu časť 1 (Inžinierske dielo)

Tím: číslo 6, AskEd
Vedúci tímu: Ing. Ivan Srba
Členovia tímu: Černák Martin, Gallay Ladislav, Hnilicová Eva, Huňa Adrián, Jandura Filip,
Žuffa Tibor
Akademický rok: 2015/2016
Autor: Černák Martin, Gallay Ladislav, Hnilicová Eva, Huňa Adrián, Jandura Filip,
Žuffa Tibor
Verzia číslo: 3
Dátum poslednej zmeny: 19.5.2016

Obsah

| | |
|---|----|
| Úvod | 3 |
| 1. Globálne ciele | 4 |
| 1.1. Ciele pre zimný semester | 4 |
| 1.2. Naplnenie cieľov za zimný semester | 4 |
| 1.3. Ciele pre letný semester | 5 |
| 1.4. Naplnenie cieľov za letný semester | 5 |
| 2. Celkový pohľad na systém | 6 |
| 2.1. Funkcionalita systému | 6 |
| 2.2. Architektúra | 10 |
| 2.3. Infraštruktúra | 10 |
| 2.4. Dátový model | 12 |
| 3. Moduly | 16 |
| 4. Príručky | 17 |
| 4.1. Inštalačná príručka | 17 |

Úvod

Tento dokument predstavuje ucelený pohľad na vývoj systému Askalot v rámci predmetu *Tímový projekt* v akademickom roku 2015/2016 tímu číslo 6. Zahŕňa ciele projektu pre jednotlivé semestre a celkový pohľad na štruktúru a funkcionálnosť systému.

Názov témy nášho projektu je Askalot2edX a jeho cieľom je integrácia existujúceho fakultného komunitného systému pre odpovedanie na otázky v komunitách, Askalotu, do vzdelávacieho systému edX. EdX je jedným zo systémov, ktoré sú označované ako poskytovatelia masívnych otvorených online kurzov (angl. Massive Open Online Courses, MOOC). V spolupráci s výskumníkmi z Harvard University je našou úlohou umožniť používanie Askalotu tisíciam študentov, ktorí sa registrujú do týchto kurzov.

Prvá kapitola dokumentu zahŕňa globálne ciele tímu pre dané obdobie, čo chce tím dosiahnuť a v čom chce uspieť. Druhá kapitola ponúka celkový pohľad na systém, zahŕňa opis architektúry a dátový model systému. V tretej kapitole uvádzame moduly systému, ktoré sme za uplynulé obdobie vytvorili. Vo štvrtej kapitole opisujeme naplnenie stanovených cieľov a na konci dokumentu sa nachádzajú príručky.

1. Globálne ciele

Pri plánovaní cieľov na letný semester sme vychádzali z dokončenej práce počas zimného semestra a následne naplánovali ďalšie vylepšenia systému.

1.1. Ciele pre zimný semester

Celkovo sme si naplánovali niekoľko krokov:

1. Zoznámenie sa s Askalotom.
 - Práca nášho tímu spočívala v úprave existujúceho riešenia. V tomto kroku sme sa preto zoznamovali s Askalotom po dizajnovej a implementačnej stránke. Hoci je Askalot voľne dostupný (angl. open source), väčšina tímu videla jeho zdrojový kód na začiatku semestra po prvý krát. Viacerí členovia navyše predtým neprogramovali v jazyku Ruby a ani nemali skúsenosť s frameworkom Ruby on Rails.
 - Cieľom tohto kroku bola príprava prostredia pre vývoj novej funkcionality a naprogramovanie drobných vlastností do systému, kde sme sa prakticky oboznámili so zdrojovým kódom a vývojovým procesom.
2. Analýza integrácie do systému edX.
 - Našou úlohou bolo zanalyzovať možnosti integrácie Askalotu a vybrať tú najvhodnejšiu. Súčasťou tohto cieľa bolo získanie čo najväčšieho množstva informácií o študentovi a kurzoch v edX systéme, a zároveň vhodne navrhnúť vloženie Askalotu do edX tak, aby pôsobil prirodzene ako súčasť stránky a plnohodnotne dokázal nahradiť existujúcu diskusiu.
3. Implementácia integrácie.
 - Tento cieľ sleduje finálnu implementáciu integrácie do systému Askalotu. Pri plnení tohto cieľa sme museli myslieť na správu zdrojového kódu a rozhodli sme sa zmeniť architektúru celého systému v prospech modulárnej architektúry.
 - Súčasťou tohto cieľa je aj redizajn Askalotu. Aktuálne používateľské rozhranie je síce moderné, no pre systém edX nevhodné, nakoľko v ňom bude Askalot zobrazený ako podprvok. Taktiež bolo potrebné upraviť a doplniť anglické preklady textov.

