

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: 2
Dátum poslednej zmeny: 14.12.2015

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, prípadne obsahuje časti systému, ktorým sa plánujeme venovať v najbližšej dobe.

1. Globálne ciele pre zimný semester

Pri plánovaní cieľov na zimný semester sme museli brať do úvahy fakt, že vo februári sa začína kurz, na ktorom bolo sľúbené používanie Askalotu, preto je našim prioritným cieľom mať dovedy hotovú a správne fungujúcu integráciu.

Celkovo sme si naplánovali niekoľko krokov:

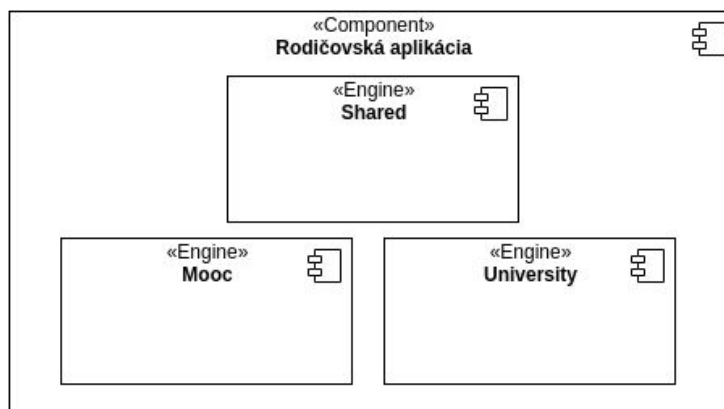
1. Zoznámenie sa s Askalotom.
 - Práca nášho tímu spočívala v úprave existujúceho riešenia. V tomto kroku sme sa preto zoznamovali s Askalotom po dizajnovej a implementačnej stránke. Hoci je Askalot open source, väčšina tímu videla jeho zdrojový kód na začiatku semestra po prvý krát. Viacerí členovia 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 navrhnuť 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.

2. Celkový pohľad na systém

2.1. 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 a MOOC verzia. 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 kontrolerov. Tento návrh je zachytený na Obr. 1.



Obr. 1 Architektúra riešenia zdieľania funkcionality medzi verziami aplikácie

Aktuálne máme tri moduly: modul pre fakultnú verziu, modul pre MOOC verziu a modul s prvkami, ktoré sú spoločné pre prvé dva moduly. 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*.

2.2. 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.

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>. 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	Port	Zložka na serveri
fiit_production	university	production	gama	13021	askalot-fiit-production
fiit_staging	university	staging	gama	13011	askalot-fiit-staging
fiit_demo	university	demo	gama	13001	askalot-fiit-demo
fiit_experimental	university	experimental	gama	13031	askalot-fiit-experimental
edx_production	mooc	production	gama	13020	askalot-edx-production
edx_staging	mooc	staging	gama	13010	askalot-edx-staging
edx_demo	mooc	demo	gama	13000	askalot-edx-demo
edx_experimental	mooc	experimental	gama	13030	askalot-edx-experimental

Modul opisuje názov použitého modulu, *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. Momentálne je vo všetkých prostrediach nasadená vývojová vetva *gama*, ktorá ako jediná obsahuje novú modulárnu architektúru. V budúcnosti sa tieto vetvy oddelia pre jednotlivé prostredia.

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 pristupuje k *unicorn* HTTP serveru. Jednotlivé inštancie bežia každá na vlastnom porte.

