

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: 2
Dátum poslednej zmeny: 19.05.2016

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

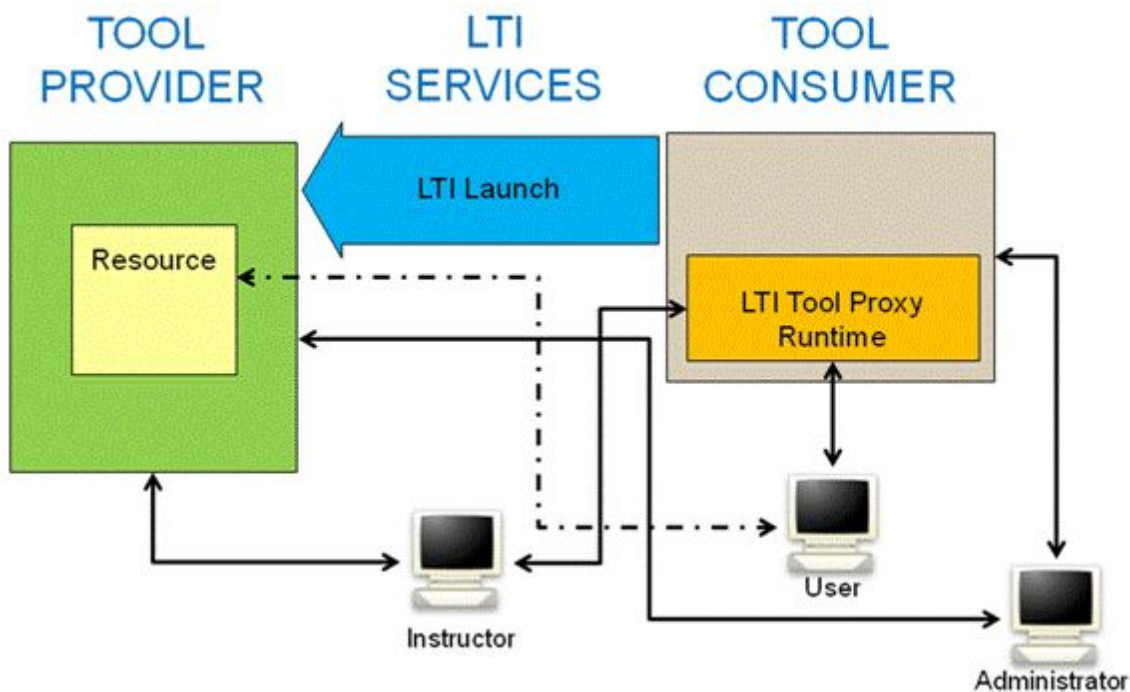
Celková analýza

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 úlohou bolo zistiť, aké informácie poskytuje edX v roli *Tool Consumer* externým systémom v rolách *Tool Provider*. Dôležité sú najmä informácie o používateli 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 podsekcii. Používateľské 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 (Askalot), ktoré vystupujú ako *Tool Providers*. V rámci úlohy je potrebné implementovať *OAuth* autentifikáciu do systému Askalot, ktorú LTI komponent používa na zabezpečenie 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 a rolu.

Testovanie

Testovanie registrácie používateľa do Askalot databázy pomocou LTI prebehlo manuálne.

1.2. Pohľad na lekcii (angl. unit)

Analýza

Edukačný materiál v systéme edX sa nachádza v lekciiach-unitoch. LTI komponent je možné k týmto lekciiam priložiť. Problém bol v tom, že pohľad na otázky z globálneho pohľadu obsahoval priveľa informácií, 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 päťou,
- taby s filtrami otázok,
- zbytočne veľké okraje,
- gravatar,
- formulár pre novú otázku sa skrýva pod tlačidlo,
- hlavná kategória už nie je zobrazovaná.