1.2. Naplnenie cieľov za zimný semester

Po dokončení úloh zimného semestra môžeme zhodnotiť, že sa nám vo veľkej miere podarilo naplniť ciele stanovené na začiatku semestra. Všetci členovia tímu sa úspešne zoznámili s Askalotom a zlepšili si svoje znalosti z oblasti vývoja Ruby on Rails aplikácií. Podarilo sa nám zanalyzovať rôzne možnosti integrácie Askalotu do systému edX. Na základe analýzy sme vybrali najvhodnejšie riešenie, ktoré nám zabezpečuje korektnú autentifikáciu používateľa. Zistili sme však, že toto riešenie nie je možné použiť pre globálny pohľad a preto sme ďalšou analýzou navrhli najvhodnejší spôsob odstránenia tohto problému.

Na záver sa nám úspešne podarilo integráciu implementovať, pričom okrem zabezpečenia komunikácie edX systému s Askalotom sme taktiež vytvorili pohľad, ktorý sa bude zobrazovať pri jednotlivých učebných materiáloch. Počas riešenia integrácie sme vyriešili problém zdieľania funkcionality medzi verziami pre univerzitu a pre edX navrhnutím a zrealizovaním novej modulárnej architektúry systému. Redizajn Askalotu pre potreby edX kurzov sa nám nepodarilo zrealizovať a túto úlohu si nechávame ako prácu na nasledujúci semester.

1.3. Ciele pre letný semester

Prioritou pre letný semester bolo dokončenie implementácie integrácie do systému edX a nasadenie na niektorý edX kurz. Druhým dôležitým bodom bolo pripravenie systému pre nasadenie na univerzite v Lugane, kde sa Askalot nasadzuje v rámci projektu SCOPES. Zároveň sme si naplánovali zlepšenia systému, ktoré však neboli nevyhnutné pre nasadenie na edX kurze. Hlavné ciele, ktoré sme si naplánovali sú:

1. Dokončenie implementácie integrácie.
 - Hoci sa nám po zimnom semestri podarilo pripraviť integráciu Askalotu do funkčného stavu, v systéme nám chýbala dôležitá funkcionálna vo forme izolovania obsahu jednotlivých kurzov. Druhým nedostatkom bola nemožnosť zdieľania obsahu medzi verziami kurzu, čo je jednou z vlastností, ktoré sme chceli implementovať. V tomto kroku, preto bolo za cieľ dokončiť implementáciu a nasadiť Askalot na nejakom edX kurze.
2. Nasadenie na univerzite v Lugane.
 - Tento cieľ obsahuje úpravu systému pre potreby univerzity v Lugane a nasadenie Askalotu pre túto univerzitu. Jedinou požiadavkou na zmenu systému je možnosť plne anonymnej registrácie používateľov.
3. Vylepšovanie funkcionality Askalotu.
 - Identifikovali sme viaceré menšie funkcionality, ktoré prispievajú ku skvalitneniu Askalotu a cieľom bolo ich implementovať. Identifikované funkcionality sú: opis kategórií, novinky na domovskej stránke, sledovanie kategórií na vyššej úrovni, notifikácie do edX rozhrania, možnosť vytvárať nie len otázky ale aj diskusie.

1.4. Naplnenie cieľov za letný semester

Po skončení letného semestra sa nám podarilo splniť všetky stanovené globálne ciele. V rámci letného semestra nás už nespomaľovala zmena architektúry a mohli sme sa plne sústrediť na rozširovanie funkcionality systému, nasadenie Askalotu na edX a vytvoriť tak stabilnú verziu pripravenú na používanie. Na záver sa nám ešte navyše podarilo obohatiť systém o predtým neplánovanú funkcionality osobnej nástenky (angl. dashboard), ktorá používateľom ponúka prehľadnú štatistiku posledných udalostí (pridaných otázok, odpovedí a komentárov, úpravy príspevkov a pod.).

2. Celkový pohľad na systém

2.1. Funkcionalita systému

V tejto podkapitole opisujeme systém Askalot vrátane funkcionality, ktorú implementoval predchádzajúci tím v akademickom roku 2013/2014. V prípade, že sme danú funkcionality implementovali my, tak je na to špeciálne upozornené.

Askalot je systém pre odpovedanie na otázky v komunitách (angl. Community Question Answering, CQA). Základnou funkcionality takýchto systémov je pýtať sa otázok, odpovedanie na ne, komentovanie otázok a odpovedí, hodnotenie otázok a odpovedí hlasovaním a nasledovanie aktivity v systéme. Obsah v systéme Askalot je organizovaný dvojúrovňovo - v prvom rade má každá otázka priradenú kategóriu a zároveň jej je možné priradiť niekoľko značiek. V rámci nášho tímového projektu sme prerobili systém kategórii do hierarchickej štruktúry.

Otázky v Askalote je možné sa pýtať na dvoch miestach. Ak je otázka položená globálne, vidia ju všetci používatelia systému. Otázky je možné sa opýtať aj anonymne a vtedy je identita používateľa skrytá. Druhou možnosťou je pýtať sa otázky v skupinách. Funkcionality skupín používame len v prostredí *university*. Skupiny je možné vytvárať buď vo verejnom móde alebo vytvárať súkromné skupiny pre študentov. Obsah skupín je založený na dokumentoch (napríklad prednášky), ku ktorým sú priradené otázky a odpovede.

