Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Askalot meets Harvard Courses at edX

[Askalot2edX]

Moduly systému

(Integrácia Askalotu do edX)

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:	1
Dátum poslednej zmeny:	14.12.2015

Úvod

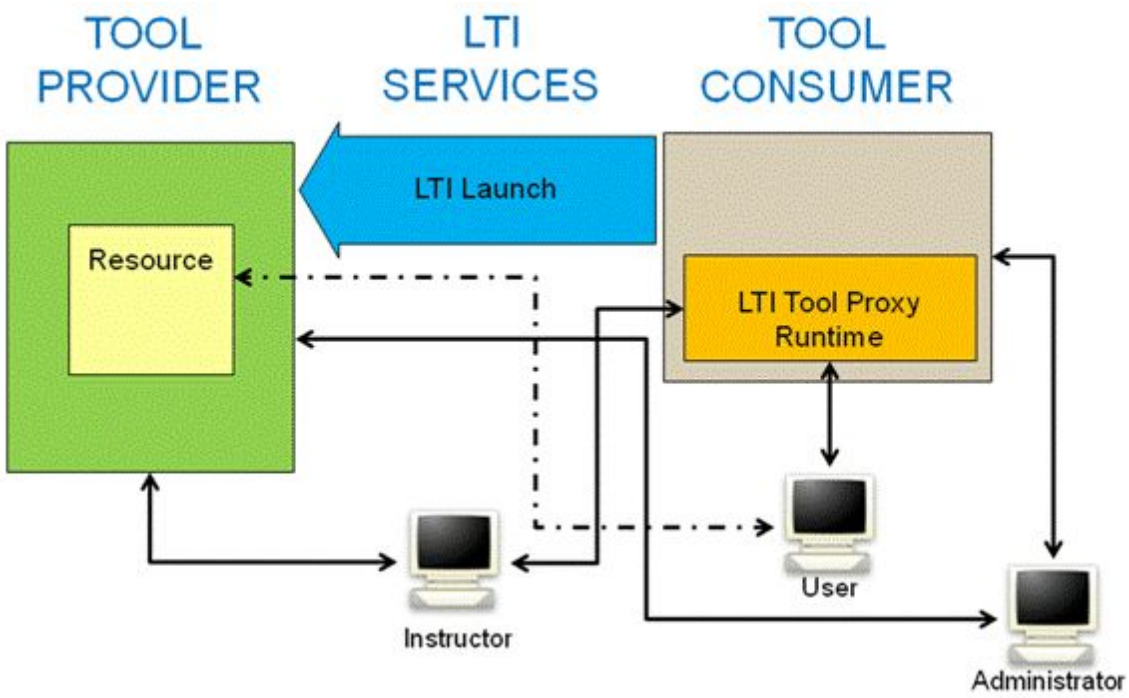
Tento dokument obsahuje všetky navrhnuté a implementované časti modulu *Integrácia Askalotu do edX*. Ku každému z nich sú uvedené štyri zložky: analýza, návrh, implementácia a testovanie.

1. Vloženie Askalotu do edX

Celková analýza
Po zoznámení sa s rozhraním edX kurzov sme vyhodnotili, že Askalot bude potrebné zobrazovať v rámci lekcie a aj globálne ako stránka kurzu. Následne sme analyzovali rôzne konkrétne možnosti tohto vloženia.

1.1. LTI

Analýza
Pre zakomponovanie systému Askalot do systému edX sme počas analýzy našli dve riešenia - zobrazovať Askalot ako iframe alebo ako LTI komponent. LTI komponent (ang. learning tool interoperability), predstavuje rozhranie, ktoré umožňuje zakomponovať obsah externých nástrojov do systémov LMS (ang. learning management system). Zistili sme, že LTI nám automaticky autentifikuje používateľa, čo je jeho hlavná výhoda oproti použitiu iframe. Ďalšou analýzou bolo zistiť, aké informácie poskytuje edX v roli Tool Consumera externým systémov v rolách Tool Providera. Dôležité sú najmä informácie o používateľovi ako email a používateľské meno. Z analýzy sme zistili, že edX poskytuje používateľské ID a jeho rolu v systéme. Ďalej poskytuje kompletný URL link, z ktorého sa pristupuje do Tool Providerov. Na základe linku vieme zistiť id kurzu a podsekcií. Používateľský meno a email je možné získať po nastaveniach Request user's email a Request user's username na true.