Samotný pohľad na lekcii sa zobrazuje po *LTI requeste*, kde na základe LTI komponent ID vieme nájsť príslušný zoznam otázok a tento zoznam zobrazíť. Okrem toho bolo potrebné všetky externé linky (okrem zoznamu otázok a formuláru novej otázky) presmerovať na globálny pohľad. Išlo o pohľady:

- profil používateľa
- filtrovanie podľa tagov
- markdown pomoc

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 riešenie 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 prispôbovať veľkosť vloženého Askalotu v edX vo vloženom rámci (angl. iframe). Použili sme knižnicu [iframe-resizer](#)¹, 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*.

¹ <https://github.com/davidjbradshaw/iframe-resizer>

2. Integrácia obsahu s edX

2.1. Štruktúra edX

Analýza

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

Návrh a implementácia

Pre ukladanie štruktúry kurzov sa používajú hierarchické kategórie. Prvá úroveň je kurz. Ten pozostáva z viacerých častí - sekcií, ktoré tvoria druhú úroveň. Tretiu úroveň tvoria podsekcie (napr. lekcía v týždni) a poslednú úroveň lekcie (menšia časť jednej podsekcie). Každá lekcía 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 názov a id:

- kurzu,
- sekcie,
- podsekcie,
- lekcie.

Taktiež sa získa LTI id a HTML obsah lekcie. Skript sa spustí pri načítaní lekcie. Vytvorenie stromovej štruktúry v databáze prebieha nasledovne:

1. Ak sa v databáze nájde kategória so zisteným LTI, ktorá nemá rodiča, tak sa vytvorí celá štruktúra odhora nadol (ak niektorá kategória vo vytvárannej štruktúre už existuje, tak sa nevytvorí nová, ale použije sa tá existujúca) a kategória sa pod ňu zavesí.
2. Ak taká kategória v databáze nie je vôbec, vytvorí sa a rovnako sa pre ňu vytvorí štruktúra, pod ktorú sa následne zavesí.
3. Ak sa kategória so zisteným LTI v databáze nachádza a má aj celú štruktúru, nedeje sa nič.
4. V prípade, že komponentov je v lekcií 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 funkcionálna bola otestovaná manuálne, priamo cez prehliadač.

2.3. Univerzálne kategórie

Analýza

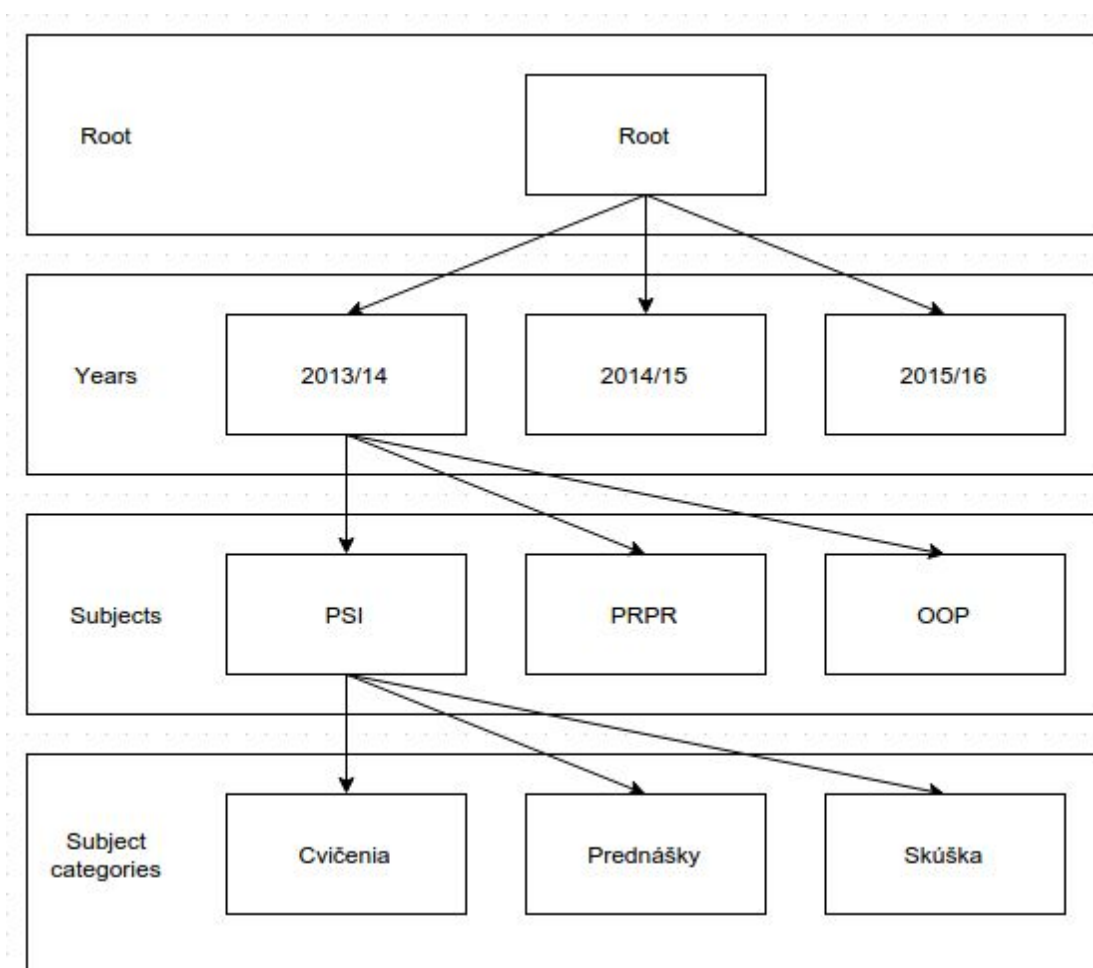
Bolo potrebné upraviť štruktúru kategórií tak, aby sa dala použiť v rámci Askalotu aj edX.

