

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Priebežné overovanie prípravy študentov na cvičeniach [WebTest]

Osičky

Dokumentácia k dielu

Vedúci tímu: Ing. Branislav Steinmüller

Členovia tímu: Bc. Lukáš Csóka, Bc. Martin Dekan, Bc. Michal Farkaš, Bc. Silvia Macejková, Bc. Roman
Pikna, Bc. Pavel Sluka

Školský rok: 2015/2016

Obsah

Obsah	2
1 Úvod	4
2 Slovník pojmov	5
3 Globálne ciele pre zimný semester	7
4 Globálne ciele pre letný semester	8
5 Celkový pohľad	9
5.1 Naše zmeny	10
5.2 Prototyp	16
6 Moduly systému	18
6.1 Prototyp	18
6.1.1 Modul administrátor	18
6.1.1.1 Pridávanie testových otázok	18
6.1.2 Modul vyučujúci	19
6.1.2.1 Učiteľ a admin nemôžu písať test	19
6.1.2.2 Cvičiaci zobrazí prístupový kód aktuálneho testu	20
6.1.2.3 Cvičiaci zobrazí anonymizované odpovede študentov	20
6.1.2.4 Cvičiaci po ukončení testu môže zobrazíť základnú štatistiku odpovedí	21
6.1.2.5 Prihlasovanie cvičiacich cez konfiguráciu	22
6.1.3 Modul študent	23
6.1.3.1 Študent sa prihlási cez AIS LDAP	23
6.1.3.2 Študent odpovedá na otázky s otvorenou odpoveďou	24
6.1.3.3 Študent odpovedá na otázky s viacerými možnosťami	24
6.1.3.4 Študent si zobrazí test s otázkami	25
6.1.3.5 Systém neumožní nič okrem povolených akcií	26
6.1.3.6 Vynechanie prázdnych odpovedí multichoice otázok pri generovaní testu a zobrazení štatistiky	27
6.2 AlefTNG	28

6.2.1	Modul Administrátor	28
6.2.1.1	Administrátor pri štarte systému nahrá súbor s cvičeniami	32
6.2.1.2	Administrátor vytvorí termíny cvičení počas semestra	33
6.2.1.3	Administrátor vloží termíny cvičení a otázky do systému	35
6.2.2	Modul vyučujúci	38
6.2.2.1	Cvičiaci zobrazí test na cvičení	43
6.2.2.2	Cvičiaci ukončí test na cvičení	43
6.2.2.3	Cvičiaci si zobrazí zoznam svojich termínov cvičení.....	44
6.2.2.4	Učiteľ a admin nemôžu písať test	45
6.2.2.5	Cvičiaci skontroluje dochádzku študentov na cvičení.....	45
6.2.2.6	Učiteľ na domovskej stránke môže spustiť test	46
6.2.3	Modul študent	47
6.2.3.1	Študent môže prísť na hociktoré cvičenie.....	49
6.2.3.2	Študent po odoslaní odpovede nemôže test písať druhýkrát	50
6.2.3.3	Študent odošle svoju odpoveď na test	51
6.2.3.4	Študent odpovedá na obrázkovú otázku	51
6.2.3.5	Študent odpovedá na otvorené otázky.....	52
6.2.3.6	Študent odpovedá na výberové otázky	52
6.2.3.7	Študent si v rámci testu zobrazí pridelené otázky	53
6.2.3.8	Študent si zobrazí aktuálny test	54
6.2.3.9	Všetci študenti dostanú špeciálnu zhodnú otázku alebo zhodné otázky na teste	54
6.2.3.10	Študent dostane otázku podľa zadaného kľúča.....	55
7	Používanie prototypu – testovanie reálnymi používateľmi	59

1 Úvod

Overovanie vedomostí bolo vždy výzvou nielen pre študentov, ale aj pre vyučujúcich. Vytvorenie aplikácie, ktorá by bola zaujímavá a motivujúca zároveň je veľkou výzvou. Sme presvedčení, že práve my dokážeme priniesť riešenie tohto problému. Motiváciou k vytvoreniu tejto aplikácie je pomoc študentom so štúdiom a vylepšenie súčasného systému priebežného overovania vedomostí. S rozmachom mobilných zariadení sa zväčšujú možnosti využitia mobilných zariadení. Preto sme sa rozhodli priniesť webovú aplikáciu optimalizovanú najmä pre mobilné zariadenia. Priebežné overovanie vedomostí má za následok priebežnejšie učenie a v konečnom dôsledku pre študentov menej stresu na zápočtových písomkách, pri zadaniach a v neposlednom rade aj na skúškach.

Prvým predmetom, ktorý bude testovať túto aplikáciu sú Operačné systémy. Na tomto predmete sú overované vedomosti pomocou nášho prototypu a študenti môžu získať bonusové body. Aplikáciu sme sa rozhodli vytvoriť ako webovú aplikáciu so schopnosťou prispôbiť sa veľkosti displeja a jeho rozlíšeniu.

Aplikácia je vytváraná pre použitie pri výučbe akéhokoľvek predmetu. Aplikácia bude schopná vyhodnocovať testové otázky. Otázky s odpoveďami, ktoré majú ľubovoľnú dĺžku odpovede bude vyhodnocovať cvičiaci.

2 Slovník pojmov

Výraz	Vysvetlenie
Alef	Skratka pre Adaptive LEarning Framework
AngularJS	Je klientský javascript framework
API	Skratka pre <i>Application programming interface</i>
Node.js	Je serverový javascript framework
front-end	Časť aplikácie viditeľná bežným návštevníkom
AlefTNG	Skratka pre Adaptive LEarning Framework The Next Generation
OS	Skratka pre operačný systém
ERB	Skratka pre Embedded RuBy
Hash funkcia	Funkcia, ktorá prevádza vstupný reťazec znakov na iný
Hash	Reťazec znakov vytvorený hash funkciou
Chybová hláška 404	Stránka nebola nájdená
csv	Comma separated values
Google chart	Nástroj od spoločnosti Google slúžiaci na vykresľovanie grafov na web stránkach
LDAP	Z anglického <i>Lightweight Directory Access Protocol</i> je protokol na ukladanie a prístup k dátam na adresárovom serveri. Jeho súčasťou je autentifikácia klienta.
import	Prevod elektronicke uložených dát z do aktuálneho kódu, formátu programu
CSV	<i>Comma-separated values</i> je jednoduchý súborový formát vo forme čistého určený na ukladanie tabuľkových dát
Rake	Rake je Ruby utilita (Ruby náhrada za Unix make) najčastejšie používaná na administrátorské úlohy. Jej vhodnosť vyplýva najmä z možnosti pristupovať k jednotlivým častiam Ruby on Rails aplikácie (model, dáta, ai.).
Obrázková úloha	Úloha, ktorej zadanie obsahuje obrázkov

Otvorená otázka	Otázka na ktorú sa odpovedá voľným textom, opravuje sa manuálne
Recommender	Odporúčajúci algoritmus, v našej aplikácii vyberá otázky, ktoré dostane študent
Highcharts	Je JavaScript knižnica určená na kreslenie interaktívnych grafov vo webových aplikáciách.
JSON	skratka pre JavaScript Object Notation, je to formát pre výmenu dát vychádzajúci z programovacieho jazyka JavaScript.
QR kód	dvojrozmerný čiarový kód, ktorý je navrhnutý s ohľadom na rýchle dekodovanie
Tex	Jazyk určený na tvorbu dokumentov
KaTeX	je JavaScript knižnica pre vykresľovanie matematických zápisov vo formáte TeX

3 Globálne ciele pre zimný semester

V zimnom semestri by sme chceli mať plne funkčný prototyp. Ďalším identifikovaným cieľom bolo získanie štatistík z prototypu. Študenti vďaka prototypu môžu získavať bonusové body, ktoré môžu prispieť k lepšiemu hodnoteniu na predmete. Takmer každý predmet používa na overovanie vedomostí iný systém. Napríklad na PSI je to AIS a Alef, na TZI je to moodle. Zároveň sme z analýzy AlefTNG identifikovali hlavný cieľ tvorby systému použiteľného na akomkoľvek predmete. Cieľom bude pokryť čo najväčšie množstvo typov otázok. Preto sme sa rozhodli systém AlefTNG rozšíriť pre použitie všetkých známych typov otázok.

Mobilné zariadenia sú výzvou pre takmer každý systém. Takmer každé mobilné zariadenie má iné rozlíšenie, veľkosť obrazovky, systém a prehliadač. Na trhu existuje veľké množstvo zariadení, prehliadačov a v neposlednom rade je vidieť aj veľký rozptyl používaných verzií jednotlivých OS.

Najhlavnejším cieľom je rozvoj systému AlefTNG. Ďalším cieľom je vytvorenie rozhrania pre administrátora a pre vyučujúcich.

Ďalšou výzvou bude rozvoj schopností pre prácu v tíme. Zlepšenie tímovej komunikácie a získanie nových poznatkov pre vedenie tímu a moderovanie stretnutí. Ďalším vytýčeným cieľom je zlepšenie programátorských schopností.

4 Globálne ciele pre letný semester

Počas tímových stretnutí sme identifikovali nasledujúce ciele. Pre letný semester sa sústreďíme na vylepšenie vzhľadu aplikácie. Zároveň sa venujeme zlepšeniu ovládania a používateľskému zážitku. Na overenie používateľského zážitku opäť používame testovanie reálnymi používateľmi. Keďže je náš projekt používaný na mnohých predmetoch, ktoré majú v otázkach vzorce a príklady tak pridáme aj zobrazenie vzorcov zapísaných v štandardných formátoch ako je LaTeX. Zobrazenie testov v administrácií bude možné aj pre cvičiacich. Testovanie používateľmi, ktoré bolo aj v zimnom semestri sa odohráva aj v letnom semestri. V letnom semestri testujeme na viacerých predmetoch naraz. Testovanie nášho produktu sa stáva automatizovaným. Tento krát je výzvou testovanie s veľkým množstvom používateľov na prednáškach predmetu Analýza a zložitosť algoritmov. Zároveň ako vylepšenie projektu a produktivity v letnom semestri začíname používať Continuous integration kde automaticky testuje novo nasadená verzia. Aplikáciu stále koncipujeme ako ľahko použiteľnú aplikáciu s jednoduchým používateľským rozhraním, ktoré je malou záťažou pre mobilné zariadenia. Pôvodný systém bol stále v stave vývoja a preto dokončíme alebo opravíme rozpracované časti systému. Príkladom dopracovania funkcie je odosielanie spätnej väzby.

Prioritami pre letný semester sú nasledujúce body. Sú zoradené od najväčšej priority po najmenšiu

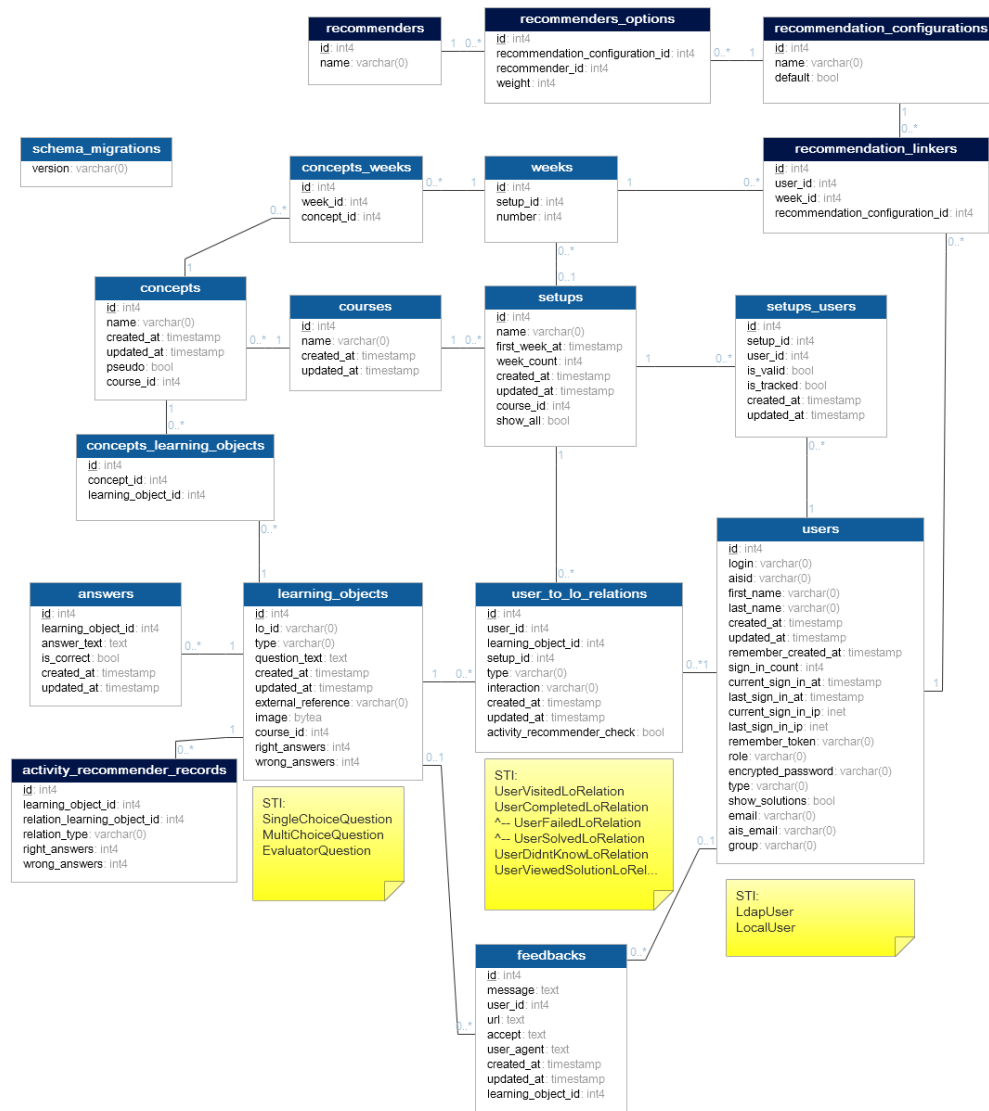
1. Vylepšenie funkčnosti, opravy a zlepšovanie použiteľnosti
2. rozvoj aplikácie a rozširovanie funkčností
3. Testovanie na veľkom množstve predmetov a s tým spojené získavanie dát
4. Automatizácia procesu testovania a automatizácia nasadenia