Každý používateľ má vytvorený profil, ktorý zobrazuje vytvorený obsah a prehľad aktivity používateľa. Jednotlivých používateľov je možné nasledovať a filtrovať ich aktivitu v systéme. Aktivitu je možné sledovať aj pre kategórie a značky. Sledovaním kategórie na vyššej úrovni dostane používateľ notifikácie pre všetky kategórie na nižšej úrovni. Túto funkcionality sme implementovali počas letného semestra. Po pridaní novej otázky do sledovanej kategórie alebo značky používateľ následne dostane notifikáciu.

Systém aktivít a notifikácií je implementovaný návrhovým vzorom *observer* (pozn: zaužívaný názov návrhového vzoru, v slovenčine ako *pozorovateľ*). Každá akcia v systéme (pridanie otázky, odpovede, zobrazenie otázky, ...) notifikuje *pozorovateľov* (angl. *observer*) a tí podľa typu aktivity vyhodnotia svoje ďalšie aktivity. Typy *pozorovateľov* sú:

- aktivity - vytvorenie objektu aktivity,
- facebook - odoslanie notifikácie na spárovaný facebook používateľa,
- notifikácie - vytvorenie notifikácie v systéme pre sledované kategórie, značky, ...
- reputácia - úprava reputácie používateľa.

Notifikácie v systéme sa zobrazujú v hlavnom menu. V letnom semestri sme tieto notifikácie vylepšili a pridali ajaxové označenie notifikácie za prečítané.

Kategóriám v systéme je možné priradovať používateľov v rámci určitých rolí. Týmto spôsobom sú napr. určití študenti priradení ako učitelia pre kategórie *Študentský radca*. V prípade, že má kategória priradených učiteľov, tak sa pri jej názve zobrazuje ikona *university*. Danú funkcionality sme implementovali v zimnom semestri.

Hlavný zoznam otázok zobrazuje otázky zoradené podľa poslednej aktivity v nich (napr. pridanie komentára). Pri pohľade na otázku je okrem tradičných možností CQA systémov (odpovedanie, komentovanie) možné otázku archivovať v prípade, že nemá odpovede (dorobené v zimnom

semestri). Ak je používateľ učiteľom, tak môže hodnotiť otázku formou označenia *Veľmi zle*, *Zle*, *Dobre*, *Veľmi dobre*, *Výborne*. Rovnako môže hodnotiť aj odpovede. Ak učiteľ pridá otázku alebo odpoveď, tak je jeho odpoveď zvýraznená žltým pozadím. Funkcionalitou, ktorú sme dorobili v zimnom semestri je polozenie otázky v rovnakej kategórii, čo je možné po kliknutí na tlačidlo v spodnej časti obrazovky pred formulárom pre novú odpoveď.

Pri profilovom obrázku používateľa sa zobrazuje ikona reputácie. Používateľ môže patriť do jedného zo štyroch úrovní (angl. level) reputácie - zlatá, strieborná, bronzová, záporná. Reputáciu implementoval v rámci bakalárskej práce študent Adrián Huňa. Reputácia je založená na náročnosti otázky (ako dlho trvá otázku zodpovedať) a užitočnosti otázok a odpovedí (hlasovanie používateľov).

V nastaveniach si môže používateľ upraviť zobrazované meno, email na profilový obrázok gravata¹, upraviť text profilu, nastaviť posielanie emailových notifikácií (notifikácie sú odosielané raz za deň), pridať odkazy na profil na sociálnych sieťach, nastaviť zobrazovanie mena a emailu ostatným používateľom a v prípade, že nejde o účet vytvorený cez AIS LDAP aj zmeniť si používateľské heslo.

Dôležitým prvkom, ktorý sme implementovali v letnom semestri je nástenka (angl. dashboard) na úvodnej stránke po prihlásení. Tento prehľad zobrazuje počet nových otázok, odpovedí a komentárov v systéme a v sledovaných kategóriách od posledného prihlásenia. Súčasťou nástenky je aj prúd aktivít a noviniek v systéme. Aktualizáciu posledného času prihlásenia je možné vykonať ručne, čo je užitočné, ak sa používateľ nezvykne odhlasovať.

Administrácia Askalotu umožňuje:

- pridávať a upravovať kategórie:
 - funkcionality, ktorú sme implementovali v letnom semestri je kopírovanie podstromu kategórií. Okrem toho sme rozšírili vlastnosti kategórií o označenie, či sa majú otázky zdieľať, medzi rovnakými kategóriami v rôznych verziách kurzu/v rôznych akademických rokoch a o vlastnosť, či je možné sa v danej kategórii pýtať otázky,
- priradovať roly používateľov k jednotlivým kategóriám,
- upravovať textový opis verzií systému,
- spravovať novinky na úvodnej stránke:
 - okrem klasického pridávania, úpravy, mazania noviniek je možné nastaviť, či sa má novinka zobrazovať alebo nie. Funkcionality sme implementovali v letnom semestri,
- odoslať hromadný email všetkým používateľom:
 - túto funkcionality sme implementovali v letnom semestri. Odoslať je možné buď čisto textový alebo HTML email.