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 databázovú techniku *nested set*. Tak isto je potrebné zmigrovať aktuálnu štruktúru Askalot-u do hierarchie. Hierarchia sa skladá z koreňa, ktorého deťmi sú školské roky, pod nimi sú predmety, 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 funkcionálna otestovaná manuálne, priamo cez prehliadač. Automatické testy budú napísané počas implementácie ďalších úloh.

2.4. Administrácia kurzu

Analýza

Administrátor alebo správca systému chce upravovať kategórie. Potrebuje meniť názov, vytvárať nové kategórie a nastavovať, do ktorých kategórií je možné vložiť otázku a ktoré kategórie sú zdieľané. Zdieľanie znamená, že v danej kategórii sa zobrazujú otázky zo starších kategórií, 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 úrovni. 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 úplné 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 (zavisi od `shared` atribútu), `all_related_questions` - všetky aj predošlé otázky k celému podstromu (zavisi 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 označovať `shared/askable` vlastnosť. `Ctrl+Click` označí/odznačí celý podstrom. V rámci úlohy boli opravené aj výkonnostné (angl. performance) problémy pri zobrazovaní administrácie a načítavni 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 prvotná uvedená funkcionálna otestovaná manuálne, priamo cez prehliadač. Ďalšie čiastkové funkcionality boli pokryté testami a používateľské rozhranie spolu s výkonnosťou boli otestované manuálne.

2.5. Kontext

Analýza

Doteraz bol obsah v Askalote spoločný pre všetky predmety, čo bol vyhovujúci stav, avšak v systéme edX musí byť obsah odlišných kurzov navzájom izolovaný. Vďaka hierarchickým kategóriám išlo o pomerne jednoduchý princíp => stačí zobrazovať obsah, ktorý súvisí s kategóriou (a jej podkategóriami), v ktorej kontexte sa používateľ práve nachádza.

Návrh a implementácia

Pôvodný návrh predpokladal ukladanie kontextu do session, avšak tento návrh sa ukázal ako nedostatočne robustný a preto sa prešlo k zobrazovaniu kontextu v URL adrese. Kontext v URL sa zobrazuje len pre prostredie *MOOC* nakoľko v *University* je kontext spoločný pre všetkých

používateľov. Kontext reprezentuje v URL prvý parameter (askalot.fiit.stuba.sk/edx/{KONTEXT}) a v prípade, že tento parameter nie je vyplnený, tak sa automaticky identifikuje a doplní.