5 Celkový pohľad

Dostali sme systém AleFTNG, ktorý je nástupcom známeho systému Alef. Tento systém je vytvorený v známom frameworku Ruby on Rails.

Ruby on rails využíva architektonický štýl model-view-controller. Dátový model, logika a používateľské rozhranie sú tým pádom od seba oddelené. Grafické prostredie je vytvorené pomocou ERB. Použitá databáza je postgresql. Boli nám poskytnuté informácie od tvorcov.

Pri analýze sme zistili nutnosť rozšírenia dátového modelu, ktorý je už dosť komplexný a poskytuje nám mnohé možnosti delenia otázok do týždňov semestra, rôzne druhy testových otázok a vyhodnocovanie správnosti odpovedí. Dátový model odráža štruktúru tried modelu.



Obrázok 1: Pôvodný dátový model

AlefTNG môžeme rozdeliť na dve časti. A to backend a frontend. Backend je serverová časť kde beží Ruby on Rails server. A frontend, ktorý je tvorený pomocou ERB. Štýl frontendu, ktorý bol vytvorený ešte pred tým ako sme tento systém obdržali.

Aplikácia umožňuje dva typy prihlasovania. Prvý je lokálnymi účtami a druhý je použitím LDAP. LDAP poskytuje našej aplikácii nutné informácie pre overenie používateľa. Implementované je prepínanie na záložný LDAP server v prípade, že nastane výpadok hlavného LDAP servera.

5.1 Naše zmeny

Našimi zmenami majú byť rozšírenia dátového modelu pre prezeranie testov vyučujúcimi, pridanie rôznych druhov testových otázok, rozdelenie do týždňov. Tak isto sme mysleli aj na posuny týždňov a možnosti administrácie testov. V administrácii testov vystupuje vyučujúci (cvičiaci) a administrátor. Tieto dve roly sme sa rozhodli oddeliť, kvôli dostupnosti globálnych úprav týždňov a ďalším potencionálnym problémom.

Študent môže odpovedať na otázky s ľubovoľnou dĺžkou odpovede, odpovedať na obrázkové zadanie alebo odpovede s viacerými možnosťami. Študent môže test absolvovať len jedenkrát. Študent môže prísť aj na iné cvičenie ako je jemu určené a týmto spôsobom si nahradiť zameškaný test. Študent môže pracovať v dvoch režimoch. Prvý je učiaci režim kde vidí zvýraznené správne odpovede a neodpovedá na otázky. Aby študent mohol absolvovať test počas nahradeného cvičenia, je umožnené aby test písal na cvičení v inom čase. V testovom móde označuje správne odpovede a systém mu poskytuje spätnú väzbu. Študent každý týždeň dostáva otázky z nového okruhu. Naďalej sa môže učiť aj otázky z predchádzajúcich týždňov. Študentom sme dali dve možnosti ako vstúpiť do testu opísaním krátkeho vstupného kódu alebo skenovaním QR kódu. Študenti majú možnosť pozrieť si hodnotenie testu.

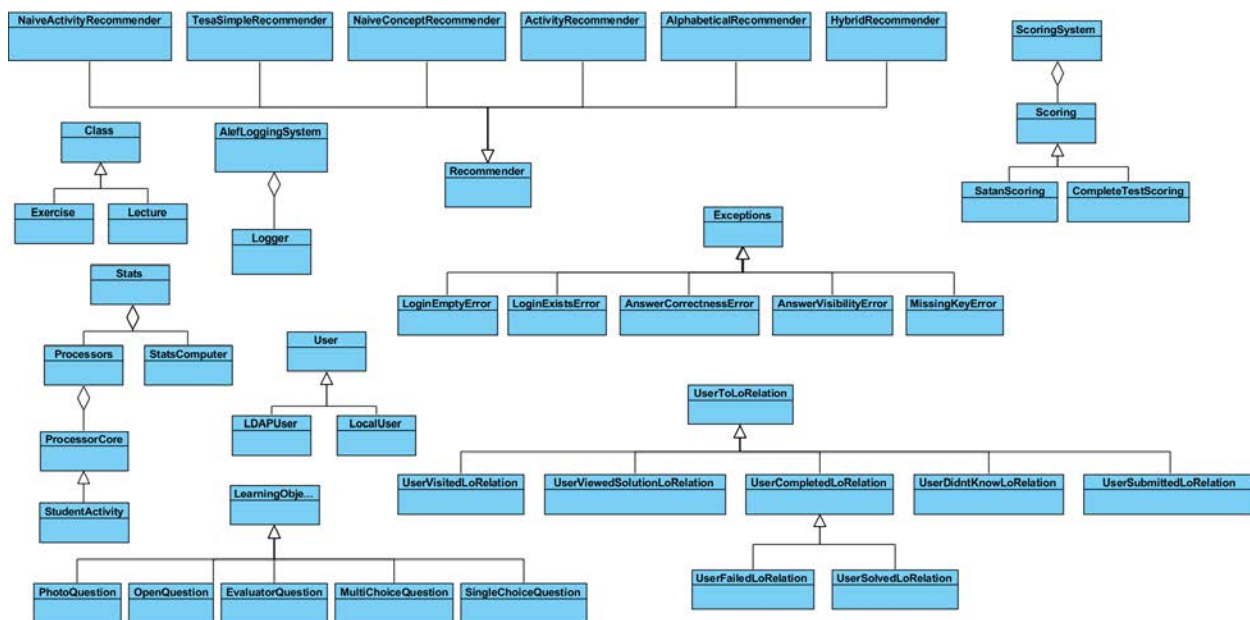
Administrátori majú možnosť nahráť testy prostredníctvom súborov vo formáte CSV. Administrátori majú práva pre pridávanie, úpravu a odoberanie termínov. Zároveň budú mať možnosť pridávať, odoberať a upravovať študentov a cvičiacich (resp. meniť ich oprávnenia pre jednotlivé cvičenia). Otázky môžu obsahovať obrázky, vzorce a text. Dôležitým nastavením je spôsob akým sa pridelujú body napríklad bod sa pridelí, ak boli všetky odpovede správne, body za každú správnu odpoveď, alebo strhávanie bodov za nesprávne odpovede, pol bodu za správnu odpoveď, atď. Administrátor môže nastaviť globálny čas, po ktorom sa test automaticky ukončí a uzavrie. Administrátor má možnosť nastaviť aj čas po, ktorom sa študentom zobrazia výsledky testu. Administrátor má možnosť nastaviť aj špeciálne otázky, ktoré sú zhodné pre všetkých študentov behom zvoleného obdobia.

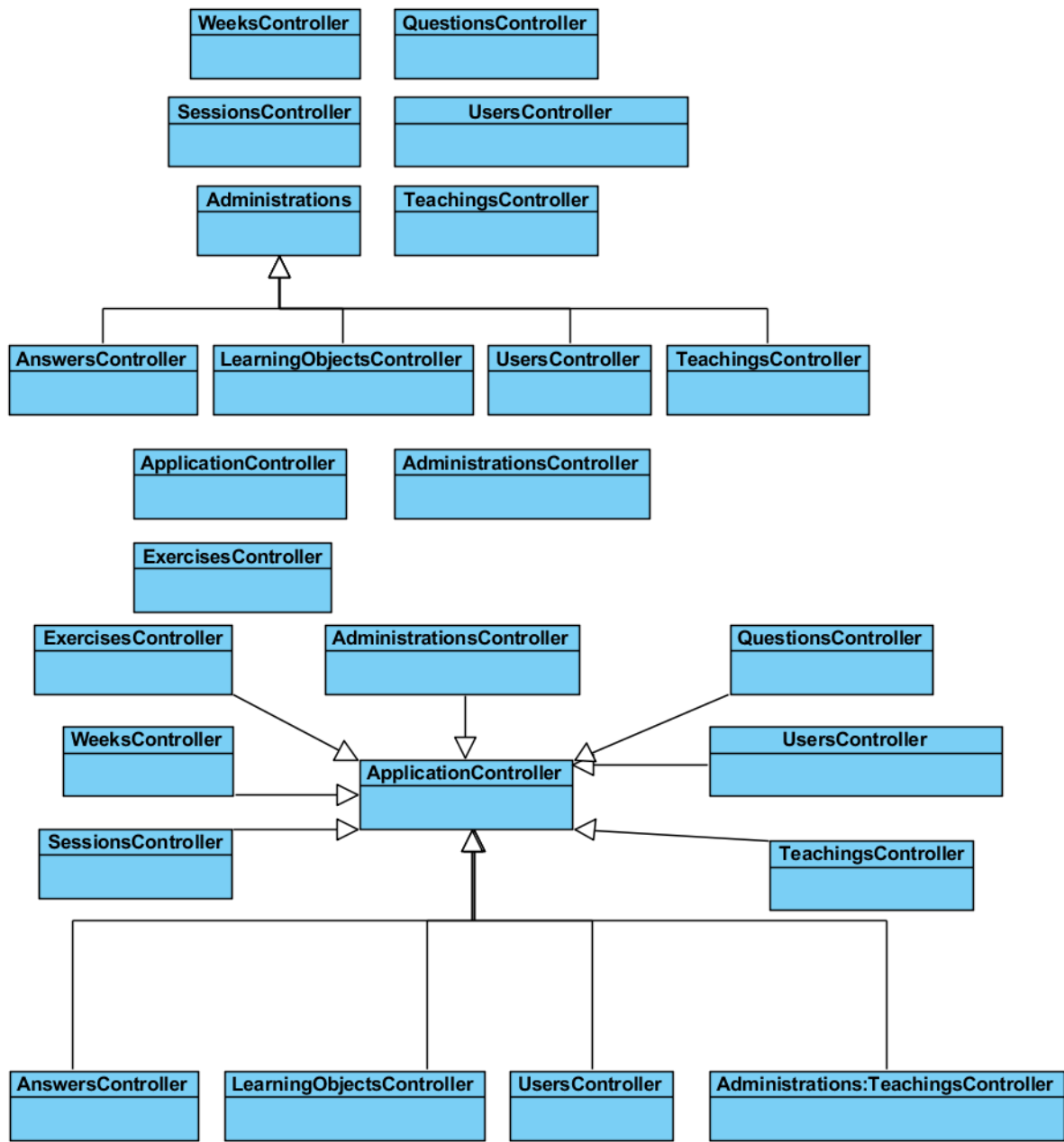
Vyučujúci má okrem ovládania prístupu k testu možnosť aj zobrazíť si termíny svojich cvičení v prehľadnej tabuľke. Zavedený bol systém prístupu pomocou prístupových hesiel. Tieto heslá sa uverejnia na cvičení alebo prednáške a umožnia absolvovanie testu. Dĺžku prístupu k testu ovláda vyučujúci. Vyučujúci vygeneruje prístupový kód, otvorí prístup a následne ho aj uzavrie. Vyučujúci môžu testy len zobrazíť, ale nie absolvovať. V pôvodnej verzii bol vyučujúci braný rovnako ako študent. Po konci testu je možné zobrazíť štatistiky a to aj pre jednotlivé otázky. Požiadavkou vyučujúcich bola možnosť exportovania odpovedí. Aby sa predišlo problémom, ktoré iné systémy neriešia umožňujeme aby vyučujúci znovu otvoril test. Aby mal vyučujúci prehľad o aktuálnom stave, môže v reálnom čase vidieť počet odovzdaných testov. Ak je na cvičení alebo prednáške vykonaný test tak sa do systému zapisuje ak dochádzka. Bolo