Administrátori (v *university* prostredí aj učitelia) majú k dispozícii aj štatistiky, kde vidia podrobnú aktivitu a štatistické informácie o obsahu v systéme.

2.1.1. Opis funkcionality v systéme edX

Do systému edX sa vloží Askalot vo forme *iframe* elementu ako záložka kurzu. Pri zobrazení tejto záložky Askalot zistí, či je používateľ prihlásený a ak nie, tak zobrazí upozornenie, že používateľ bude čoskoro presmerovaný na prihlasovaciu lekciu (angl. unit, jednotka kurzu, v ktorej môže - a v

¹ <http://sk.gravatar.com/>

tomto prípade musí, byť vložený Askalot). Ak prihlasovaciu URL nie je možné získať, tak Askalot informuje o tomto stave chybovou hláškou.

Ak je používateľ prihlásený, tak sa Askalot zobrazí v *iframe* elemente ako klasická webová stránka. *Iframe* je z dôvodu lepšieho používateľského zážitku spravovaný knižnicou *iframe resizer*², ktorá zobrazí plnú výšku obsahu Askalotu, vďaka čomu má používateľ pocit, akoby bol Askalot prirodzenou súčasťou systému edX.

Druhým prvkom integrácie do edX je pohľad na lekcii (angl. unit view). Ide o zjednodušený pohľad, v ktorom sú zobrazené otázky z jednej kategórie a formulár na pridanie novej otázky. Komunikácia s Askalotom v tomto prípade prebieha cez Learning Tools Interoperability (LTI) štandard. Systém edX odošle POST dopyt (angl. request), ktorý obsahuje aj s používateľove údaje, do Askalotu, kde používateľa na základe tohto dopytu prihlásime, resp. mu vytvoríme účet. Následne sa Askalot zobrazí v elemente *iframe* a opäť sa použije knižnica *iframe resizer*.

Pri zobrazení Askalotu v systéme edX sa v prípade neprečítaných notifikácií zobrazí počet neprečítaných notifikácií nad odkazom na Askalot. Táto funkcionality aj s pohľadom na lekcii (angl. unit view) je zobrazená na Obr. 1.

2.1.2. Opis funkcionality pre použitie v aplikáciách tretích strán

Zjednodušený pohľad *unit view* sme využili aj pre možnosť integrácie Askalotu do akejkoľvek aplikácie tretej strany. V tomto prípade predpokladáme klasické používanie Askalotu ako webovej aplikácie s možnosťou zobrazenia jednoduchého pohľadu na kategóriu v *iframe* elemente.

V administrácii je potrebné nastaviť pri kategórii atribút *hash*. Tento *hash* sa použije na identifikovanie kategórie a následne je možné pod danou kategóriou automaticky vytvárať podkategórie. Jednotlivé podkategórie sú zobrazené v pohľade na lekcii (angl. unit view). Podrobnejší návod na použitie tejto funkcionality systému uvádzame v samostatnom návode.

² <https://github.com/davidjbradshaw/iframe-resizer>

Dokumentácia k tímovému projektu časť 1 (Inžinierske dielo) [Askalot2edX]

The screenshot displays the AskALot interface within an edX course. At the top, the user is logged in as 'askalotteacher'. The course is viewed from the 'Student' perspective. The navigation menu on the left includes 'Home', 'Course', and 'Askalot' (with 15 notifications). The main content area shows a video player for 'Machine learning > Welcome Video > Welcome Video'. Below the video, there is a section for 'Askalot (External resource)' with a 'New question' button. Two questions are listed:

| Question | Favorites | Votes | Answers | Views |
|--|-----------|-------|---------|-------|
| Octave version Which version of Octave is most suitable for purpose of the project? *[An example of a question which aims to obtain a recommendation]* Added by Sophia 15. March 2016 | 0 | 0 | 1 | 7 |
| What's is the difference between clustering and classification? What is the difference between clustering and classification? *[An example of general question]* Added by Sophia 25. February 2016 | 1 | 0 | 1 | 6 |

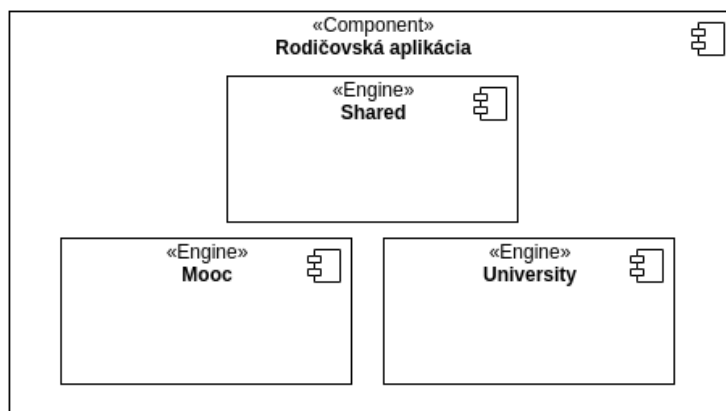
At the bottom of the page, there are links for 'About', 'Blog', 'Help Center', 'Contact', and 'Donate'. The edX logo and 'POWERED BY OPENEDX' are also visible.