Obr 2. *Pohľad na LTI architektúru*

Návrh
V našom prípade systém edX vystupuje ako Tool Consumer, teda obsahuje externé nástroje, ktoré vystupujú ako Tool Providers čiže poskytujú služby iným. V rámci úlohy je potrebné implementovať OAuth autentifikáciu do systému Askalot, ktorú LTI komponent používa na zabezpečenia spojenia medzi nástrojmi. Po overení požiadavky je potrebné automaticky prihlásiť používateľa na základe jeho ID a sprístupniť mu funkcie systému Askalot.

Implementácia
V implementácii bol pre autentifikáciu použitý gem *ims-lti*. Pomocou tohoto gemu overujeme tajný kľúč odoslaný systémom edX a po úspešnom overení automaticky vytvoríme používateľa s daným ID, pokiaľ v systéme ešte neexistuje a prihlásime ho. Zaznamenáme jeho emailovú adresu a používateľské meno.

Testovanie
Testovanie registrácie používateľa do Askalot databázy pomocou LTI prebehlo manuálne.

1.2. Pohľad na lekciu (angl. unit)

Analýza
Edukačný materiál v systéme edX sa nachádza v lekciách-unitoch. LTI komponent je možné k týmto unitom priložiť. Problém bol v tom, že pohľad na otázky z globálneho pohľadu obsahoval priveľa informácii, ktoré v rámci diskusie nie sú relevantné. Na základe celkovej analýzy sme určili, že v každej lekcii bude potrebný zjednodušený náhľad na otázky a odpovede.

Návrh a implementácia
V rámci integrácie Askalotu do edX bolo potrebné upraviť vzhľad zoznamu otázok, ktoré sa budú zobrazovať pod jednotlivými lekciami. Keďže priestor v komponente lekcii je obmedzený, boli odstránené nadbytočné prvky:

- hlavné menu s footrom
- taby s filtrami otázok
- zbytočne veľké okraje
- gravatar
- formulár pre novú otázku sa skrýva pod tlačidlo
- hlavná kategória už nie je zobrazovaná

Samotný unit sa zobrazuje po LTI requeste, kde na základe LTI component ID vieme nájsť príslušný zoznam otázok a tento zoznam zobraziť.

Testovanie
Testovanie správne zobrazovaného obsahu prebehlo manuálne.

1.3. Globálny pohľad

Analýza
Takisto ako je potrebný samostatný pohľad na otázky a odpovede, taktiež bude potrebný globálny pohľad na všetky otázky a odpovede.

Návrh a implementácia
Rozhranie edX kurzov obsahuje stránky s rôznym obsahom. Tieto stránky slúžia napríklad ako miesto pre prehľad hodnotení testov, informácie o štruktúre kurzu, pre diskusné fórum a iné. Z tohto dôvodu je žiadúce, aby sa aj Askalot zobrazoval ako jedna z týchto stránok. Askalot je v globálnom pohľade riešení ako iframe, ktorý zobrazuje klasický pohľad na Askalot.

1.4. Prispôsobenie vloženého Askalotu v edX

Analýza
Napriek tomu, že aktuálne rozhranie systému je dobre navrhnuté, nie je toto rozhranie vhodné aj pre zobrazenie v systéme edX. Preto bolo potrebné navrhnúť prispôsobenie vloženého Askalotu.

Návrh a implementácia

Je potrebné prispôsobovať veľkosť vloženého Askalotu v edX vo vloženom rámci (angl. iframe). Použili sme knižnicu *iframe-resizer*, ktorá sa skladá z dvoch častí. Prvú časť sme vložili do JavaScriptu, ktorý sa nachádza na strane Askalotu, druhú časť na strane edX v komponente, kde sa nachádza extrakcia obsahu. Pre každý komponent, v ktorom sa nachádza náš rámec (angl. iframe) bude potrebné vložiť HTML komponent, ktorý načíta JavaScript s verejnou časťou knižnice iframe-resizer

2. Integrácia obsahu s edX

2.1. Štruktúra edX

Analýza

Obsah edX je uložený v stromovej štruktúre, kde na prvej úrovni sa nachádzajú kurzy. Kurz sa skladá zo sekcií, ktoré majú podsekcie. Sekcia je v doméne edX nazývaná aj modul alebo týždeň. Podsekcia obsahujú lekcie. Lekcia sa skladá z komponentov. Navigácia v sekciách a podsekciách prebieha pomocou bočného menu. Komponenty môžu byť typu HTML kód, LTI komponent, video, pdf.