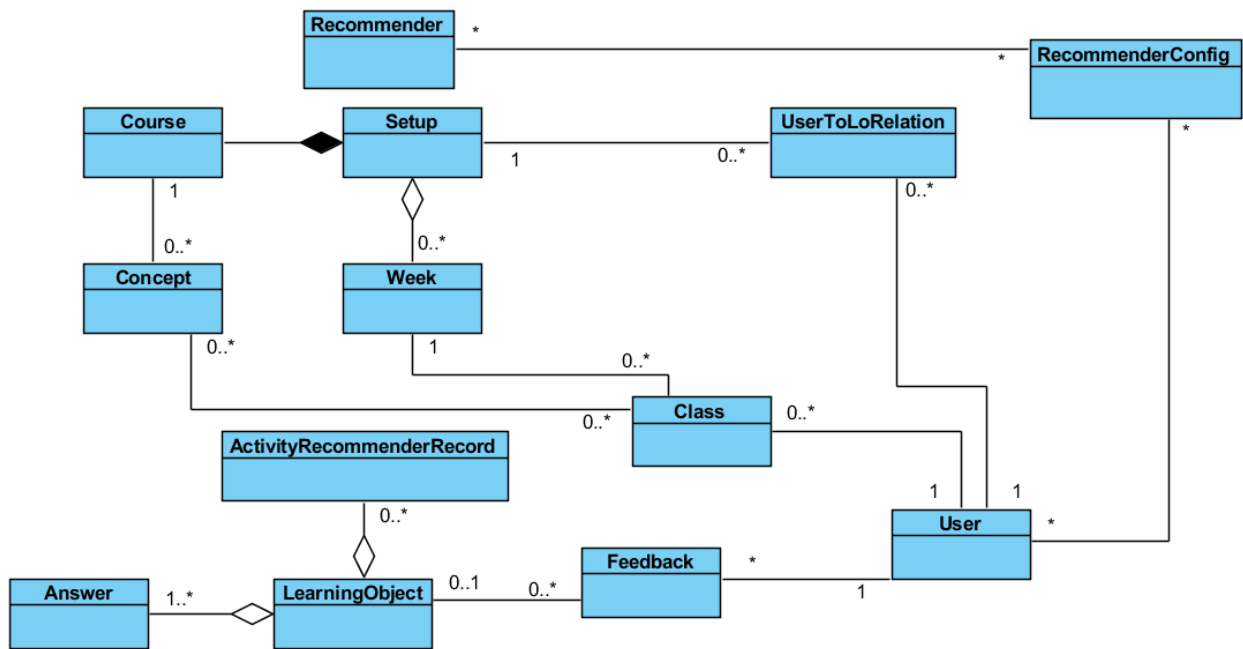
pridané aj automatické hodnotenie odpovedí na otázky s jednou správnou odpoveďou aj s viacerými správnymi odpoveďami.

Keďže pôvodný systém neobsahoval automatizované testy, doplnili sme automatizované unit testy. Automatizované testovanie zabezpečujú gemy minitest, test-unit. Testovanie je doplnené o Devise, ktorý ovláda autentifikáciu a http požiadavky. Taktiež sme vytvorili dokumentáciu a pridali tooltipy. Z pohľadu bezpečnosti boli pridané vstupné kódy na test, zakázanie možnosti viacnásobného absolvovania testu, logovanie vykonaných akcií a oddelili sme role používateľov. Ďalšími zmenami sú opravy pôvodných chýb a dopracovanie možnosti odoslania spätnej väzby. Rozhodli sme sa nepoužiť náš starý frontend vytvorený v AngularJS, ale použiť prevzatý a opraviť ho. Boli vykonané opravy šablóny a vylepšený bol aj jeho UX. Aplikácia zostala multiplatformová. Do aplikácie bol pridaný LaTeX gem mathjax, ktorý sa ukázal ako nestabilný a bol nahradený KaTeX gemom.

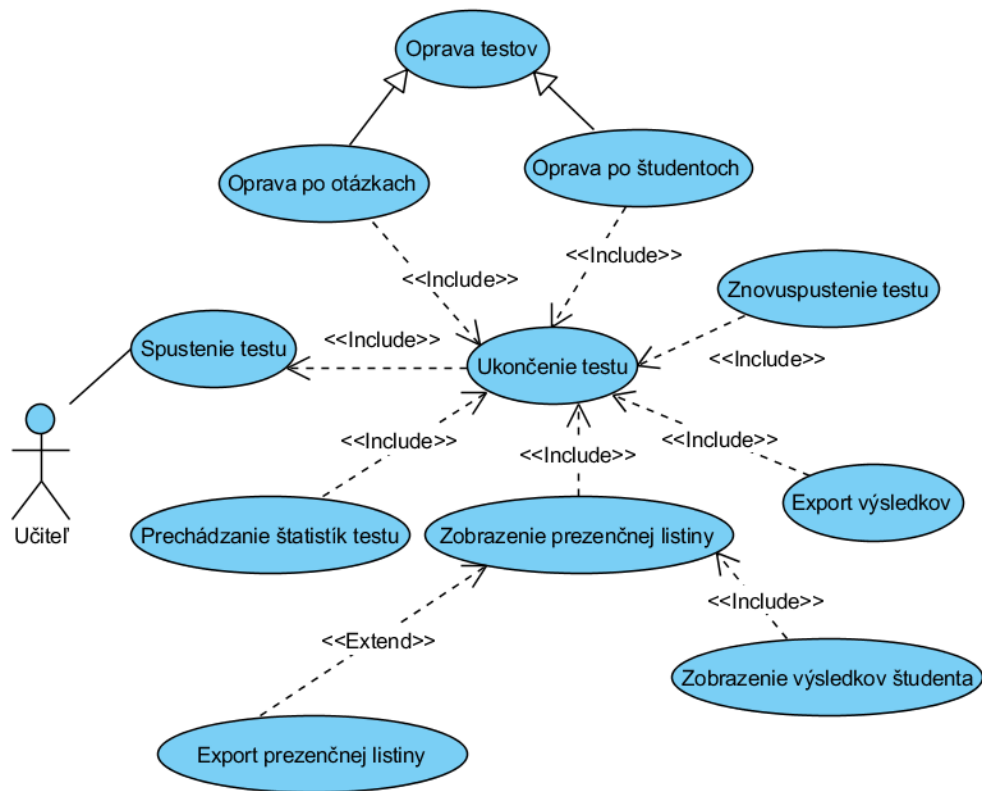
5.1.1 Diagramy tried našej aplikácie:

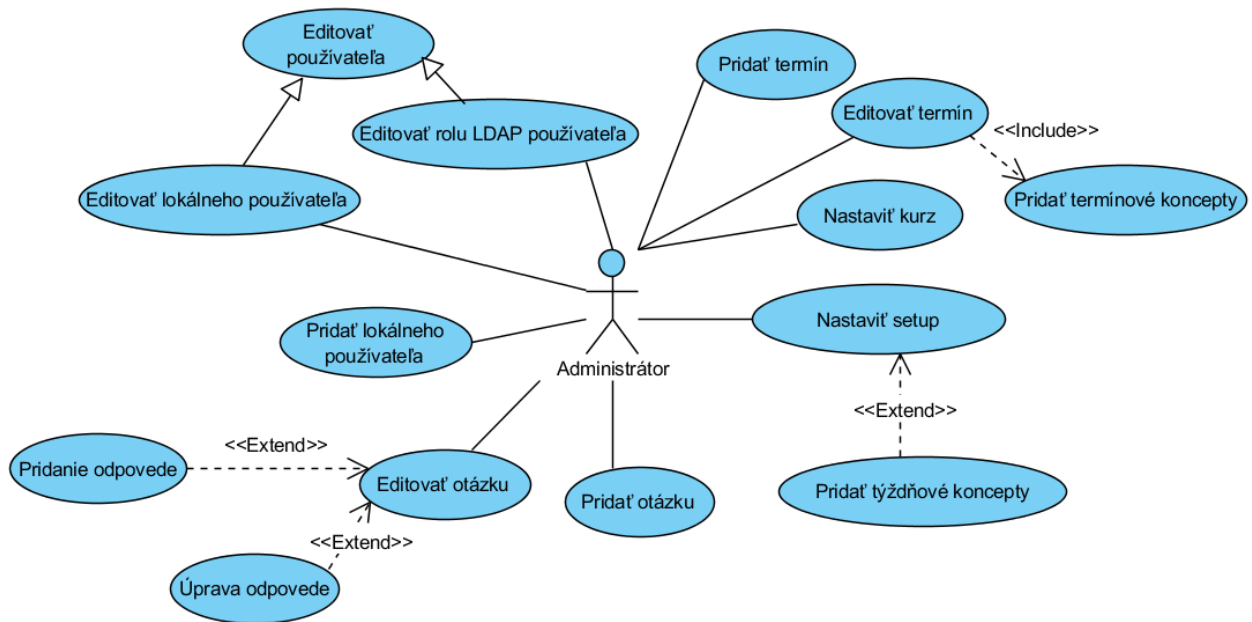
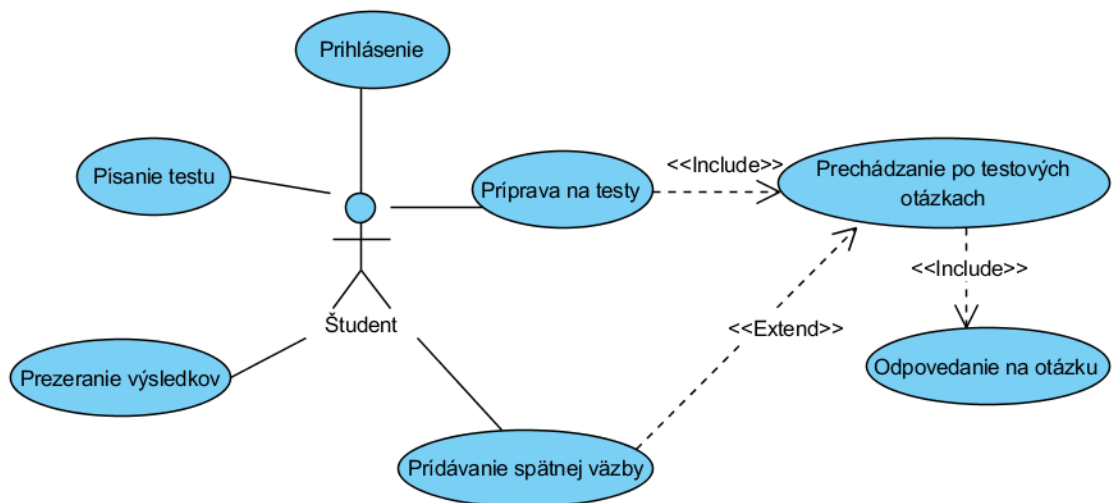