Obr. 1 Pohľad na unit view a notifikácie v systéme edX

2.2. Architektúra

Askalot je štandardná webová aplikácia, ktorá je implementovaná pomocou návrhového vzoru MVC - model, pohľad a kontrolér (angl. Model View Controller). Ako databáza sa používa PostgreSQL vo verzii 9.3 a vyhľadávací server ElasticSearch vo verzii 1.1. Aplikácia je implementovaná v jazyku Ruby s využitím rámca Ruby on Rails vo verzii 4.2 a prvky používateľského rozhrania využívajú rámec Bootstrap vo verzii 3. Aplikácia využíva množstvo knižníc tretích strán.

Technológie, v ktorých je aplikácia Askalot vyvinutá, sa oproti pôvodnému riešeniu nezmenili. Výrazne sa však zmenila architektúra z dôvodu potreby udržiavania zdrojového kódu pre dve rôzne prostredia, ktoré časť funkcionality zdieľajú: fakultná verzia (*university*) a MOOC verzia (*mooc*). Na základe diskusie sme sa rozhodli pre využitie modulárnej architektúry prostredníctvom takzvaných *rails engines*. V novom architektonickom riešení existuje jedna aplikácia, ktorá slúži ako obal pre jednotlivé moduly. Podľa konfigurácie sa potom v rámci jednej inštancie Askalotu spustí buď fakultný alebo MOOC modul. Cieľom je oddeliť rozdielne funkcionality a dizajn pre rôzne inštancie, ako napríklad dizajn a špecifickú funkcionality a naopak udržať pokope rovnaké komponenty, ako niektoré časti dátového modelu a kontrolérov. Tento návrh je zachytený na Obr. 2.



Obr. 2 Architektúra riešenia zdieľania funkcionality medzi verziami aplikácie

Aktuálne máme tri moduly: modul pre fakultnú verziu (*university*), modul pre MOOC verziu (*mooc*) a modul s prvkami, ktoré sú spoločné pre prvé dva moduly (*shared*). Vďaka takejto architektúre môžeme využívať jeden repozitár pre zdrojový kód a dochádza k jednoduchému zdieľaniu novej funkcionality medzi verziami Askalotu. Rodičovská aplikácia načíta ako prvý *shared* modul a následne buď *university* alebo *mooc*. Toto poradie nám umožňuje v moduloch *university* a *shared* prepísať všeobecnú funkcionality, ktorá je v zdieľanom module *shared*.

2.3. Infraštruktúra

V rámci vývoja a integrácie systému ďalej používame nástroje na správu zdrojového kódu GIT a službu pre vzdialené ukladanie a manažovanie zmien Github. Vývojári sú povinní pravidelne odosielať zmeny z lokálneho repozitára do vzdialeného repozitára. Tým je celému tímu vždy k dispozícii aktuálna verzia zdrojového kódu.

Po odoslaní kódu sa nad novými zmenami spustia automatické testy v systéme Codeship. Zároveň je toto testovanie prepojené so systémom Github, kde je možné vizuálne vidieť výsledok testovania a prekliknúť sa na detaily. To pomáha pri prehliadke zdrojového kódu a spájaní tzv. *pull-requestov*.

Automatické hodnotenie kvality kódu pomocou systému CodeClimate pravidelne monitoruje zmeny a podľa niekoľkých interných pravidiel vyhodnocuje mnoho ukazovateľov súvisiacich s duplicitami v kóde, bezpečnosťou, štandardmi používanými pri písaní zdrojového kódu a iné.

Aby sme zabezpečili plynulý chod a rýchle riešenie problémov, Askalot je napojený na NewRelic, ktorý monitoruje chyby a správny beh aplikácie v reálnom čase. V prípade problémov príde na tímovú e-mailov adresu oznámenie o chybe s technickými detailami: kedy, komu a aký problém nastal.

Askalot je nasadený na univerzitnom serveri na FIIT a jeho produkčná verzia je dostupná na adrese <http://askalot.fiit.stuba.sk/fiit>. Nasadzovanie je realizované prostredníctvom doplnku (angl. gem) *capistrano*. V súvislosti so zmenou architektúry je definovaných niekoľko plánov pre nasadenie. Každý plán obsahuje kompletnú konfiguráciu, ktorá zabezpečí nakopírovanie zmenených súborov na server, úprava databázového modelu, predpríprava statických súborov a reštartovanie nasadzovanej inštancie. Každý plán sa v konfigurácii odkazuje na rovnomenné prostredie. Tab. 1 zobrazuje plány a ich konfigurácie.