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

2.3. Dátový model

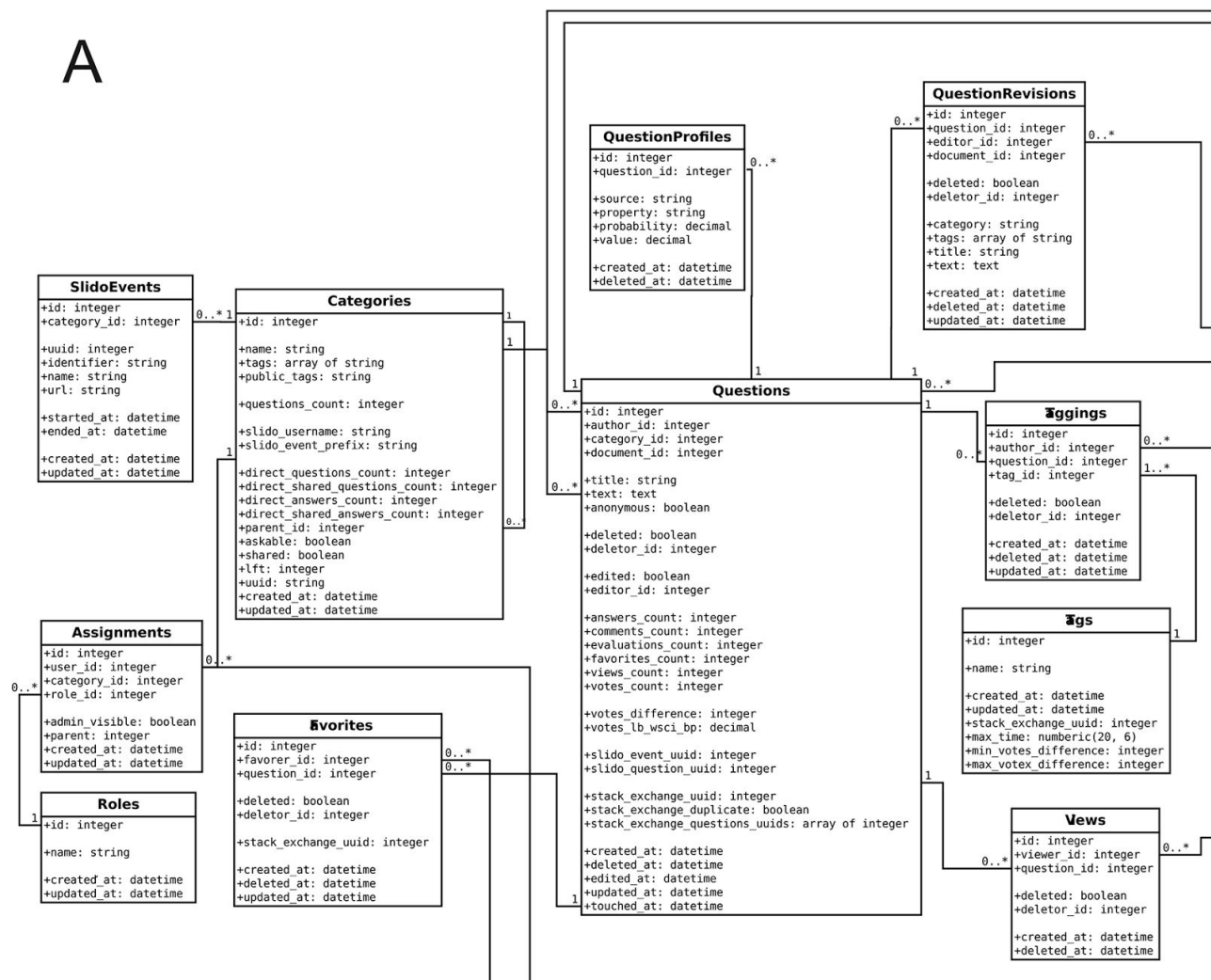
Kategórie otázok boli transformované z jednorozmerného zoznamu na stromovú štruktúru. Úrovne týchto kategórií môžu mať rôzny význam. Vo fakultnom systéme predstavuje prvá vrstva školské roky, pod ktorými sú jednotlivé predmety a na ďalšej vrstve pod nimi sú cvičenia a prednášky. V prípade edX verzie môže prvá úroveň predstavovať kurzy, ďalšia jednotlivé moduly a videá. 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`.

Na ľubovoľnej úrovni môže byť zavesená otázka. 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.