Identifikácia kontextu spočíva v nájdení kurzu, ku ktorému je používateľ priradený. Ak používateľ nemá priradený kurz, použije sa kontext s názvom *default*. Identifikácia kontextu prebieha prostredníctvom metódy *before_action* v kontrolery *Shared::ApplicationController*.

Do jednotlivých dátových modelov bol pridaný *scope in_context*, ktorého parametrom je ID kategórie kontextu. Dôležitým prvkom nášho prístupu je zachytenie možnosti zdieľania obsahu medzi verziami kurzu. Tejto funkcionalite sa venujeme v nasledujúcej kapitole.

Testovanie

Nakoľko *scope in_context* ovplyvňuje takmer celú aplikáciu slúžili už existujúce testy ako validácie implementácie. Okrem toho sme vytvorili niekoľko testov, ktoré validovali zmenu kontextu a s tým súvisiace filtrovanie obsahu.

2.6. Zdieľanie obsahu medzi verziami kurzu

Analýza

Mnoho kurzov v systéme edX sa pravidelne opakuje a vytvárajú sa tak nové verzie rovnakého kurzu. V rámci opakujúcich sa kurzov je často rovnaký obsah, a teda môžeme očakávať, že môžu byť v aktuálnom kurze kladené rovnaké alebo podobné otázky, ako v jeho predchádzajúcich verziách. Preto je vhodné pridať funkcionalitu zdieľania otázok naprieč rôznymi verziami kurzu. Zároveň treba brať do úvahy, že štruktúra kurzu sa môže zmeniť a inštruktor môže lekcie ľubovoľne premiestňovať v sekciách a podsekciách.

Návrh a implementácia

Navrhli sme, že otázky sa budú zdieľať podľa priradených lekcii (angl. unit), to znamená, že ak sa vyskytne v novom kurze lekcia, ktorá už v niektorej predchádzajúcej lekcii bola použitá, jej otázky budú dostupné aj v aktuálnom kurze.

V Askalote sme túto funkcionalitu navrhli všeobecne, každá kategória má okrem svojho identifikátora priradené aj *uuid*. V prípade, že ide o rovnakú kategóriu, iba v napríklad inej verzii kurzu, má toto *uuid* rovnaké. Zároveň každá kategória má stĺpec *shared*, ktorý hovorí o tom, či v danej kategórii (v edX je to lekcia) sú viditeľné aj otázky z ostatných verzií.

Kvôli zdieľaniu sa otázka môže nachádzať vo viacerých kategóriách (tieto vzťahy sa pamätajú v tabuľke *CategoryQuestions*), avšak v modeli *Question* si pamätá, v ktorej kategórii vznikla, aby sa pri prípadnom vypnutí zdieľania zobrazovala na správnom mieste. Pri zobrazovaní celého obsahu v kontexte (napríklad otázok) je potrebné odlišovať, či je kategória zdieľaná a uvažuje sa aj obsah z jej ostatných verzií, alebo nie je zdieľaná a uvažuje sa obsah iba z nej.

Testovanie

Funkcionalita bola otestovaná pomocou 6 nových testov, nachádzajúcich sa v adresári *spec/shared/models*, ktoré testujú najmä správne priradenie otázok do kategórií pri rôznych nastaveniach ich zdieľania.

2.7. Notifikácia v rozhraní

Analýza

Aby sme používateľovi v edX poskytli lepší prehľad o aktuálnych notifikáciách, bolo potrebné pridať číslo s počtom neprečítaných notifikácií v Askalote do hlavného menu v edX. Keďže používateľ sa môže pohybovať po rôznych lekciách, nie vždy sledujú globálnu obrazovku askalotu. Preto je dôležité mu počet neprečítaných notifikácií zobrazit' na každej stránke, aby tak nestratil prehľad o dianí a novinkách v systéme.

Návrh a implementácia

Pri načítaní každej stránky sa do vyhradeného meta tagu v HTML uloží počet neprečítaných notifikácií. Po pripravení celého obsahu stránky sa na strane Askalotu pošle pomocou *iframe-resizer* knižnice správa do nadradenej edX aplikácie.