Tab. 1 Nasadzovacie plány a ich konfigurácia.

| Názov plánu | Modul | Prostredie | Vetva | Zložka na serveri |
|-------------------|------------|------------|---------|----------------------------|
| fiit_production | university | production | master | askalot-fiit-production |
| fiit_staging | university | staging | staging | askalot-fiit-staging |
| fiit_demo | university | demo | master | askalot-fiit-demo |
| edx_production | mooc | production | master | askalot-edx-production |
| edx_staging | mooc | staging | staging | askalot-edx-staging |
| edx_demo | mooc | demo | master | askalot-edx-demo |
| lugano_production | university | production | master | askalot-lugano-productionl |

Modul opisuje názov použitého modulu (angl. engine) ktorý sa použije v kombinácii s modulom *shared*. Hoci každý plán používa rovnomenné samostatné prostredie, vychádza toto prostredie zo štandardných nastavení pre pôvodné prostredia opísané v tabuľke.

Na serveri sú zdrojové kódy umiestnené v adresári `/home/deploy/projects/`, kde je každá inštancia umiestnená vo vlastnej podzložke. Každá inštancia má potom nakonfigurovaný *nginx* server, ktorý prístupuje k *unicorn* HTTP serveru. Jednotlivé inštancie bežia každá pod samostatnou URL adresou, napríklad <http://askalot.fiit.stuba.sk/fiit-demo>, <http://askalot.fiit.stuba.sk/edx-staging>. Pokiaľ návštevník pristúpi na hlavnú stránku <http://askalot.fiit.stuba.sk>, bude presmerovaný na produkčnú verziu prostredia pre FIIT - <http://askalot.fiit.stuba.sk/fiit>.

Okrem toho je k dispozícii ešte statická stránka <http://askalot.fiit.stuba.sk/demo>, ktorá slúži ako rozcestník pre *demo inštancie*.

Každá inštancia používa vlastnú Postgres databázu a vlastný Elasticsearch index. Názov databázy a indexu ja podobný názvu zložky, iba namiesto pomlčky obsahuje podtržník, napr. `askalot_fiit_production`.

2.4. Dátový model

Kategórie otázok boli transformované z jednorozmerného zoznamu na stromovú štruktúru. Úrovne týchto kategórií môžu mať rôzny význam. Vo fakultnom systéme predstavujú hlavnú úroveň školské roky, pod ktorými sú jednotlivé predmety a na ďalšej vrstve pod nimi sú cvičenia a prednášky, prípadne iné delenie predmetu. V prípade edX verzie hlavná prvá úroveň predstavuje samotné kurzy, pod ňou sú podkategórie a ďalej jednotlivé lekcie (angl. unit). Tieto úrovne sú aj sémanticky definované v konfiguračnom súbore, ktorý sa môže líšiť pre každú inštanciu. Zároveň pribudli kategóriám atribúty `full_tree_name` a `full_public_name`. Prvý obsahuje kompletný názov kategórie aj s menami rodičovských kategórií oddelených pomlčkou. Druhý obsahuje názvy len tých rodičov, ktorí sa nachádzajú na úrovniach, ktoré sa zobrazujú. V konfiguračnom súbore je pre každú úroveň definované okrem jej názvu aj to, či sa názvy kategórií zobrazujú v celkovom verejnom názve kategórie `full_public_name`.