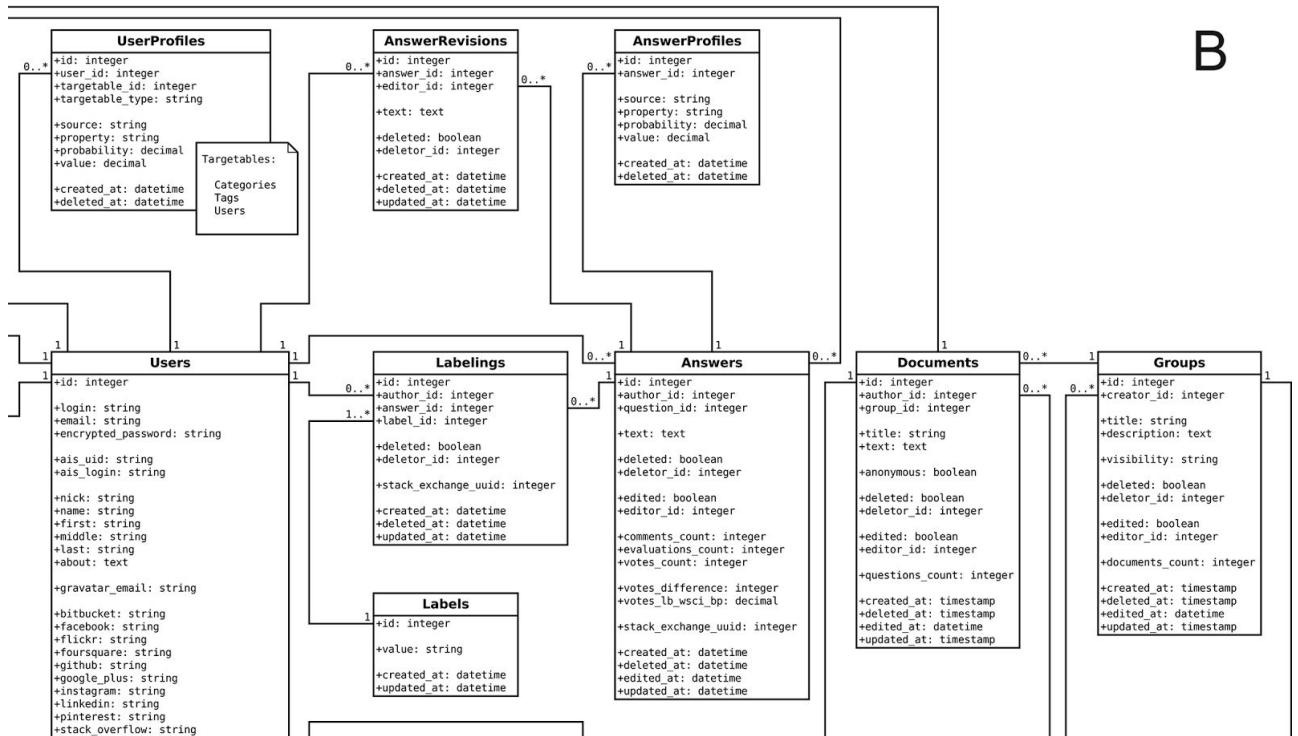
Kategórie je možné prispôbovať a určiť, ktoré sú aktívne a dá sa do nich pýtať otázka, prípadne ktoré kategórie a otázky v nich sa majú zdieľať naprieč jednotlivými verziami, čiže s kategóriami, ktoré majú rovnaký systémový identifikátor.

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. tags) 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`.