Návrh a implementácia

Pre ukladanie štruktúry kurzov sa používajú hierarchické kategórie. Prvá úroveň je kurz. Ten pozostáva z viacerých častí - sekcii, ktoré tvoria druhú úroveň. Tretia úroveň je podsekcia (napr. lekcia v týždni) a posledná úroveň je lekcia (menšia časť jednej podsekcie). Každá lekcia sa ešte skladá z viacerých komponentov rôzneho typu, nás bude zaujímať pri spracovaní *lti komponent*.

Testovanie

Táto časť nevyžadovala žiadne testovanie.

2.2. Extrakcia obsahu

Analýza

Potrebné informácie sa nachádzajú v zdrojovom kóde stránky. Nachádzajú sa tu všetky názvy a všetky identifikátory (id) potrebné pre vytvorenie štruktúry v databáze.

Návrh a implementácia

Pre extrakciu obsahu z lekcie sme použili skript, pomocou ktorého získame potrebné informácie. Zo stránky extrahujeme: kurz (id + názov), sekcia (id + názov), podsekcia (id + názov), lekcia (id + názov), lti (id) a navyše obsah lekcie. Skript sa spustí pri načítaní lekcie. Vytvorenie stromovej štruktúry v databáze prebieha nasledovne: ak sa v databáze nájde kategória so zisteným lti, ktorá nemá žiadneho rodiča, tak sa vytvorí celá štruktúra odhora nadol a kategória sa pod ňu zavesí, ak taká kategória v databáze nie je vôbec, vytvorí sa taká kategória, vytvorí sa pre ňu štruktúra a zavesí sa opäť pod ňu. Ak sa kategória so zisteným lti v databáze nachádza a má aj celú štruktúru, nedeje sa nič. V prípade, že komponentov je v lekcii viac ako jeden, nájde sa posledný a berie sa jeho id (aktuálne vychádzame z predpokladu, že diskusia - náš lti komponent bude vždy na konci lekcie - teda ako jeho posledný komponent).

Testovanie

Uvedená funkcionalita bola otestovaná manuálne, priamo cez prehliadač.

2.3. Univerzálne kategórie

Analýza

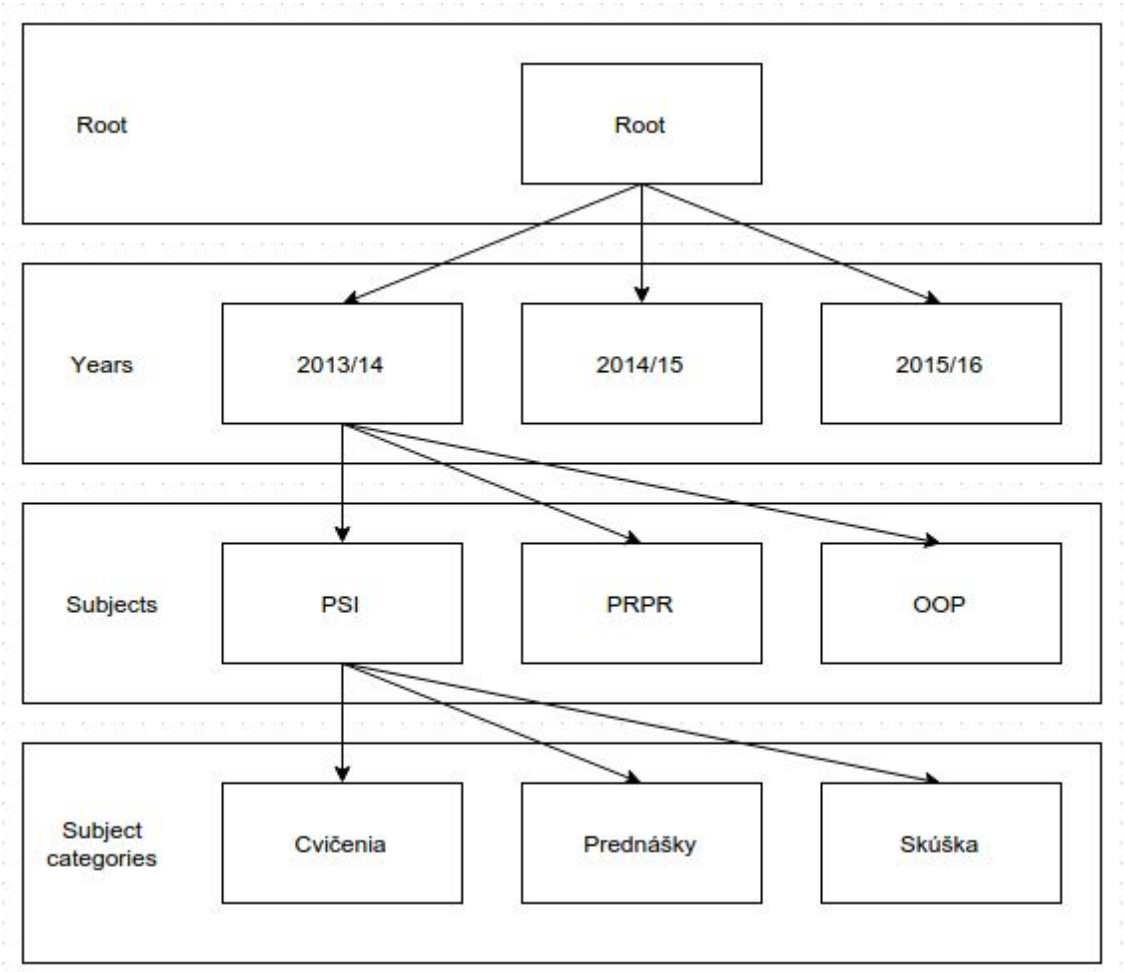
Bolo potrebné upraviť štruktúru kategórií tak, aby sa dala použiť v rámci Askalotu aj edX.