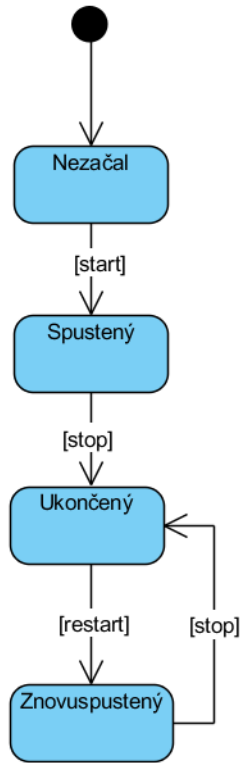


5.1.2 Diagramy prípadov použitia:





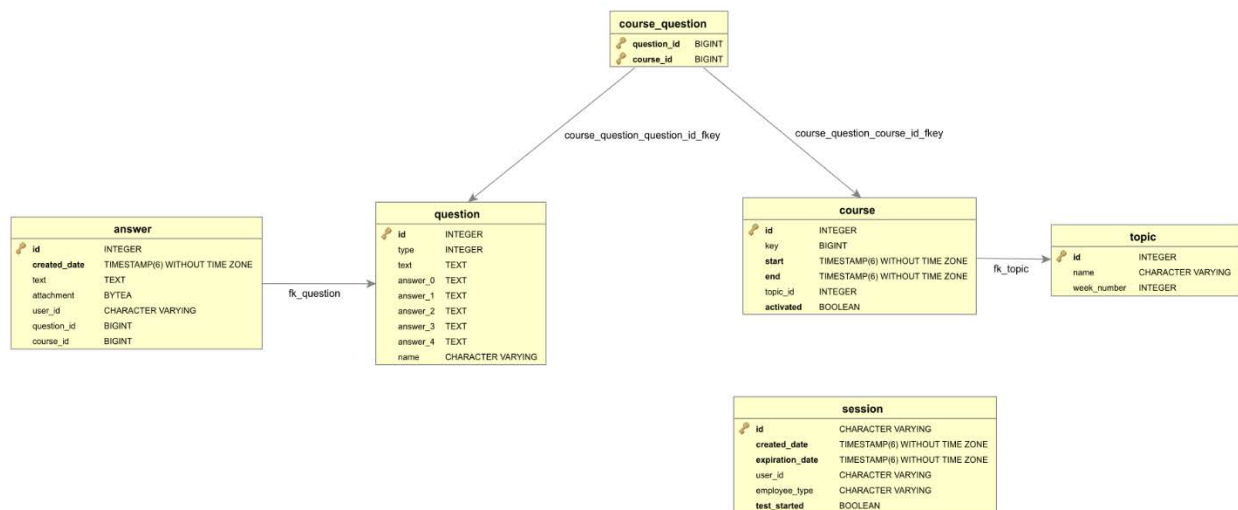
Test je jednoduchý a prechádza len niekoľkými stavmi



5.2 Prototyp

Prototyp nášho tímového projektu slúži na overenie si funkcionality a správnosti našich biznis plánov skutočnými používateľmi. Pomocou neho chceme získať primárne spätnú väzbu od študentov a učiteľov. Taktiež ním cieľme overiť, či je používateľské rozhranie dostatočne jednoduché a ľahko použiteľné. V neposlednom rade ale potrebujeme zistiť ako bude naša aplikácia reálne vplyvať na prípravu študentov, aby sme tieto získané informácie mohli použiť pri vývoji našej hlavnej aplikácie.

Ide o jednoduchšiu webovú aplikáciu, ktorá overuje znalosti a prípravu na cvičenia študentami. Skladá sa z dvoch častí – frontendu a backendu. Vo frontende sa rieši zobrazenie aplikácie. Použili sme náš framework Mobile Angular UI. Je to spojenie Angular JS s Bootstrap-om, ktoré sú upravené primárne pre potreby mobilných zariadení, ktoré považujeme za cieľové nástroje pre používanie nášho prototypu používateľmi. Backend je zostrojený pomocou Node JS s databázou PostgreSQL. Informácie medzi týmito dvoma časťami sa posielajú ako správy vo formáte JSON.



Obrázok 2: dátový model prototypu

Táto aplikácia sa tento semester aktívne používa na cvičeniach predmetu Operačné systémy na FIIT. Na tieto účely je dostupná na stránke <https://tesa.fiit.stuba.sk/os>.

V tejto aplikácii sú dve role používateľov : učiteľ a študent. Každá z nich má prístup k iným typom obrazoviek čo znamená, že majú aj iné scenáre používania. Základná obrazovka rieši prihlásenie používateľa. Bez prihlásenia sa používateľovi nezobrazia žiadne informácie a ani funkcionality stránky. Po prihlásení je používateľ presmerovaný na obrazovku, ktorá prislúcha roly prihláseného používateľa.

Ak je jeho rola učiteľ:

1. Dostane sa na obrazovku zobrazujúcu jedinečný kód testu, ktorý zverejní svojim študentom na cvičení. Samotným prihlásením učiteľa sa konkrétny test aktivuje. Táto obrazovka obsahuje aj tlačidlo na ukončenie testu. Pokiaľ sa učiteľ prihlási v iný čas ako má cvičenie, táto obrazovka mu oznámi, že nie je možné aktivovať test, keďže neprebíha cvičenie a nezobrazí sa mu ani kód testu.

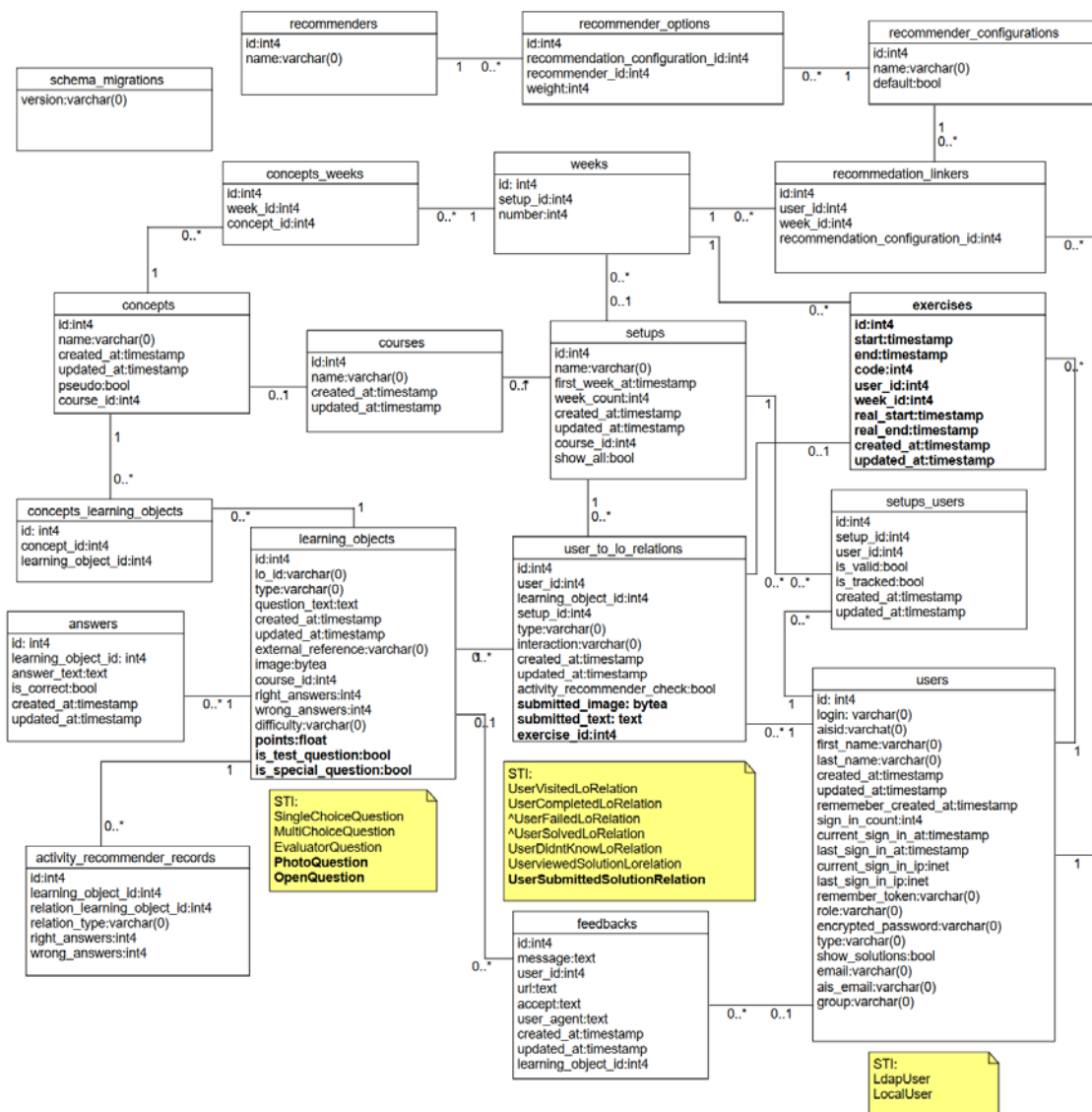
- Keď učiteľ klikne na tlačidlo pre ukončenie testu, zobrazí sa mu obrazovka zobrazujúca anonymizovanú grafovú štatistiku odpovedí študentov na všetky otázky.

Ak je jeho rola študent:

- Po prihlásení do aplikácie sa presmeruje na obrazovku na zadanie kódu testu resp. cvičenia, ktorý pre nich cvičiaci zverejnil.
- Po zadaní správneho kódu sa študent dostane na obrazovku zobrazujúcu otázky testu. Pri odosielaní odpovedí ho systém automaticky odhlási z aplikácie.

Vyššie opísané základné scenáre opisujú primárne obrazovky aplikácie. Okrem nich aplikácia obsahuje aj menu s obrazovkou so základnými informáciami nášho projektu.

Pre Porovnanie na nasledovnom obrázku je dátový model aktuálnej verzie AleFTNG.



Obrázok 3: Aktuálny dátový model systému AleFTNG pre porovnanie s dátovým modelom prototypu a pôvodného dátového modelu

6 Moduly systému

Opísané sú dva systémy. Prvý je náš prototyp, ktorý delíme do troch častí podľa používateľa, ktorý s ním pracuje. Druhý je AlefTNG, ktorý rozdelíme rovnakým spôsobom do troch častí.

6.1 Prototyp

Prototyp je jednoduchší a vhodnejší pre testovanie požiadaviek. Boli vytvorené používateľské príbehy, ktoré sa neskôr preklopiť aj do systému AlefTNG.

6.1.1 Modul administrátor

6.1.1.1 Pridávanie testových otázok

Analýza

Prototyp má za úlohu testovať študentov počas cvičenia. Administrátor predmetu v spolupráci s cvičiacimi každý týždeň pripraví otázky vo vhodnom formáte. Následne sa pomocou generátora tieto otázky vložia do databázy aj so závislosťami k jednotlivým termínom (cvičeniam).

Generátor generuje SQL dopyt do databázy, pričom vyžaduje dve csv (otázky a termíny).

Návrh

Administrátor pripraví dve csv podľa formátu, spustí generátor a dostane výsledný SQL dopyt, ktorý zadá do databázy.

Riešenie

Vytvorili sme generátor, ktorý prijíma dva súbory vo formáte CSV. Jeden súbor obsahuje otázky, druhý termíny. Generátor ako argument prijíma aj číselné poradie týždňa, pre ktorý sa generujú otázky a počet otázok pridelených jednému cvičeniu. Generátor prideliť všetkým študentom jednu špeciálnu otázku, ak sa medzi vloženými otázkami nachádza. Je označená príznakom v názve otázky.

Návod na prácu s generátorom

Generátor sa spúšťa príkazom:

```
java -jar gen.jar
```

Generátor prijíma 4 argumenty:

1. číslo týždňa
2. súbor s termínmi (cesta)
3. súbor s otázkami (cesta)
4. počet otázok prislúchajúcich jednému termínu (číslo)

Môže byť spustený napríklad takto:

```
java -jar gen.jar 5 courses.csv tema05.csv 7
```

csv súbor s termínmi má nasledovný formát, čísluje sa od 1, teda mesiac január má číslo 1:
rok;mesiac;deň;hodina

csv súbor s otázkami ma formát:

```
tema;id;otazka;odpoved_a;odpoved_b;odpoved_c;odpoved_d;odpoved_e;typ;spravne
```

Výstup generátora je súbor query.sql

Generátor predpokladá prítomnosť otázky s id (stĺpec name) v tvare steinmul-NN-01, kde NN je číslo týždňa, ktorú prideli všetkým termínom. V prípade, ak tam takto označená otázka nebude, každému termínu vygeneruje o otázku menej.

Generátor kontroluje rôzne verzie otázok. Majme napríklad otázky označene bernat-05-01a, bernat-05-01b, bernat-05-01c, generátor zvolí maximálne jednu z týchto otázok pre jeden termín.

Generátor nahradí výraz "|" výrazom "\n" v texte otázky a v odpovediach.

Testovanie

Generátor bol testovaný administrátorom predmetu pred začatím testovania študentov. Výsledné query.sql bolo nahrané do databázy, pričom sa v ňom nezistili chyby.

Z dosiaľ nezistených príčin ale niektoré platné csv súbory spôsobujú pád generátora.

6.1.2 Modul vyučujúci

6.1.2.1 Učiteľ a admin nemôžu písať test

Analýza

Aby bolo možné rozlišovať v systéme používateľské role, je potrebné niekde v systéme ukladať túto informáciu. Na základe tohto údaju je možné rozhodnúť, ktorá časť aplikácie sa používateľovi zobrazí. V kontexte našej aplikácie to chápeme tak, že študent mať možnosť iba k zadaniu kódu cvičenia a vyplneniu testu. Na druhej strane, cvičiaci (administrátor) má prístup iba k administrácii testov a štatistike.