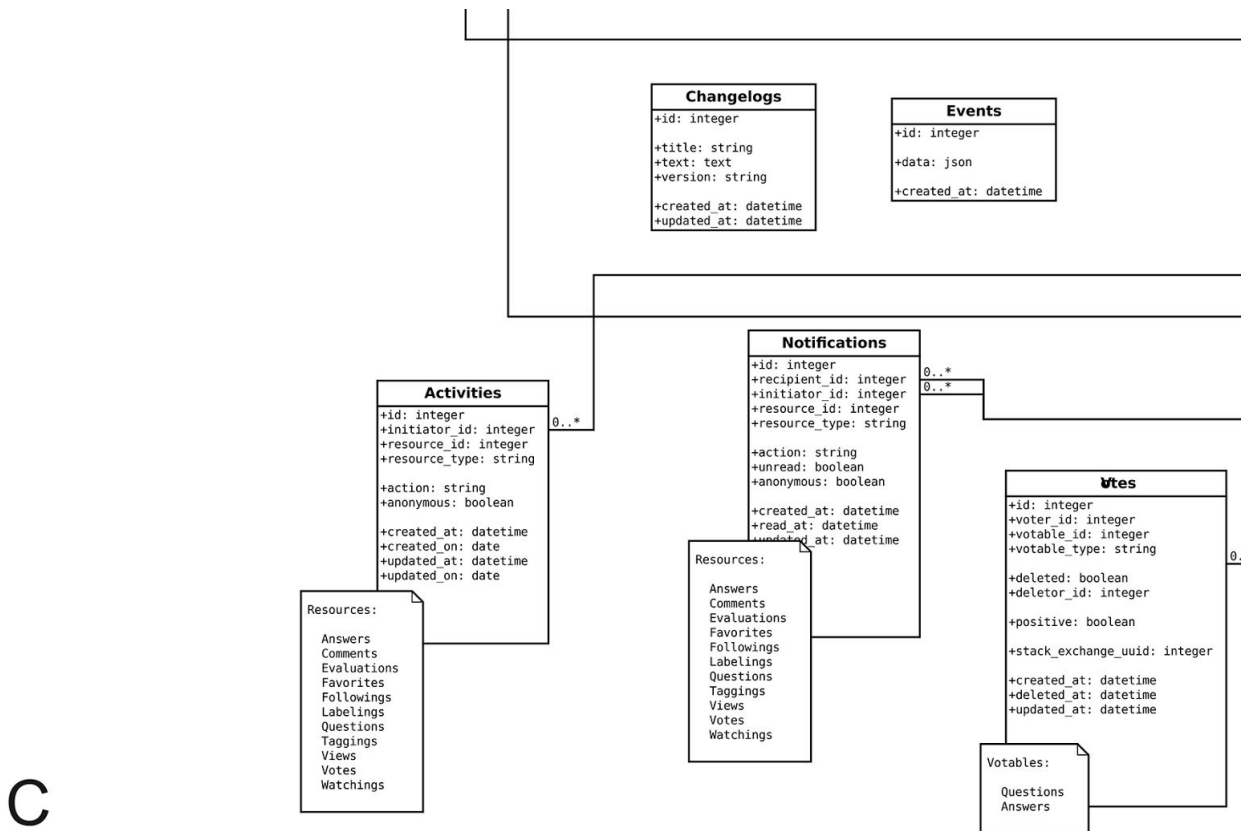
Diagram dátového modelu zobrazujeme na Obr. 2, Obr. 3, Obr. 4, Obr. 5 a Obr. 6. Prvý obrázok zachytáva pohľad na celý dátový model a zvyšné oblasti sa pre zabezpečenie lepšej čitateľnosti sústreďujú na časti diagramu.