Návrh

Aby sme dokázali obsiahnuť štruktúru edX aj Askalotu v jednom modeli, rozhodli sme sa upraviť model kategórií tak aby sa dali ukladať ako hierarchia. Aby sa následne dalo v danej hierarchii rýchlo dotazovať, rozhodli sme sa použiť databázovov techniku *nested set*. Tak isto je potrebne

zmigrovať aktuálnu štruktúru Askalot-u na do hierarchie. Hierarchia sa skladá z koreňa na ktorého deťmi sú školské roky, pod nimi sú predmety a listy sú tvorené už neštandardizovanými kategóriami ako napríklad cvičenie, prednáška atď.

Implementácia
Na implementáciu *nested set*-u sme využili gem *Awesome Nested Set*. Následne bola vytvorená séria migrácií ktoré najskôr vytvorili štruktúru *nested set*-u, napojili roky z tagov na koreň stromu, následne sú napojené predmety a ich kategórie za využitia kódu z *CategoriesHelper* na združovanie kategórií podľa predmetov. Ako posledný krok bolo treba upraviť referencie z tabuliek *questions* a *assignments*.



Obr 1. *Príklad štruktúry kategórií z Askalot-u pre FIIT*

Testovanie
Vzhľadom na neukončenú migráciu testov pri prechode na modulárnu architektúru bola uvedená funkcionalita otestovaná manuálne, priamo cez prehliadač. Automatické testy budú napísané počas implementácie ďalších úloh.

2.4. Administrácia kurzu

Analýza
Administrátor alebo správca chce upravovať kategórie. Na to, aby takúto akciu mohol vykonať, potrebuje vhodné prostredie. Potrebuje meniť názov, vytvárať nové kategórie a nastavovať, do

ktorých kategórií je možné vložiť otázku a ktoré kategórie sú zdielané. Zdielanie znamená, že v danej kategórii sa zobrazujú aj otázky zo starších kategórii, ktoré spolu súvisia. V kontexte edX to znamená, že sa zobrazujú otázky z predošlých kurzov.

Návrh a implementácia
Bol pridaný model `CategoryDepth`, ktorý popisuje sémantiku úrovní. Úrovne sa konfigurujú v `config/configuration.yml` Do modelu *Category* boli pridané stĺpce
depth a *children_count* (podporované doplnkom *awesome nested set*), *full_tree_name* - úplný názov s rodičmi, cez pomlčky, *full_public_name* - názov s vybranými rodičmi (napr. iba predmet a časť, bez školského roku). Úrovne, ktoré sú súčasťou verejného názvu sú definované v modeli *CategoryDepth*, atribút *is_in_public_name.* Tieto úplne názvy sú automaticky aktualizované pri zmene v kategórii.

Do modelu *Category* boli tiež pridané metódy: *all_directly_related_questions* - všetky aj predošlé otázky k danej kategórii (zavisí od *shared* atribútu), *all_related_questions* - všetky aj predošlé otázky k celému podstromu (zavisí od *shared* atribútu).

Kategórie majú vlastnosť *askable*, a podľa toho sa zobrazujú medzi kategóriami, na ktoré sa dá pýtať. V administrácii sú kategórie zobrazené v strome (vlastný stromový pohľad), dá sa oznacovať *shared/askable* vlastnosť. *Ctrl+Click* označí/odznačí celý podstrom. V rámci úlohy boli opravené aj výkonnostné (angl. performance) problémy pri zobrazovaní administrácie a načítavní kategórií podporovaných učiteľom. Administrácia bola z pôvodného jednostránkového pohľadu rozdelená na viacero podstránok.