Návrh

Počas prihlásenia sa získajú údaje o roli používateľa, na základe ktorej bude aplikácia vedieť rozhodnúť ktorú obrazovku mu následne zobrazí. Táto kontrola bude prebiehať pri každej akcii vykonanej používateľom volaním služby servera.

Riešenie

Vytvorili sme časť systému slúžiacu na rozlišovanie používateľských rolí. Počas procesu overenia prihlásenia sa získa informácia o roli používateľa, ktorá sa uloží spolu s jednoznačným identifikátorom. Pri každej interakcii s aplikáciou je tak kontrolovaný nielen jednoznačný identifikátor ale aj používateľská rola.

Testovanie

Úlohu sme testovali použitím rôznych používateľských účtov. Druhá vlna testovania prebiehala na používateľoch, ktorý prešli celým používateľským príbehom, nakoľko je toto jeden zo základných stavebných prvkov aplikácie.

6.1.2.2 Cvičiaci zobrazí prístupový kód aktuálneho testu

Analýza

Aby mohli študenti začať vyplňovať test, musia zadať kód cvičenia. Ten musí byť buď jednotný pre celý týždeň a všetci cvičiaci ho musia poznať, alebo bude unikátny a cvičiacemu sa zobrazí hneď po aktivácii testu.

Návrh

Cvičiaci sa po prihlásení ako administrátor dostane na obrazovku aktivácie testu, kde sa mu zobrazí hláška o stave aktivácie. Po aktivovaní sa zobrazí adresa aplikácie a dostatočne veľký unikátny kód cvičenia aby ho mohli študenti prečítať z projektora.

Riešenie

Po prihlásení cvičiaceho (administrátora) v čase cvičenia je presmerovaný na obrazovku aktivácie cvičenia. Tu sa odošle požiadavka, ktorá aktivuje test pre práve prebiehajúce cvičenie. Následne sa mu zobrazí hláška o aktivácii, alebo dôvode chyby pri aktivácii testu. Po úspešnej aktivácii sa na obrazovke zobrazí veľký kód cvičenia, na základe ktorého sa študentovi zobrazí test.

Testovanie

Úlohu sme testovali počas času cvičenia, mimo cvičení, ale pokúsili sme sa aj aktivovať už aktivovaný test. Následne bol používateľský príbeh testovaný na cvičení. Počas testovania nevznikli žiadne výhrady zo strany testujúcich.

6.1.2.3 Cvičiaci zobrazí anonymizované odpovede študentov

Analýza

Aby bolo možné podnietiť na cvičení diskusiu ohľadom preberanej problematiky, ideálnym prostriedkom na to sú práve vyplnené odpovede študentov na otázky týkajúce sa tejto témy. Po ukončení testu sa cvičiacemu zobrazia všetky otázky daného cvičenia so zvolenými, vyplnenými odpoveďami vo forme

grafu pre každú otázku. Študenti tak majú možnosť prekonzultovať svoju odpoveď a porovnať si ju so svojimi kolegami.

Návrh

Po vyplnení testu všetkými študentmi cvičiaci rozhodne, že ukončí test. To mu je umožnené stlačením tlačidla, ktoré sa nachádza hneď pod kódom cvičenia. Stlačením tlačidla sa načítajú z databázy jednotlivé odpovede študentov, pričom sú zoskupené podľa otázok. Odpovede na otvorené otázky sa zobrazia pod sebou ako zoznam odpovedí, pričom najvyššie sa bude vyskytovať odpoveď s najväčšou početnosťou. Odpovede na otázky s viacerými možnosťami budú zobrazované formou grafu zoradené podľa prirodzeného poradia tak, ako sú odpovede definované pri otázke.

Riešenie

Vytvorili sme časť systému, ktorá zobrazuje anonymizované odpovede. Stlačením tlačidla na ukončenie testu sa odošle požiadavka na server. To spôsobí, že sa práve aktívny test uzavrie. Následne sa získajú z databázy odpovede študentov z tohto cvičenia a ich odpovede sú zoskupené podľa otázok. Odpovede na otázky s viacerými možnosťami sú zobrazené na grafe Google chart, kde sa nachádza text a početnosť každej odpovede. Odpovede na otvorené otázky sú zoskupované podľa spoločných vzorov nájdených v odpovediach. Sú zobrazené vo forme zoznamu pod sebou, pričom najvyššie sa nachádza odpoveď s najväčšou početnosťou.

Testovanie

Počas testovania sme skúmali zobrazovanie a správne vykresľovanie odpovedí v grafe, ale aj v zozname odpovedí. Kontrolovali sme relevantnosť výsledkov, správnosť nájdených vzorov, porovnávali početnosti odoslaných odpovedí s početnosťami v aplikácii. Následne sme testovali na koncových používateľoch, ktorý prešli celým používateľským príbehom. Tí nemali k testovaniu žiadne pripomienky.

6.1.2.4 Cvičiaci po ukončení testu môže zobrazit' základnú štatistiku odpovedí

Analýza

Aby bolo možné podnietiť na cvičení diskusiu ohľadom preberanej problematiky, ideálnym prostriedkom na to sú práve vyplnené odpovede študentov na otázky týkajúce sa tejto témy. Po ukončení testu sa cvičiacemu zobrazia všetky otázky daného cvičenia so zvolenými, vyplnenými odpoveďami vo forme grafu pre každú otázku. Študenti tak majú možnosť prekonzultovať svoju odpoveď a porovnať si ju so svojimi kolegami. Po získaní skúseností a otestovaní konceptu na našom prototypu tak môžeme odstrániť zistené nedostatky a doplniť chýbajúcu funkcionálnu akou bolo napríklad zvýrazňovanie aktuálne prezeranej položky, rozdelenie osy nie na desatinné časti a iné.

Návrh

Po vyplnení testu všetkými študentmi cvičiaci rozhodne, že ukončí test. To mu je umožnené stlačením tlačidla, ktoré sa nachádza hneď pod kódom cvičenia. Stlačením tlačidla sa načítajú z databázy jednotlivé odpovede študentov, pričom sú zoskupené podľa otázok. Odpovede na otvorené otázky sa zobrazia pod

sebou ako zoznam odpovedí, pričom najvyššie sa bude vyskytovať odpoveď s najväčšou početnosťou. Odpovede na otázky s viacerými možnosťami budú zobrazované formou grafu zoradené podľa prirodzeného poradia tak, ako sú odpovede definované pri otázke. Grafy budú usporiadané do záložiek tak ako tomu bolo v prototypu s tým, že aktuálne vybraná otázka bude zvýraznená. Graf odpovedí na otvorené otázky bude zarovnaný na stred s vylepšeným dizajnom.

Riešenie

Vytvorili sme časť systému, ktorá vychádzala z konceptu aplikovaného v prototypu. Tá zobrazuje anonymizované odpovede usporiadané do grafu. Stlačením tlačidla prejsť na vyhodnotenie testu sa získajú z databázy odpovede študentov z aktuálneho cvičenia a ich odpovede sú zoskupené podľa otázok. Odpovede na otázky s viacerými možnosťami sú zobrazené na grafe Highchart, kde sa nachádza text, a početnosť každej odpovede. Odpovede na otvorené otázky sú zoskupované podľa spoločných vzorov nájdených v odpovediach. Sú zobrazené vo forme zoznamu pod sebou, pričom najvyššie sa nachádza odpoveď s najväčšou početnosťou.

Testovanie

Počas testovania sme skúmali zobrazovanie a správne vykresľovanie odpovedí v grafe, ale aj v zozname odpovedí. Kontrolovali sme relevantnosť výsledkov, správnosť nájdených vzorov, porovnávali početnosti odoslaných odpovedí s početnosťami v aplikácii. Následne sme testovali na koncových používateľoch, ktorý prešli celým používateľským príbehom. Tí nemali k testovaniu žiadne pripomienky.

6.1.2.5 Prihlasovanie cvičiacich cez konfiguráciu

Analýza

Cvičiaci, ktorí sú študentmi doktorandského štúdia sú vedení v AIS ako študenti. Preto je potrebné tento ich status po overení prihlasovacieho mena a hesla prepísať na učiteľa. Zároveň je potrebné umožniť im definovať vlastné heslá v prípade, že nesúhlasia so zadávaním svojho všeobecného hesla do aplikácie. Poslednou možnosťou je prihlasovanie iba cez konfiguráciu kombináciou mena a hesla.

Návrh

Vyplnením prihlasovacieho mena a hesla na úvodnej obrazovke sa údaje odošlú na overenie. Systém overí, či sa prihlasovacie meno nenachádza v konfiguračnom súbore a pokiaľ nastane zhoda, skontroluje sa či existuje k tomuto prihlasovaciemu menu aj konfiguračné heslo. Ak sa nenachádza, je táto kombinácia odoslaná na overenie cez LDAP službu. Po úspešnej autorizácii sú k dispozícii informácie o prihlásenej osobe, ktoré sa podľa potreby modifikujú podľa konfigurácie pre zadané prihlasovacie meno. Konfiguračné nastavenia informácií sú nadradené nad štandardnými nastaveniami.

Riešenie

Vytvorili sme časť systému, ktorá slúži na čiastočné alebo úplne obídanie autorizácie cez LDAP. To sme zabezpečili vytvorením konfiguračného súboru vo formáte JSON. Jeho obsahom sú spomedzi všetkých dát

aj konfiguračné nastavenia pre cvičiacich a osoby zahrnuté v procese výučby. Po odoslaní prihlasovacích údajov nastane overenie prihlasovacích údajov a získanie dodatočných informácií o používateľovi. Ak sa v konfigurácii nachádzajú dodatočné informácie pre toto prihlasovacie meno, pôvodne získané informácie sa nahradia konfiguračnými. V prípade, že sa v konfigurácii nachádza kombinácia prihlasovacieho mena a hesla, vynechá sa autorizácia cez LDAP protokol a overenie sa vykoná porovnaním s konfiguračnými nastaveniami.

Testovanie

Testovanie prebiehalo použitím prihlasovacích údajov členov tímu, testovacích používateľov, cvičiacich a cvičiacich študujúcich doktorandské štúdium.

6.1.3 Modul študent

6.1.3.1 Študent sa prihlási cez AIS LDAP

Analýza

Aby mohol cvičiaci sprístupniť a študent vyplniť test je potrebné aby sa prihlásili do nášho informačného systému. Na úvodnej stránke vyplnia a odošlú svoje prihlasovacie meno a heslo. Prihlasovacie údaje sú po odoslaní overené a používateľ môže využívať náš informačný systém. V prípade nesprávnej kombinácie mena a hesla je vyzvaný na opakované zadanie prihlasovacích údajov.

Návrh

Používateľ zadá na úvodnej stránke do formuláru svoje prihlasovacie meno a heslo. Je potrebné použiť rovnaké prihlasovacie údaje ako tie, ktoré zadáva do akademického informačného systému. Po odoslaní sú prihlasovacie údaje odoslané na server, kde sa spracujú a overia. Po úspešnom overení prihlasovacieho mena a hesla je používateľ oprávnený vstúpiť do informačného systému. V prípade nesprávnej kombinácie mena a hesla je mu zobrazená chybová hláška a je vyzvaný na opakované zadanie prihlasovacích údajov.

Riešenie

Vytvorili sme časť systému, ktorá slúži na overenie prihlasovacích údajov a získanie dodatočných informácií o používateľovi. Pozostáva z prihlasovacieho formuláru, ktorý sa nachádza na úvodnej stránke a zaznamená prihlasovacie údaje používateľa. Tie sú po odoslaní požiadavky spracované serverom. Druhá časť systému, na strane servera, slúži na spracovanie informácií a ich overenie prostredníctvom služby LDAP protokolu akademického informačného systému. Služba je dostupná na dvoch serveroch. Ak primárny server zlyhá, požiadavka je automaticky odoslaná na záložné servery. Po úspešnom overení kombinácie prihlasovacieho mena a hesla sa získajú o používateľovi dodatočné informácie. Tie sú uložené spolu s vygenerovaným jednoznačným identifikátorom prihlásenia. Po úspešnom spracovaní sa klientskej

aplikácii odošle jednoznačný identifikátor daného prihlásenia. V prípade zlyhania mu je poslaná informácia o zlyhaní vo forme chybovej hlášky a dôvod zlyhania je zaznamenaný do logu.

Testovanie

Prvotné testovanie prebiehalo použitím prihlasovacích údajov členov tímu. Použili sme správne, ale aj rôzne chybné kombinácie mena a hesla. Funkcionalita overovania prihlasovacích údajov v prípade výpadku primárneho servera bola nasimulovaná zadaním chybného adresy primárneho servera.