Obr. 3 Čiastočný pohľad na dátový model - časť A

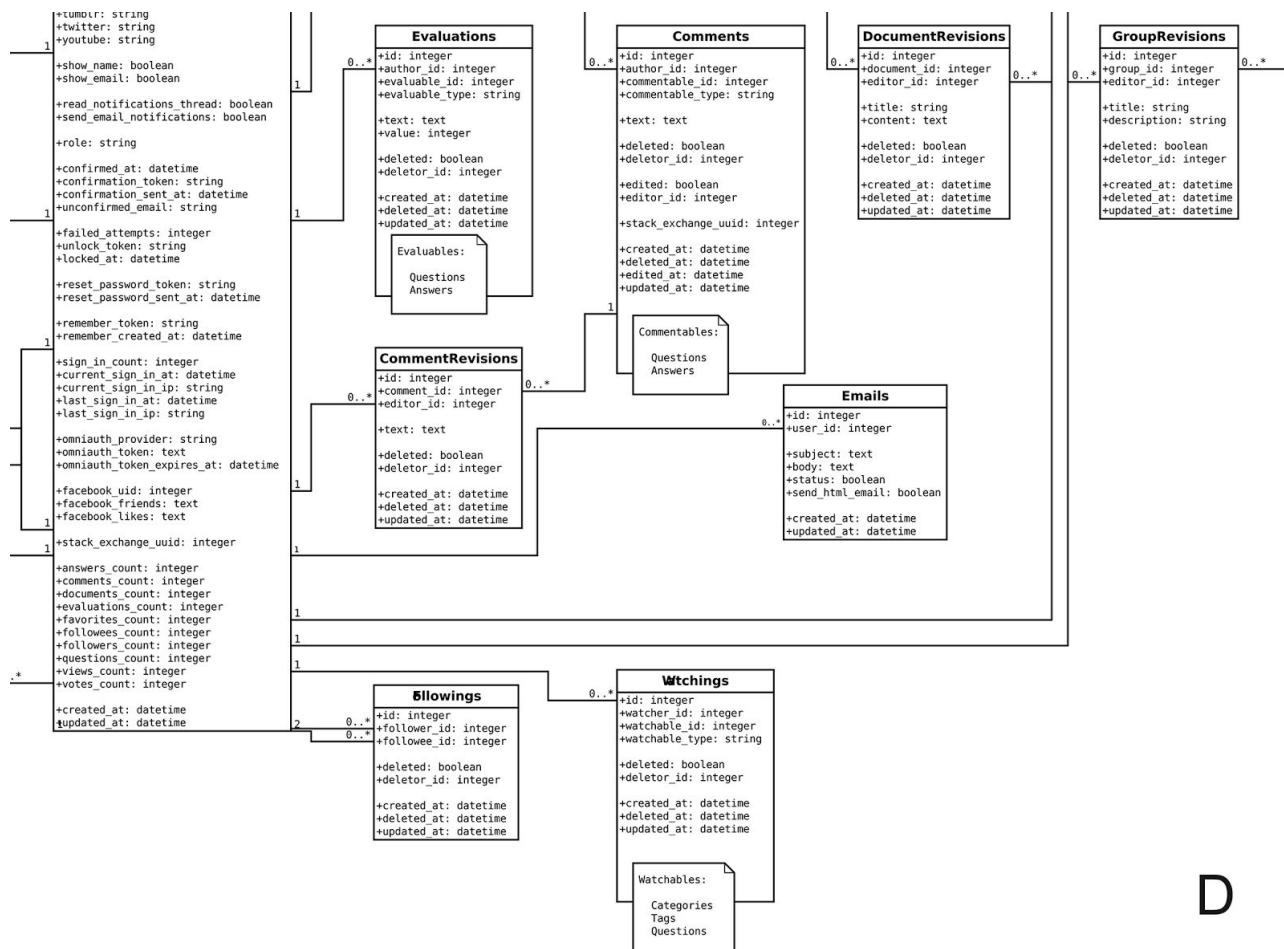


Obr. 4 Čiastočný pohľad na dátový model - časť B



Obr. 5 Čiastočný pohľad na dátový model - časť C

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



D

Obr. 6 Čiastočný pohľad na dátový model - časť D

3. Moduly

Počas uplynulých šprintov sme identifikovali tieto moduly:

- Zapojenie študentov
 - Archivácia otázok, používateľov
 - Spýtanie sa v rovnakej kategórii
 - Kategórie sledované učiteľom
 - Odoslanie emailu z Askalotu
- 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 kurzov
 - Extrakcia obsahu
 - Univerzálne kategórie
 - Administrácia kurzu
 - Integrácia používateľov z edX
 - Vytvorenie účtu, získanie údajov
 - Autentifikácia pre globálny pohľad
- Zmena architektúry
 - enginy, 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. Naplnenie cieľov za zimný semester