Testovanie
Vzhľadom na neukončenú migráciu testov pri prechode na modulárnu architektúru bola uvedená funkcionalita otestovaná manuálne, priamo cez prehliadač. Automatické testy budú napísané počas implementácie ďalších úloh.

3. Integrácia používateľov z edX

3.1. Vytvorenie účtu, získanie údajov

Analýza

V rámci modulu Vloženie Askalotu do edX sme zhodnotili, že využitím LTI komponentov vieme získať o používateľovi základné informácie ako používateľské meno, email a jeho ID v systéme edX. K viacerým dátam sme sa nedostali a preto sme implementovali registráciu len s týmito dátami.

Návrh a implementácia

LTI komponent používa na autentifikáciu OAuth protocol a pre zjednodušenie tejto implementácie sme sa rozhodli použiť existujúce riešenia pomocou gemu *ims-lti*. Po prijatí LTI requestu najskôr overíme, či už je používateľ prihlásený. Ak áno zobrazíme mu otázky pre daný unit pohľad na základe LTI component ID. Ak používateľ prihlásený nie je, overíme správnosť tajného kľúča a hesla potrebného pre OAuth autentifikáciu. Ak tieto hodnoty nie sú spŕavne zobrazíme chybové hlásenie. Nakoniec hľadáme používateľa v databáze Askalotu a v prípade, že sa tam nenáchadza zaznamenáme jeho údaje a vytvoríme mu session.

Testovanie

Testovanie prebehlo manuálne.

3.2. Autentifikácia pre globálny pohľad

Analýza

Po rozhodnutí zobrazovať Askalot ako LTI komponent sme zistili, že edX neumožňuje vkladať LTI komponenty do stránok kurzov. Nakoľko je autentifikácia používateľa dôležitým prvkom bezpečnosti, hľadali sme riešenie ako tento problém vyriešiť. Autentifikáciu zabezpečujú LTI komponenty, preto sme museli navrhnúť riešenie, ktoré bude ich využitie zahŕňať.

Návrh

Prirodzeným návrhom je presmerovať používateľa na lekciu, ktorá obsahuje Askalot. Po prihlásení by mal byť používateľ presmerovaný späť na globálny pohľad. Otázne je, ako vedieť, že sa používateľ dostal na lekciu kvôli tomu, že nebol prihlásený. Tento problém sme sa rozhodli riešiť vytvorením špeciálnej lekcie v rámci kurzu, ktorá bude slúžiť len pre prihlasovanie používateľov.

Implementácia

Riešenie sme implementovali ako *before_filter*, ktorý sa vykonáva pred volaním všetkých akcií. Tento filter nie je používaný v *controllery*, ktorý je určený pre zobrazovanie otázok pre lekciu, nakoľko tento *controller* zabezpečuje prihlasovanie používateľa. Daný filter obsahuje kontrolu, či je ako URL parameter zadaná adresa lekcie, na ktorú má používateľa presmerovať. Ak je, tak sa používateľovi na 5 sekúnd zobrazí informačný text, že nie je prihlásený a bude preto presmerovaný. Po 5 sekundách bude automaticky presmerovaný, alebo môže manuálne kliknúť na odkaz, ktorý je zobrazený pod textom. Tento text zobrazuje pre prípad, že by používateľ mal vypnutý javascript.

Po presmerovaní na lekciu je používateľ prihlásený. V *controllery* sa kontroluje, či je zadaný parameter, že táto lekcia slúži len na prihlasovanie. Ak áno, tak je používateľ hneď presmerovaný na globálny pohľad.

Testovanie

Testovanie prebehlo manuálne. Askalot bol zobrazený v rámci HTML elementu iframe, pričom url adresa mala správne nastavený parameter pre presmerovanie. Testom sme overili, že okno prehliadača bolo automaticky správne presmerované na požadovanú adresu. V druhom scenáry sme manuálne klikli na odkaz a overili správnosť presmerovania.

V treťom scenári sme parameter pre presmerovanie nenastavili. Testom sme zistili, že systém tento problém správne identifikoval a zobrazil príslušné upozornenie pre používateľa. Presmerovanie späť na globálny pohľad po prihlásení sa nám nepodarilo otestovať, nakoľko pre otestovanie tohto prípadu sme potrebovali implementáciu z inej úlohy, ktorá ešte nebola dokončená.