6.1.3.2 Študent odpovedá na otázky s otvorenou odpoveďou

Analýza

Aby mohol študent absolvovať test, je potrebné aby mu bola vybraná množina otázok na ktoré odpovie a test odošle. Množina otázok pozostáva z dvoch typov otázok. Jednými z nich sú otázky s otvorenou odpoveďou. Takáto otázka pozostáva iba zo zadania otázky, na ktorú študent napíše svoju odpoveď.

Návrh

Zadaním kódu na cvičení je študentovi náhodne vybraná množina otázok, z ktorých je zložený test. Otvorené otázky poskytnú študentovi priestor na vloženie odpovede, ktorá môže pozostávať z pár slov alebo celej vety. Je potrebné aby boli otázky rozdeľované podľa typu už pri ich importe. Treba zvoliť odlišný spôsob spracovania pri príprave na zobrazenie klientskej aplikácii, ale aj pri spracovávaní odpovedí na tento typ otázok. Odpovede sa budú ukladať vo forme textu a bude potrebné ich manuálne ohodnotiť.

Riešenie

Otázky sú pri importe z CSV súboru rozdeľované podľa možností a typu očakávanej odpovede. Pri výbere a následnej príprave na zobrazenie je posielaný klientskej aplikácii typ otázky, na základe ktorého určí akou formou bude zobrazená. Používateľovi sa zobrazí zadanie otázky a textové pole, kam môže vložiť svoju odpoveď. Po odoslaní testu je odpoveď spracovaná a uložená do databázy ako text.

Testovanie

Zobrazenie otázok bolo testované s variabilnou dĺžkou otázky a špecifickým formátovaním otázok, kvôli správne mu zobrazeniu v aplikácii. Odpovede na otázky boli vkladané taktiež s rôznou dĺžkou, špeciálnymi znakmi, prípadne prázdne.

6.1.3.3 Študent odpovedá na otázky s viacerými možnosťami

Analýza

Aby mohol študent absolvovať test, je potrebné aby mu bola vybraná množina otázok, na ktoré odpovie a test odošle. Množina otázok pozostáva z dvoch typov otázok. Jednými z nich sú otázky s viacerými

možnosťami. Tento typ otázok pozostáva zo zadania otázky a možnosťami, z ktorých študent vyberie tie, o ktorých si myslí, že sú správne.

Návrh

Zadaním kódu na cvičení je študentovi náhodne vybraná množina otázok, z ktorých je zložený test. Otázky s viacerými možnosťami poskytnú študentovi množinu vopred pripravených odpovedí, z ktorých si následne vyberie tie, o ktorých si myslí, že sú správne. Nemusí však vybrať žiadnu, ale môže označiť aj všetky odpovede za správne. Je potrebné aby boli otázky rozdeľované podľa typu už pri ich importe. Treba zvoliť odlišný spôsob spracovania pri príprave na zobrazenie klientskej aplikácii, ale aj pri spracovávaní odpovedí na tento typ otázok. Odpovede sa budú ukladať vo forme textu, pričom sa budú ukladať čísla možností, ktoré študent zvolil. Bude ich tak možné vyhodnotiť nielen automaticky, ale v prípade nutnej kontroly aj manuálne.

Riešenie

Otázky sú pri importe z CSV súboru rozdeľované podľa možností a typu očakávanej odpovede. Pri výbere a následnej príprave na zobrazenie je posielaný klientskej aplikácii typ otázky, na základe ktorého určí akou formou bude zobrazená. Používateľovi sa zobrazí zadanie otázky a množina maximálne piatich náhodne rozdelených, vopred definovaných odpovedí, z ktorých si študent zvolí správne. Po odoslaní testu je odpoveď spracovaná a uložená do databázy vo forme textu, pričom sa neukladajú slovné odpovede, ale čísla odpovedí usporiadané vzostupne oddelené čiarkami.

Testovanie

Zobrazenie otázok bolo testované s rôznym počtom vopred definovaných odpovedí, variabilnou dĺžkou otázky, špecifickým formátovaním otázok, kvôli správne mu zobrazeniu v aplikácii a rôznym počtom vybraných možností. Odpovede na otázky boli vyberané v rôznom poradí a následne porovnávané s dátami uloženými do databázy.

6.1.3.4 Študent si zobrazí test s otázkami

Analýza

Po úspešnom prihlásení do aplikácie v roli študenta je používateľovi zobrazený formulár na zadanie kódu cvičenia. Tento kód sa zobrazí cvičiacemu po aktivovaní testu, ktorý ho dá študentom k dispozícii formou projekcie, alebo ho napíše na tabuľu. Študent po zadaní správneho, aktívneho kódu cvičenia je presmerovaný na test pozostávajúci z náhodne vygenerovanej množiny otázok určených pre toto cvičenie.

Návrh

Študent vyplní a odošle formulár s kódom cvičenia. Platnosť zadaného kódu sa overí voči databáze a ak je zadaný kód v poriadku, teda študent má právo písať test, je mu zobrazená jeho náhodne vygenerovaná množina otázok, prípadne vopred definovaných odpovedí. Po vyplnení testu študent kliknutím na tlačidlo

odovzdá test. V prípade chyby pri overovaní platnosti zadaného kódu sa študentovi zobrazí chybová hláška.

Riešenie

Vytvorili sme časť systému zabezpečujúcu zadanie kódu, overenie jeho platnosti a následné presmerovanie na vyplnenie testu. V klientskej aplikácii bolo potrebné vytvoriť obrazovku, ktorá vyzve študenta na zadanie kódu cvičenia. Po odoslaní kódu cvičenia spolu s identifikátorom aktuálneho prihlásenia sa na serveri overí, či takýto kód cvičenia existuje, následne sa overí, či je test práve aktívny a či už nebol práve prihláseným študentom niekedy vyplnený. V prípade zistenia nedostupnosti testu je študentovi zobrazená chybová hláška a vykoná sa záznam o tejto aktivite. Ak však táto kontrola prejde, náhodne sa vyberie množina otázok, ktoré sa pošlú klientskej aplikácii. Celá množina otázok je zobrazená na jednej obrazovke, každá samostatne s miestom na odpoveď. Odovzdanie testu je vykonané naraz pre celú množinu otázok.

Testovanie

Zobrazenie testu sme testovali zadávaním rôznych platných, neplatných, aktívnych a neaktívnych kódov cvičení. Tiež sme vykonali test na duplicitu vykonania testu. Na zvyšnej časti testovania s podieľali študenti, ktorý prešli celým používateľským príbehom, tak ako bolo očakávané. Tí počas testovania nemali žiadne pripomienky.

6.1.3.5 Systém neumožní nič okrem povolených akcií

Analýza

Používatelia vedia byť niekedy vynaliezaví a snažia sa obísť, prípadne nájsť slabé miesta informačného systému. Preto je potrebné predísť úniku informácií, neoprávnenému manipulovaniu s testami alebo neoprávnenému zisku bodov tak, že zabezpečíme aplikáciu pred nekorektnými vstupmi. Aby používatelia neboli zmätený pri používaní našej aplikácie, je potrebné ošetriť chybové stavy a vhodným spôsobom ich informovať o dôvode zlyhania aplikácie

Návrh

Cieľom tejto úlohy je poskytnúť používateľom spätnú väzbu o ich akciách a zobraziť ju vo forme chybovej hlášky. Je potrebné zaviesť jednoznačný identifikátor prihlásenia, ktorý je kľúčovým prvkom zabezpečenia. Následne identifikovať slabé miesta a dostatočne ich zabezpečiť pred nekorektnými vstupmi.

Riešenie

Kľúčovým prvkom zabezpečenia bolo zavedenie jednoznačného identifikátora prihlásenia, ktorého platnosť sa kontroluje pri každej akcii používateľa odoslaním požiadavky na server. Ten skontroluje časovú platnosť identifikátora a pri každej požiadavke obnoví jeho platnosť. Tým sme obmedzili prístup na všetky stránky, pokiaľ používateľ neprejde autentifikáciou. Ďalším riešením zabezpečenia bolo ošetrenie

chybových vstupov aby neprišlo k zlyhaniu databázy z dôvodu neexistujúcej referencie. Taktiež bolo potrebné pri každej nepovolenej akcii informovať používateľa o chybe a takúto aktivitu zaznamenať.

Testovanie

Aplikáciu globálne testovali všetci členovia tímu. Nemalo zmysel nechať túto úlohu otestovať používateľmi, nakoľko by sa žiaden z nich k nepovolenej aktivite nepriznal.

6.1.3.6 Vynechanie prázdnych odpovedí multichoice otázok pri generovaní testu a zobrazení štatistiky

Analýza

Pri vypĺňaní testu môže nastať situácia, že sa nevyužijú všetky preddefinované odpovede pri otázkach s viacerými možnosťami. Nie je vhodné aby sa zobrazovali prázdne miesta pri možnosti výberu správnej odpovede, ale ani pri zobrazení štatistiky. Je potrebné eliminovať výskyt takýchto možností v oboch prípadoch.

Návrh

Prázdne možnosti sa pri generovaní testu nezoberú ako relevantné odpovede a jednoducho sa preskočia. Toto treba zabezpečiť na strane servera, pri transformovaní dát a generovaní testu. Za preddefinovanú odpoveď sa bude považovať iba neprázdny reťazec znakov. Rovnako tomu bude aj pri generovaní štatistiky odpovedí, prázdne možnosti sa preskočia, čím sa aj zmenší veľkosť grafu a zvýši sa jeho čitateľnosť.

Riešenie

Túto funkcionality je potrebné implementovať na strane servera. Počas procesu generovania testu, konkrétne pri transformácii dát. Preddefinované odpovede sa kontrolujú a do zobrazenia sa dostanú iba neprázdne reťazce znakov dĺžky aspoň jedného znaku. Študent tak nemá možnosť ani omylom vybrať takúto možnosť. Rovnako pri generovaní štatistiky sa neberú do úvahy prázdne preddefinované odpovede, ani prázdne (nevyplnené) odpovede.

Testovanie

Testovanie prebiehalo použitím rôznych kombinácií preddefinovaných odpovedí. Počas testovania sa nevyskytli nijaké nežiaduce situácie.

6.2 AlefTNG

6.2.1 Modul Administrátor

6.2.1.1 Administrátor si môže zobrazit' štatistiku pre konkrétnu otázku

Analýza

Aby mal učiteľ globálny prehľad o vedomostiach študentov, potrebuje si zobrazit' ich odpovede na otázku. Zistí tak, či porozumeli otázke a či nadobudli potrebné vedomosti na jej zodpovedanie a získanie bonusových bodov. Táto funkcionality mu tiež umožní vyhodnotit' celkovú úspešnosť odpovedí študentov na otázku, identifikovať problematické oblasti učiva, prípadne prehodnotit' náročnosť otázky.

Návrh

Administrátor systému, čiže väčšinou vedúci predmetu, bude mať v prehľade otázok možnosť zobrazenia štatistiky pre konkrétnu otázku. Po kliknutí na ikonku štatistiky bude presmerovaný na zobrazenie grafu odpovedí. Princíp fungovania a zobrazovania bude rovnaký ako pri zobrazovaní štatistiky po ukončení testu s tým rozdielom, že sa zobrazí iba jedna otázka a všetky odpovede študentov na ňu.

Riešenie

Vytvorili sme komponent systému pre administrátora v sekcii otázok, ktorý mu umožňuje zobrazenie štatistiky všetkých odpovedí pre konkrétnu otázku. Princíp fungovania bol prepoužitý zo zobrazenia štatistik cvičenia, testu. Jediným rozdielom je, že nám nezáleží na identifikátore cvičenie, na ktorej bola odpoveď zadaná, ale berieme do úvahy všetky odpovede.

Testovanie

Počas testovania sme sa zamerali na správne generovanie štatistík, či sa vyberajú správne odpovede pre otázku a ako sa zobrazí veľké množstvo odpovedí v grafe. Nenašli sme na žiadne problémy ani anomálie pri generovaní a zobrazovaní štatistík.

6.2.1.2 Administrátor môže importovať otázky (vrátane ošetrovania chybových stavov)

Analýza

Musí byť možné importovať nové otázky aj skrz cez internetový prehliadač.

Importovanie otázok funguje skrz csv súbory a obrázky, ktoré musia byť nahrané na server.

Návrh

V administratívnej časti aplikácie kde sú vypísané všetky otázky v aplikácii musí byť pridaná možnosť nahrat' otázky a obrázky k nim patriace.

Riešenie

Bol pridaný modálny html element, ktorý sa rozbalí po kliknutí na tlačidlo Importovať otázky, tento element obsahuje tlačidlo a element na výber súboru.