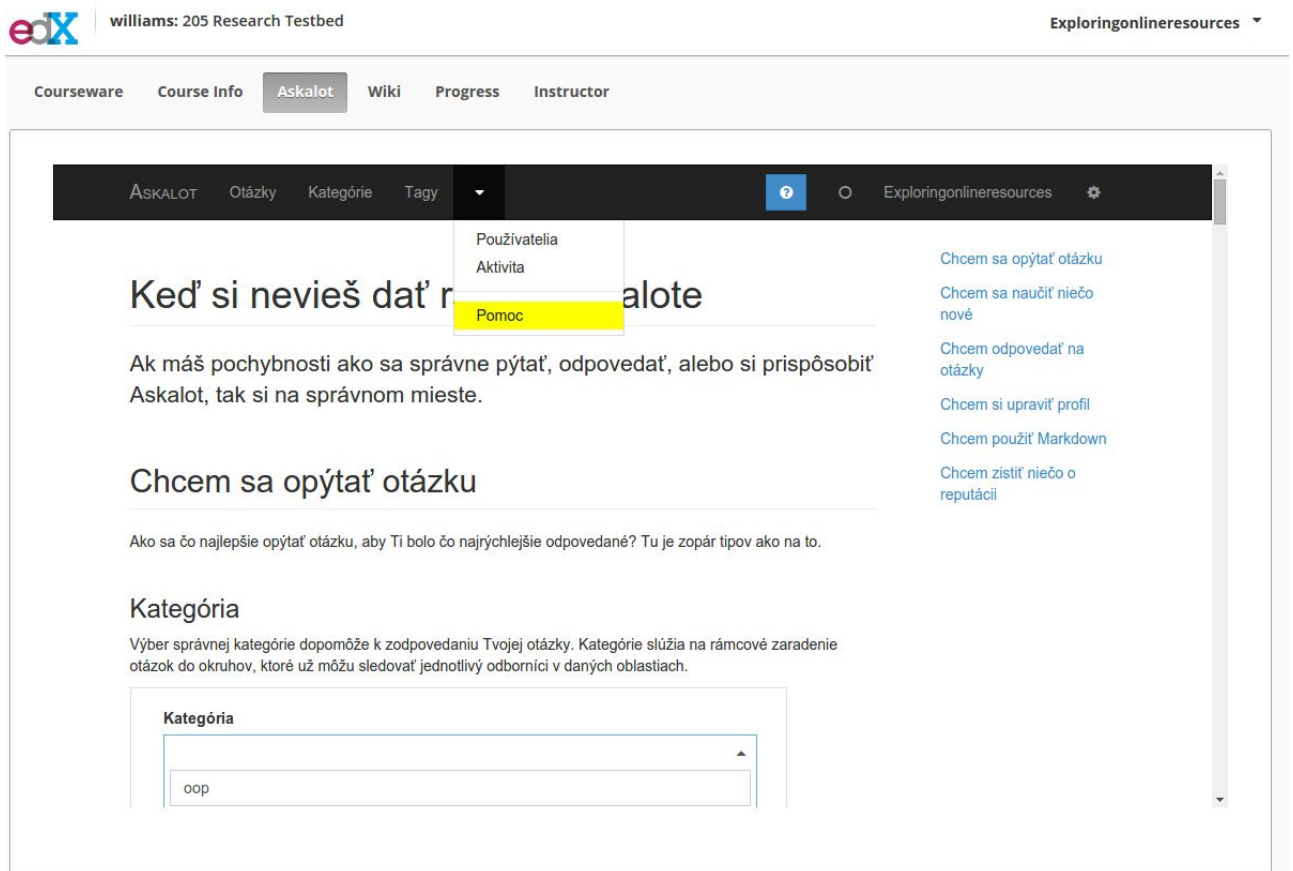
Po dokončení úloh zimného semestra môžeme zhodnotiť, že sa nám vo veľkej miere podarilo naplniť ciele stanovené na začiatku semestra. Všetci členovia tímu sa úspešne zoznámili s Askalotom a zlepšili si svoje znalosti z oblasti vývoja Ruby on Rails aplikácií. Podarilo sa nám analyzovať rôzne možnosti 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.

5. Príručky

5.1. Používateľská príručka

Príručka je určená predovšetkým inštruktorom a definuje postup ako vložiť globálny pohľad a pohľad na úrovni lekcie. Príručka pre používateľov je zahrnutá v systéme Askalot v časti Pomoc (Help), ktorá je prístupná z globálneho pohľadu tak, ako je zobrazené na Obr. 7.