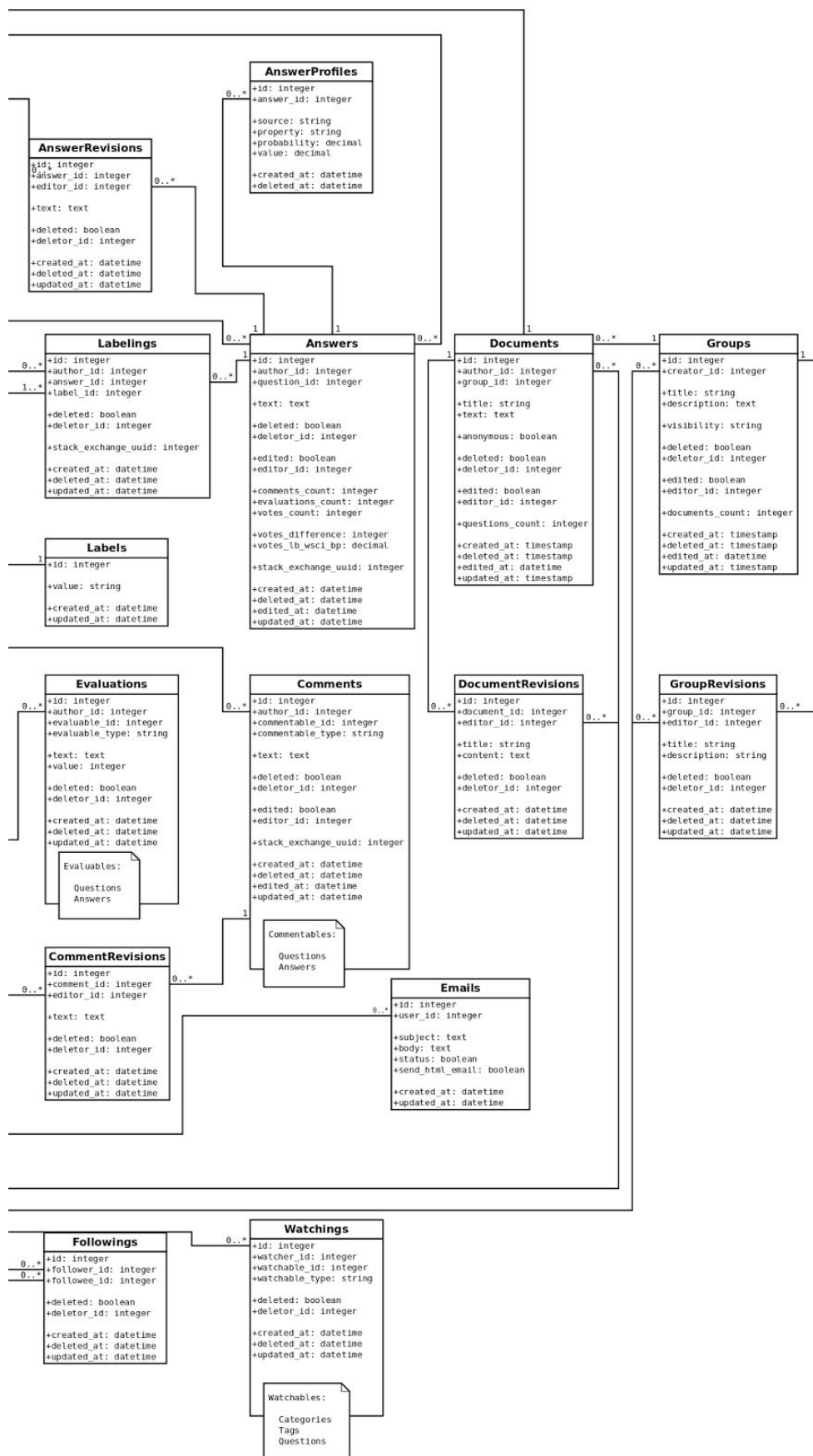
Otázky je možné pridávať na listy v strome kategórií. V nich je ešte možné dodatočne povoliť alebo zakázať pýtať sa parametrom `askable`. Kategórie je možné zdieľať medzi sebou, keďže každá kategória má systémový identifikátor. Ten istý identifikátor sa môže nachádzať na viacerých kategóriách, čo značí, že spolu súvisia. Napríklad predmet DBS v roku 2014 a 2015 majú ten istý identifikátor, čo značí, že ide o ten istý predmet, avšak v iných rokoch. Bude preto možné zobrazovať všetky otázky patriace k predmetu za všetky roky.

Kategórie je možné prispôbovať a určiť, ktoré sú aktívne a dá sa do nich pýtať otázka, prípadne ktoré kategórie a otázky v nich sa majú zdieľať naprieč jednotlivými verziami, čiže s kategóriami, ktoré majú rovnaký systémový identifikátor.

Okrem zmeny v dátovom modeli kategórií a otázok boli pridané ďalšie atribúty a tabuľky, ktoré súvisia s rozšírením štandardnej funkcionality Askalotu. Otázky je možné uzatvoriť, pokiaľ sú bez odpovede, atribút `closed_at`. Taktiež sa k jednotlivým značkám otázok (angl. tag) evidujú dodatočné štatistické hodnoty `max_time`, `min_votes_difference` a `max_votes_difference`. Systém pravidelne kontroluje zoznam používateľov a špeciálne označuje študentov, ktorí už absolvovali štúdium a opustili fakultu, tabuľka `alumni`. Okrem toho je možné hromadné rozposlanie e-mailovej notifikácie všetkým, alebo len vybraným študentom. Tieto e-maily sa ukladajú v novej tabuľke `emails`.

Pri zobrazení otázok v niektorej kategórií je žiaduce, aby sa zobrazovali všetky otázky vo všetkých podkategóriách. Keďže platí pravidlo, že otázky je možné položiť iba v listoch, t.j. najspodnejších kategóriách, ostatné kategórie by sa zobrazovali bez otázok. Kvôli výkonu aplikácie bola pridaná tabuľka `categories_questions`, ktorá prepája otázku so všetkými odpoveďami, kde sa má zobrazovať. Toto prepojenie ovplyvňuje tiež parameter `shared`, ktorý značí, či sa otázky v danej kategórií majú zobrazovať aj v ostatných podobných kategóriách so spoločným identifikátorom. V takom prípade sa pridávajú ďalšie väzby do prepojovacej tabuľky.