Je očakávaný Zip súbor, ktorý obsahuje csv otázok a obrázky. Po spracovaní je výstup zaslaný naspäť administrátorovi ako textový súbor.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.1.3 Administrátor predmetu pridáva a mení termíny

Analýza

Administrátor musí mať možnosť pridávať nové termíny a editovať tie staré. Pri vytváraní, rovnako ako aj pri menení existujúcich je potrebné nastavovať všetky potrebné súvisiace parametre pre termín.

Návrh

Do administrátorskej časti sa pridá nová časť na správu termínov. Pri vytváraní nového alebo editácii starého sa zobrazí formulár obsahujúci relevantné informácie - vedúceho termínu (cvičiaceho), týždeň termínu, plánovaný čas a dátum termínu, čas trvania testu a čas do zverejnenia výsledkov od ukončenia testu a rovnako tak aj termínové koncepty. Okrem toho aj informácie pre recommendačný systém definujúce použitie termínových a týždňových konceptov.

Riešenie

Do administrátorského modulu bola pridaná časť pre správu termínov. V nej je možné prechádzať zoznam termínov. Je rovnako možné pridávať nové termíny alebo editovať existujúce. Pre vytváranie/editáciu sa zobrazuje formulár so všetkými údajmi uvedenými hore. V závislosti na type parametrov sa údaje zobrazujú v podobe selectov – dostupní učitelia, dostupné termínové koncepty (multiselect), v podobe checkboxov – použitie termínových a použitie týždňových konceptov alebo v podobe časových selectov a textových polí akceptujúcich numerické hodnoty.

Testovanie

Pri testovaní boli vytvárané nové a editované mnohé existujúce termíny. Pre ne boli menené hodnoty jednotlivých parametrov a súvisiacich dát. Počas testovania sa neodhalili výrazné nedostatky implementovaného riešenia.

6.2.1.4 Cvičiaci/Administrátor vie definovať čas, po ktorom bude sprístupnené hodnotenie

Analýza

Výsledky testov nesmú byť často k dispozícii hneď po odovzdaní testu. A to aj napriek tomu, že platí, že pre viaceré typy otázok je možné validovať ich správnosť automaticky. Dôvody môžu byť v podobe manuálne opravovaných otázok alebo aj v snahe zaujať študentov pri potestovej konzultácii a prechádzaní štatistik testu, počas ktorých sa vysvetľujú správne odpovede. Cvičiaci musí vedieť nastaviť takýto čas.

Návrh

Do databázy je potrebné doplniť nový stĺpec zaznamenávajúci čas, po ktorom bude sprístupnené hodnotenie. Pri snahe študenta prístup k hodnoteniu sa porovnáva čas pokusu o prístup s časom ukončenia testu doplneným o čas sprístupnenia. Tento čas je potrebné nastaviť na stránke nastavení termínu, ku ktorému má prístup učiteľ aj administrátor.

Riešenie

K existujúcej funkcionalite overovania dostupnosti testu bolo doplnené uchovávanie času sprístupnenia testu vo forme čísla vyjadrujúceho počet minút do sprístupnenia hodnotenia. V zozname termínov je možné prejsť na stránku nastavení, kde je možné do textového pola zadať akúkoľvek celočíselnú hodnotu vyjadrujúcu počet minút do zverejnenia. S časom sa následne pracuje podľa pravidiel implementovaných pri prvotnom zavedení času sprístupnenia testu `cooldown_time`.

Testovanie

V nastaveniach bola overovaná možnosť zadávania rôznych časov. Formát je povolený len numerický. Pokiaľ sa čas ukončenia testu rozšírený o počet minút v `cooldown_time` nedá považovať za menší alebo rovný s aktuálnym časom, študent k výsledkom prístup v žiadnom z testovaných prípadov nemá.

6.2.1.5 Administrátor môže pridávať, meniť a upravovať otázky a používateľov a ich roly

Analýza

Pridávanie otázok je možné len prostredníctvom importov v podobe rake taskov spúšťaných cez terminál. Je potrebné, aby pridávanie bolo možné aj cez používateľské rozhranie. Rovnako tak je potrebné umožniť meniť odpovede na dané otázky, ich správnosť a počet. Tieto funkcie musia byť vytvorené ako súčasť administrátorskej časti aplikácie. V systéme existujú aj lokálni používatelia, ktorých tvorba je založená rovnako len na importe dát. Lokálnych používateľov by preto bolo vhodné rovnako vytvárať v rozhraní aplikácie. Pre existujúcich používateľov je následne možné meniť ich atribúty. V prípade používateľov, ktorý sa autentifikujú cez LDAP, je možné meniť len rolu. Tá môže byť zmenená na vyššie oprávnenie len cez zmenu stĺpca v databáze. Je preto rovnako potrebné doplniť aj túto časť do administrátorského modulu aplikácie.

Návrh

Do administrátorskej časti je potrebné pridať samostatné časti na manažment analyzovaných súčastí. Pre používateľov ako aj pre otázky je vytváraný prehľadný zoznam a stránky s formulármi určenými na editáciu a vytvorenie nového záznamu. V zozname je potrebné zobrazovať údaje potrebné pre bezproblémové rozlišovanie medzi záznamami. Vo formulároch na pridanie a editáciu je potrebné zaznamenať všetky potrebné parametre.

Riešenie

Opis riešenia napr. Vytvorili sme časť systému, ktorá prijíma súbory vo formáte CSV. Obrázky sa nahrávajú cez interface na hlavnej stránke administrátora. Pridali sme zobrazenie pridaných otázok v prehľadnej tabuľke. Pre editáciu a pridávanie otázok boli pridané formuláre obsahujúce všetky potrebné parametre súvisiace s pomenovaním a typom otázky ako aj s ich náročnosťou. V samostatnej karte v navrhnutom používateľskom rozhraní sa pridávajú odpovede a zobrazuje náhľad otázky. Pri pridávaní odpovedí je potrebné riadiť sa pravidlami správneho počtu správnych odpovedí. V prípade porušenia týchto pravidiel zasahuje do vytvárania odpovedí systém, ktorý zabráni nepovoleným akciám. Pre používateľov platia pre zoznam rovnaké pravidlá ako pre otázky. Pridávať je však možné len lokálnych používateľov. Im je možné následne aj editovať väčšinu atribútov vrátane hesla. Pre LDAP používateľov sa kontrola správnosti hesla vykonáva cez LDAP a editovať je možné len ich rolu.

Testovanie

Pri testovaní boli vytvárané nové a editované mnohé existujúce otázky a používatelia. Pre ne boli menené hodnoty jednotlivých parametrov a súvisiacich dát. Pre otázky to boli aj ich odpovede. Pre používateľov sa testovali zmeny nad lokálnymi používateľmi, rovnako ako aj nad používateľmi využívajúcimi na prihlásenie LDAP. Počas testovania sa neodhalili výrazné nedostatky implementovaného riešenia.

6.2.1.6 Systém loguje autorizačné a autentifikačné aktivity používateľov

Analýza

Administrátor celkového systému potrebuje mať prehľad o prihlasovacích aktivitách používateľov. K týmto aktivitám patrí okrem samotného prihlásenia aj prístup k testu. Správy o stave a prípadnom úspechu či neúspechu je potrebné prenášať do logov.

Návrh

Pre riešenie bol zvolený vlastný logovací prístup, odvodený od logovania, ktoré je súčasťou frameworku Ruby on Rails. Logovanie zachytáva logy na rôznych leveloch. K nim patrí warn, debug, error, info, unknown. Takéto logovanie sa následne použije v závislosti na type potrebnej zachytenej informácie. Rozdiel je v úspechu a neúspechu.

Riešenie

Do aplikácie bol pridaný nový modul v dostupných knižniciach. V nich bol vytvorený vlastný logger, ktorý na samotný výstup využíva pôvodný vstavaný. Zjednodušuje však prácu s formátovaním do správneho formátu. Akákoľvek logovaná správa obsahuje informácie v podobe správy a dodatočných parametrov. Pre jednotlivé logovacie úrovne boli zvolené iné farby výstupu na konzole. Súčasťou každého logu je časový údaj. Parametre dopĺňajú hlavičku správy. Pre prihlasovacie akcie vždy zaznamenávajú aj IP adresu používateľovho zariadenia. Pre prístup k testu sú zaznamenávané aj údaje o prihlásenom používateľovi – prihlasovacie meno.

Testovanie

Testovanie prebiehalo na opakovanej simulácii prihlasovacích aktivít s rozdielnym očakávaným výsledkom – úspechom a neúspechom. Pre každý prípad bola zaznamenaná v logoch požadovaná správa.

6.2.1.7 Administrátor pri štarte systému nahrá súbor s cvičeniami

Analýza

Administrátor pri štarte systému nahrá súbor s cvičeniami. Súbor je koncipovaný ako csv, v ktorom sa nachádzajú údaje o začiatku a konci cvičenia a pridelení cvičiaci. Pre každé cvičenie sa vytvára v súbore len jeden záznam, ktorý obsahuje termín prvého cvičenia v semestri. Následne sa za použitia generátora vygenerujú záznamy na všetky týždne, podľa preddefinovaného počtu týždňov. Všetky vytvorené záznamy sa mapujú na konkrétne týždne a generátor im doplní aj ich prístupový kód. Všetky takto vytvorené záznamy si cvičiaci môže následne prezerať.

Nahrávanie sa deje za pomoci ruby skriptu na to vytvoreného, ktorý je dodávaný ako súčasť nášho systému. Ten je možné jednoducho spustiť štandardným spustením ruby skriptu priamo z príkazového riadku. Takto vygenerované súbory sa následne pomocou vytvoreného rake tasku dajú vložiť priamo do databázy aplikácie. Všetky takto vytvorené cvičenia je následne možné prehliadať na stránkach učenia pri rolách administrátor a učiteľ.

Návrh

Administrátor dostane pri vytváraní systému k dispozícii od osoby zodpovednej za predmet súbor csv obsahujúci harmonogram cvičení v týždni. Nad týmto súborom spustí vytvorený skript, ktorý vygeneruje automaticky ďalšie údaje, rovnako ako aj záznamy pre všetky týždne semestra podľa zvolenej dĺžky semestra. Výsledkom tohto skriptu je vytvorenie súboru csv, ktorý následne pomocou rake tasku vloží do databázy.

Riešenie

Na zabezpečenie riešenia bol vytvorený skript, ktorý generuje harmonogram cvičení pre celý semester. Skript sa spúšťa štandardne z príkazového riadka pomocou run ruby. Súčasťou vstupných parametrov skriptu je súbor obsahujúci harmonogram cvičení a číslo určujúce dĺžku semestra v týždňoch. Skript pre

každý záznam vo vstupnom súbore vytvára niekoľko nových záznamov vo výstupnom súbore (podľa počtu týždňov), pričom k údajom o začiatku a konci cvičenia a identifikátoru cvičiaceho pridáva identifikátor týždňa a prístupový kód testu. Získaný súbor je možné vložiť do databázy pomocou rake tasku vytvoreného predtým.

Návod na nahrávanie cvičení

1. Na spustenie skriptu potrebujeme csv súbor obsahujúci harmonogram cvičení v týždni. Súbor obsahuje stĺpce Exercise start, Exercise end a Lecturer, záznamy sú pre každé cvičenie v prvom týždni semestra.
2. Skript spustíme príkazom:

```
run ruby exercisescsv2csv.rb [CESTA K VSTUPNÉMU SÚBORU] [DĹŽKA SEMESTRA]
```

3. Vo vygenerovanom súbore exercises.csv je možné následne manuálne upravovať hodnoty.
4. Súbor sa vloží do databázy spustením rake tasku v príkazovom riadku nasledovne:

```
rake tesa:data:importexercises[exercises.csv]
```

Testovanie

Testovanie prebiehalo vytvorením viacerých verzií súborov csv. V úlohe sa očakáva, že údaje sa do csv vkladajú v správnom formáte. Na chyby spôsobu spúšťania skriptu je administrátor upozornený.

6.2.1.8 Administrátor vytvorí termíny cvičení počas semestra

Analýza

Administrátor pri štarte predmetu vloží súbor (csv) s potrebnými údajmi na rozvrhnutie cvičení a cvičiacich. Je potrebné, aby administrátor vložil tieto údaje z dôvodu základnej funkcionality pridávanie do Alef-TNG vo verzii Tesa. Úloha sa čiastočne v priebehu vývoja nahradila úlohou: Administrátor pri štarte systému nahrá súbor s cvičeniami, ktorá obsahuje pokročilejšie riešenie problému.

Návrh

Administrátor pri štarte systému nahrá rozpis cvičení na semester pomocou jemu dodaného súboru csv, ktorý obsahuje údaje potrebné pre identifikáciu cvičenia (termín, kód cvičenia, cvičiaci, týždeň semestra). Všetky tieto údaje administrátor vloží do databázy spustením rake tasku.