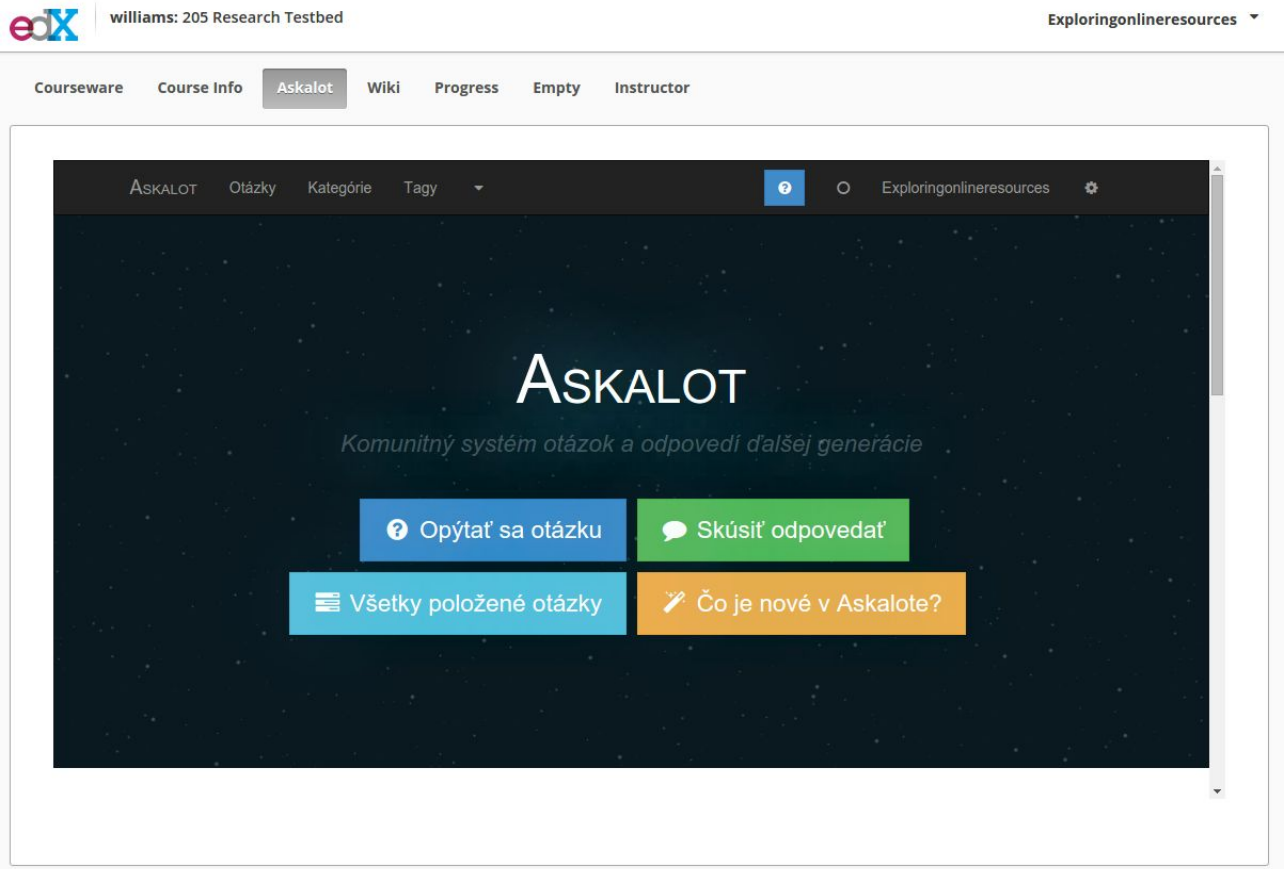


Obr. 7 Používateľská príručka pre systém Askalot

5.1.1. Global view

1. Otvoriť si edX v studio móde s príslušným kurzom. (napr. <https://studio.edge.edx.org/course/williams/205/a>)
2. Prejsť do nastavení stránok cez Content -> Pages a vytvoriť novú stránku s názvom Askalot
3. Vymazať celý obsah v editore
4. Upraviť obsah ako HTML a pridať nasledujúci obsah

```
<p id="askalot-wrapper"></p>  
<script src="http://askalot.fii.stuba.sk:13020/mooc/public.js"></script>
```
5. Po uložení zmien a prekliknutí na Askalot stránku by sa mala zobraziť stránka ako na Obr. 8 čím je tento proces úspešne ukončený.