Pre rozšírenú funkcionality systému o diskusie sme sa rozhodli pridať typ otázok k jednotlivých otázkam prostredníctvom tabuľky `question_types`. Tento typ slúži na ďalšiu kategorizáciu



Obr. 5 Čiastočný pohľad na dátový model -pravá časť

3. Moduly

Počas uplynulých šprintov sme identifikovali tieto moduly:

- Zapojenie študentov
 - Archivácia používateľov
 - Kategórie sledované učiteľom
 - Odoslanie emailu z Askalotu
 - Anonymná registrácia používateľa
 - Osobná nástenka (angl. dashboard)
 - Integrácia do aplikácií tretích strán
 - Novinky
 - Presun kategórií do ďalšieho roka
 - Fórum
 - Ponuka kategórií na sledovanie
 - Opis kategórií
- Integrácia Askalotu do edX
 - Vloženie Askalotu do edX
 - LTI
 - Pohľad na lekciu (angl. unit)
 - Globálny pohľad
 - Prispôsobenie vloženého Askalotu v edX
 - Integrácia obsahu s edX
 - Štruktúra edX
 - Extrakcia obsahu
 - Univerzálne kategórie
 - Administrácia kurzu
 - Kontext
 - Zdieľanie obsahu medzi verziami kurzu
 - Notifikácia v rozhraní
 - Integrácia používateľov z edX
 - Roly používateľov
 - Vytvorenie účtu, získanie údajov, prihlasovanie
 - Autentifikácia pre globálny pohľad
- Zmena architektúry
 - Modulárna architektúra
 - Prostredia

Opis všetkých modulov vedieme v samostatných dokumentoch *Moduly systému* (v zátvorke pod týmto nadpisom na titulnej strane sa nachádza názov skupiny modulov, ktoré sa v dokumente nachádzajú), ktoré sú priložené k tomuto dokumentu.

4. Príručky

4.1. Inštalčná príručka

Príručka je určená systémovým administrátorom a definuje postup pri vytváraní a nasadzovaní novej inštancie. V prípade, že sa jedná o úplne novú inštanciu, treba vytvoriť nové prostredie (angl. environment), nový nasadzovací plán a nastaviť serverové prostredie. Vzdialený repozitár sa nachádza na adrese <https://github.com/isrba/askalot>.

4.1.1. Vytvorenie nového prostredia

1. Vytvoriť nový súbor `config/environments/{type}_{environment}.rb`, kde `{type}` je názov umiestnenia, napr. `fiit` a `{environment}` je názov štandardného prostredia, napr. `production`.
2. Nastaviť konfiguráciu prostredia, mailový server a iné.
3. V súbore `config/environments.yml` nastaviť podľa názvu súboru prostredia použitie *rails engine* (modulu) a štandardný názov prostredia, napr:

```
fiit_production:
  module: university
  env_type: production
  name: fiit
```

4.1.2. Vytvorenie nového nasadzovacieho plánu

1. Vytvoriť nový súbor `config/deploy/{type}_{environment}.rb`, kde `{type}` je názov umiestnenia, napr. `fiit` a `{environment}` je názov štandardného prostredia, napr. `production`.
2. Nastaviť server, používateľské meno, vetvu, ktorá sa nasadzuje a použité prostredie.

4.1.3. Nastavenie servera

1. Prihlásiť sa na server
 - a. V prípade Askalot servera spustiť príkaz
`ssh deploy@askalot.fiit.stuba.sk`
Treba mať nastavený prístup na server od administrátora.
2. Vytvoriť projektovú zložku, napr.
`/home/deploy/projects/askalot-{type}-{environment}`,
`/home/deploy/projects/askalot-fiit-production` napr.
3. Vytvoriť konfiguráciu pre *unicorn*
 - a. Vytvoriť súbor `/etc/init.d/unicorn-askalot-{type}-{environment}`,
napr. `/etc/init.d/unicorn-askalot-fiit-environment`
 - b. Pomenovať *unicorn* inštanciu
 - c. Nastaviť prístup do projektovej zložky vytvorenej v kroku 2
4. Vytvoriť konfiguračný súbor pre *nginx*
 - a. Vytvoriť konfiguračný súbor
`/opt/nginx/sites-available/askalot-{type}-{environment}`, napr.
`/opt/nginx/sites-available/askalot-fiit-production`
 - b. Nastaviť port, prípadne adresu a volanie na nastavenú *unicorn* inštanciu
 - c. Vytvoriť symbolickú linku

```
ln -s  
/opt/nginx/sites-available/askalot-fiit-production  
/opt/nginx/sites-enabled/askalot-fiit-production
```

5. Spustiť prípravu nasadenia
`cap {type}_{environment} deploy:setup`
6. Nastaviť databázu
 - a. Vytvoriť nového databázového používateľa
 - b. Vytvoriť novú databázu
 - c. Skopírovať `config/database.yml.example` súbor do
`/etc/init.d/unicorn-askalot-{type}-{environment}/shared/config/database.yml`
 - d. Nastaviť prístup do databázy
7. Konfigurácia prostredia - skopírovať a upraviť `config/configuration.yml.example` súbor do
`/etc/init.d/unicorn-askalot-{type}-{environment}/shared/config/configuration.yml`

4.1.4. Nasadenie

Pre správne nasadenie je potrebné, aby všetky zmeny boli nahrané vo vzdialenom repozitári.

1. V príkazovom riadku prejsť do priečinka projektu
2. Spustiť príkaz `cap {type}_{environment} deploy`, napríklad `cap fiit_production deploy`
3. HTTP server *unicorn* sa po úspešnom dokončení nasadenia reštartuje a aplikácia bude dostupná na nakonfigurovanej adrese a porte.