V edX už máme vložený vlastný JavaScript kód, ktorý je schopný komunikovať a prijímať *iframe-resizer* správy. Pokiaľ detekuje správu označenú ako počet neprečítaných notifikácií, zobrazí ich počet v hlavnom menu v edX hneď vedľa odkazu do globálneho Askalotu v edX.

Tento počet v červenom krúžku sa zobrazí iba vtedy, pokiaľ je počet notifikácií väčší ako nula. Pokiaľ je počet správ väčší ako 99, zobrazí sa iba 99+ namiesto reálneho čísla z priestorých dôvodov. Formátovanie čísla je urobené výhradne pomocou priameho štýlovania elementu, keďže sa jedná o edX kde nemáme prístup modifikovať CSS štýlovacie súbory a nevkladáme tam žiadne vlastné.

Testovanie

Funkcionalita bola otestovaná pomocou manuálne priamo v testovacom prostredí edX. Keďže sa jedná o zásah do zobrazenia v edX, nie je možné pre túto funkcionality napísať automatický test.

3. Integrácia používateľov z edX

3.1. Roly používateľov

Analýza

Vzhľadom na účel edX, majú jednotliví používatelia systému priradené rôzne roly. Rozoznávame následné úrovne používateľov (zoradené podľa množstva priradených práv od najvyššieho stupňa po najnižší):

administrator

- správca celého systému, môže meniť všetky kategórie, práva používateľov a všetky ostatné informácie,

teacher

- je správca (administrátor, organizátor) daného kurzu - môže pridávať *teacher_assistant* do kurzu, táto rola sa používa pri kategóriách používaných učiteľom,

teacher_assistant

- je podobne, ako *teacher* jedným z organizátorom kurzu, táto rola sa rovnako používa sa pri kategóriách podporovaných učiteľom, ale zároveň nemá pridelené práva na úpravu kurzu,

student

- je to bežný študent, ktorý má zapísaný daný kurz.

Návrh a implementácia

Definovanie rolí používateľov v edX kurze umožňuje určiť dve relevantné roly: *Admin* a *Staff*. Počas implementácie sme však zistili, že v LTI requeste sú tieto roly inak nazvané:

Admin je označovaný ako *Instructor* a *Staff* ako *Administrator*. Z tohto dôvodu sme rozhodli pre nasledujúce mapovanie rolí:

- *Instructor* - *teacher*
- *Administrator* - *teacher_assistant*

Rola *teacher_assistant* v Askalote má rovnaký prístup ako rola *teacher*, avšak priradiť rolu *teacher_assistant* môže v administrácii aj inštruktor kurzu.

3.2. Vytvorenie účtu, získanie údajov, prihlasovanie

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

Testovanie

Testovanie prebehlo manuálne. Počas testovania sme skúsili viaceré scenáre kde boli LTI autentifikačné údaje nastavené správne a takisto aj kde nastavené neboli. V takomto prípade informujeme používateľa o nesprávnom nastavení a zmene údajov. Po správnom nastavení LTI kľúča a tajného hesla je používateľ automaticky zaregistrovaný a prihlásený do systému.

3.3. 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. To, že po presmerovaní na lekciu je potrebné presmerovať späť riešime parametrom v URL adrese, ktorého prítomnosť Askalot skontroluje a v ak je prítomný, tak presmeruje naspäť.

Implementácia

Riešenie sme implementovali ako *before_filter*, ktorý sa vykonáva pred volaním všetkých akcií. Tento filter nie je používaný v *controlleri*, 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 lekcii, 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. Po načítaní javascriptu v Askalote sa skontroluje, či je prítomný parameter v URL značiaci, že používateľ sa sem dostal kvôli neprihlásenému stavu v global view. Ak je prítomný, tak pomocou komunikácie cez knižnicu *iframe resizer* sa okno s edX-om presmeruje späť 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 scenári 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. Následné presmerovania späť fungovali korektne až po tom, čo sme na komunikáciu využili komunikačné rozhranie poskytnuté knižnicou *iframe resizer*.