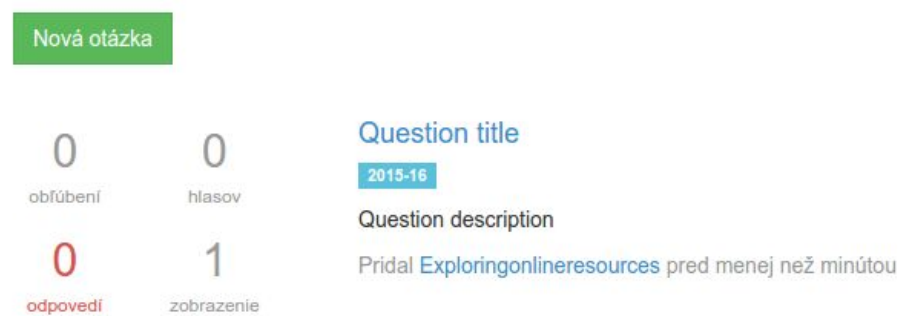
Obr. 8 Používateľská príručka pre systém Askalot

5.1.2. Unit view

1. Otvoriť si edX v studio móde s príslušným kurzom. (napr. <https://studio.edge.edx.org/course/williams/205/a>)
2. Prejsť do rozšírených nastavení cez Settings -> Advanced Settings
3. V časti *Advanced Module List* pridať modul "Iti", napr.
[
 "iti"
]
4. Do časti LTI passports pridať LTI passport, ktoré bude poskytnutý so systémom Askalot napr.
[
 "askalot_id:askalot_key:askalot_secret"
]
5. Uložiť nastavenia a prejsť do konkrétneho unitu, v ktorom chceme pripojiť diskusiu
6. V unite view pridať *Advanced komponent LTI*. Tento komponent musí byť posledným LTI komponentom v unite
7. V nastaveniach komponentu je potrebné zvoliť nasledujúce parametre:
 - a. Do *Custom Parameters* pridať pár *page_url* a url adresu smerujúcu na globálny pohľad askalotu v aktuálnom kurze (napr. [page_url=https://preview.edge.edx.org/courses/williams/205/a/f206b3c87a5e48a19aeb4c6f3d0e74a9/](https://preview.edge.edx.org/courses/williams/205/a/f206b3c87a5e48a19aeb4c6f3d0e74a9/))
 - b. Do *Display name* pridať hodnotu *Askalot*
 - c. *Hide External Tool* nastaviť na *false*

- d. Do *LTI ID* zadať hodnotu ID takú ako bola definovaná v LTI passport zo 4. kroku
 - e. Do *LTI URL* zadať <http://askalot.fiit.stuba.sk:13020/units>
 - f. *Open in New Page* nastaviť na *false*
 - g. *Request user's email* nastaviť na *true*
 - h. *Request user's username* nastaviť na *true*
8. Uložiť nastavenia (Pozn. Diskusia nie je funkčná v studio móde, pretože LTI neodosiela všetky potrebné parametre. Pre správne zobrazenie je potrebné prejsť do *Preview*). V *Preview* by sa mala zobrazovať diskusia s možnosťou pridať novú otázku ako je zobrazené na Obr. 9.

ASKALOT (EXTERNAL RESOURCE)



Obr. 9 Pohľad na diskusiu v unit pohľad

5.2. 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>.

5.2.1. 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
```

5.2.2. 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.

5.2.3. 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},` napr.
`/home/deploy/projects/askalot-fiit-production`
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 vyvorenej 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/conf`
`ig/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/con`
`figuration.yml`

5.2.4. 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.