Riešenie

Administrátor pre nahratie cvičení do databázy spúšťa rake task v tvare:

```
rake tesa:data:importexercises[filename.csv].
```

Skript obsahuje štandardné prechádzanie záznamov súboru, pre ktoré vytvára záznam v databáze, pričom mapuje stĺpce csv súboru na stĺpce databázy.

Testovanie

Testovanie prebiehalo využitím viacerých verzií csv súborov. Počas testovania sa zistili nedostatky riešenia, v ktorom je potrebné vytváranie csv súboru pre celý semester. Z tohto dôvodu bola vytvorená nová user story, ktorej úlohou je predgenerovať csv pre celý semester len z údajov pre jeden týždeň. Tá vytvára súbory csv presne podľa stanoveného formátu, a preto sa pri importe neobjavujú žiadne problémy.

6.2.1.9 Administrátor vloží termíny cvičení a otázky do systému

Analýza

Administrátor vkladá termíny cvičení pomocou už predtým vytváraných csv súborov, ktorých popis je v časti Administrátor pri štarte systému nahrá súbor s cvičeniami. Otázky do systému sa vkladajú využitím csv súboru. Tento súbor obsahuje konkrétne otázky, pre ktoré sú definované dôležité údaje charakterizujúce cvičenie. Ide o tému, zadanie otázky, nadpis otázky, odpovede, typ otázky a správne odpovede. Všetky takéto otázky sa pridelujú týždňom podľa pravidla z pôvodného prototypu – jedna téma na konkrétny týždeň. Téma teda v riešení určuje koncept aj týždeň otázky zároveň. Otázky sú vkladané pomocou rake tasku do databázy aplikácie. Z otázok v databáze sa následne vyberajú otázky pre testy v týždňoch.

Návrh

Vytvorené otázky vloží vedúci predmetu do súboru csv podľa danej štruktúry stĺpcov. Tie obsahujú:

```
tema;id;otazka;odpoved_a;odpoved_b;odpoved_c;odpoved_d;odpoved_e;typ;spravne
```

Tieto údaje sa vkladajú do databázy pomocou rake tasku, ktorý musí spracovať jednotlivé stĺpce každého záznamu a vytvoriť odpovedajúce záznamy v databáze. Téma určuje týždeň, rovnako ako aj koncept (tému okruhu otázok). Otázka obsahuje samotné zadanie, id obsahuje nadpis otázky, odpovede obsahujú jednotlivé odpovede. Maximálny počet je 5, ale môže byť aj nižší. Typ otázky obsahuje informáciu o spôsobe odpovedania na otázku. To môže byť výberom z viacerých možností (multichoice question) alebo otvoreným textom (opentext question). Stĺpec spravne obsahuje zoznam správnych odpovedí (pri otázkach s výberom viacerých možností).

Riešenie

Administrátor na vloženie otázok spúšťa rake task:

```
rake tesa:data:import_tests[filename.csv].
```

Tento task prechádza postupne jednotlivými záznamami otázok a pre každú z nich vytvára položky v databáze charakterizujúce otázku. Vkladá otázky, odpovede, koncepty a prípadné týždne, ak v sa databáze v dobe behu tasku nenachádzajú. Správnosť odpovedí zistí z posledného stĺpca csv súboru.

Testovanie

Testovanie prebiehalo vytvorením viacerých verzií súborov csv. V úlohe sa očakáva, že údaje sa do csv vkladajú v správnom formáte. Na chyby spôsobu spúšťania skriptu je administrátor upozornený.

6.2.1.10 Administrátor môže nastaviť globálny čas, po ktorom sa zavrú testy/termíny

Analýza

Administrátor môže nastaviť globálny čas, po ktorom sa zavrú testy. Čas určuje dĺžku trvania testu. Po skončení odpočítavania tohto času učiteľ uzavrie termín testu a študenti tak už nemôžu pristupovať k testu, rovnako tak nemôžu ani odovzdať začatý test. Časomiera testu je zobrazovaná na stránke termínu spolu s prihlasovacími údajmi a počtom odovzdaných testov. Čas sa aktualizuje každú sekundu a môže byť zmenený aj v priebehu testu. Zmeny sa prejavia v časomiere okamžite.

Návrh

Učiteľ má v zozname testov možnosť prejsť na stránku nastavení termínu, v ktorom môže nastaviť čas trvania testu. Po spustení testu sa na jeho stránke zobrazuje okrem potrebných prístupových údajov k testu (kód, linka, QR kód) a počtu odovzdaných testov aj pole s časomierou. Tá je pravidelne aktualizovaná. Blížiaci sa koniec testu je signalizovaný farebným zvýraznením a blikaním zostávajúceho času.

Riešenie

Učiteľ má v zozname položku nastavenia termínu. Po prejdení na ňu sa mu zobrazí textové pole na zadanie čísla určujúceho počet minút trvania testu. Nastavenie potvrdzuje tlačidlom. Po spustení testu sa vo vrchnej časti zobrazuje digitálny zápis času, ktorý sa pravidelne aktualizuje. Keďže v riešení bolo potrebné prihliadať na možnosť zmeny času, je tento čas získavaný pravidelnými Ajax volaniami na server, ktoré už boli implementované v súvislosti so zobrazovaním počtu odovzdaných testov. Doplnené tak bolo len odosielanie časovej informácie. V posledných sekundách sa farba zobrazovaného textu mení na červenú a text pravidelne bliká, čo je zabezpečené pridaním ďalších CSS tried.

Testovanie

Testovanie riešenie spočívalo v zadávaní rozdielných textových záznamov do textového pola pre čas, pričom akceptované boli len numerické zápisy. V zobrazovaní testu sa následne testoval správny spôsob odpočítavania časomiere, pri nastavenom aj nenastavenom čase v databáze.

6.2.1.11 Systém umožní zadať matematické vzorce v zadaní

Analýza

Aby bolo možné zobraziť systémom netriviálne matematické výrazy, prípadne oddeliť, zvýrazniť časť otázky, musí rozumieť Tex notácii. Tá poskytuje široké možnosti zápisu matematických výrazov, vďaka čomu je veľmi obľúbená v akademickom prostredí. Vhodným prostriedkom na to je knižnica KaTeX

poskytujúca práve túto funkcionality. Podľa porovnaní s ostatnými alternatívami, tejto knižnici trvá vykreslenie výrazu najkratšie.

Návrh

Pri vkladaní bude matematický výraz ohraničený oddeľovacími znakmi „[“ z ľava a „]“ z prava. Po vykreslení stránky sa spracujú všetky textové polia nachádzajúce sa na stránke. Ak knižnica zistí prítomnosť matematického zápisu ohraničeného špeciálnymi znakmi, pokúsi sa o jeho porozumenie a spracuje ho do formy matematického výrazu.

Riešenie

Vytvorili sme časť systému, ktorá automaticky spracováva špeciálne naformátované zápisy matematických výrazov v jazyku Tex. Mechanizmus automatickej detekcie výrazov po vykreslení stránky prehľadá všetky textové polia a zistí v nich prítomnosť oddeľovacích znakov. Pokiaľ nájde časť textu oddelenú takými značkami, jej obsah pošle na spracovanie knižnici KaTeX. Tá vráti výraz spracovaný v prirodzene čitateľnej forme, v akej by bol zapísaný napríklad v knihe.

Testovanie

Počas testovania sme skúmali zobrazovanie a správne vykresľovanie matematických výrazov v zadaní otázky a preddefinovaných odpovediach. Sústredili sme sa na použitie rôznych formátovacích značiek a správne zobrazovanie v rôznych prehliadačoch. Počas testovania sme zistili absenciu podpory niektorých formátovacích značiek nachádzajúcich sa v jazyku Tex.

6.2.1.12 Systém umožňuje formátovať vzorec

Analýza

Počas testovania úlohy „System umožní zadať matematické vzorce v zadaní“ sme narazili na nedostatok knižnice KaTeX, ktorým je absencia formátovacích značiek nového riadku. Tie sú pre náš systém dôležité, aby sa dal viac riadkový matematický výraz napísať do jedných formátovacích značiek, akým je napríklad časť kódu RAM.

Návrh

Výskum funkcionality knižnice ukázal, že je možné použiť formátovacie značky nového riadku „//“ po obalení rovnice do formátovacích značiek „\begin{matrix}“ na začiatku a „\end{matrix}“ na konci. V takomto prípade systém formátuje matematický výraz ako maticu, kde je možné jednotlivé riadky oddeliť špeciálnym znakom nového riadku.

Riešenie

Vytvorili sme funkciu v jazyku JavaScript, ktorá pri spracovaní matematického výrazu prezrie, či sa v ňom nenachádza formátovacia značka nového riadku. Ak výraz značku obsahuje, funkcia pridá na jeho začiatok

reťazec „\begin{matrix}” a na koniec reťazec „\end{matrix}“. Výsledný výraz bude rozdelený na riadky podľa vzorca v texte.

Testovanie

Počas testovania sme sa zamerali na použitie formátovacích značiek v texte matematického zápisu. Otestovali sme, či funguje správne ak sa značka nového riadku v texte nenachádza. Formátovacia funkcia fungovala podľa požiadaviek a nepozorovali sme žiadne nežiaduce správanie

6.2.2 Modul vyučujúci

6.2.2.1 Učiteľ ohodnotí otvorené odpovede študentov aj pomocou regulérnych výrazov

Analýza

Automatické hodnotenie nie je schopné ohodnotiť otvorené otázky. Návrhom zákazníka bola možnosť opravovať otázky pomocou regulárnych výrazov.

Návrh

Cvičiaci, ktorý opravuje otázky si ich všetky prečíta a na záver napíše regulárny výraz, ktorý sa použije na ohodnotenie študentských odpovedí.

Riešenie

Do zoznamu odpovedí študentov k otázke bol pridaný prvok kde cvičiaci môže napísať regulárny výraz. Po stlačení tlačidla je zaslaný na server kde sa spracuje vo funkcii *submit_regexp*, ktorá určí stav odpovede ale nie obodovanie.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.2.2 Učiteľ si vie vybrať spôsob bodovania

Analýza

Pre predmet musí byť určené aké bodovanie bude použité na ohodnotenie otázok.

Je rozdiel medzi automatickým ohodnotením ako takým a obodovaním otázok.

Automatické ohodnotenie otázok rozhoduje iba o správnosti odpovede v rámci áno/nie. Bodovanie hodnotí v rámci reálnych čísel a berie do úvahy jednotlivé vyznačenia možností.

Návrh

Navrhujeme implementovať nový modul systému zvaný Scoring od, ktorého budú akékoľvek iné systémy bodovania dediť. Obodovanie testu bude volané pri ukončovaní testu.

Informácia o systéme bodovania bude obsiahnutá v databáze pre setup. Volanie správneho systému bodovania je spravené dynamicky.

Riešenie

Modul Scoring obsahuje funkciu *doScoringForExercise*, ktorá berie ako vstup id termínu, ktorý sa má obodovať. Táto funkcia volá nad každým testom študenta funkciu *doScoring*, ktorá berie ako vstupy aj id termínu aj id konkrétneho študenta.

V prípade implementovania vlastného systému skórovania je odporúčané implementovať iba funkciu *doScoring*.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.2.3 Cvičiaci si zobrazí konkrétny odovzdaný test jedného študenta

Analýza

Cvičiaci musí byť schopný zobrazíť konkrétny študentom odovzdaný test.

Je nutné prídanie zoznamu odovzdaných testov v rámci jedného cvičenia a v ňom si vybrať konkrétneho študenta.

Prezeranie testu musí minimálne poskytovať rovnakú funkcionálnosť ako prezeranie testu študentom.

Návrh

Obrazovku obsahujúcu zoznam odovzdaných testov pridať do časti aplikácie kde sa nachádzajú informácie o ukončenom teste.

Ako obrazovka zobrazujúca odovzdaný test sa môže použiť funkcionálnosť pre zobrazenie odovzdaného testu študentom.

Riešenie

V obrazovke ukončeného testu bola pridaná tlačidlo vedúce k zoznamu všetkých odovzdaných testov v rámci toho cvičenia.

Po kliknutí na prezrieť test je cvičiaci poslaný na tú istú akciu ako keby si študent prezeral vlastné odpovede, avšak id študenta je vložené ako parameter navyše a cvičiaci môže výsledky prezerať okamžite.

Testovanie

Bol otestovaný správny prístup zo študenta a cvičiaceho ako aj časy počas, ktorých je test nedostupný.

6.2.2.4 Systém rozpoznáva študentov s AIS rolou zamestnanca

Analýza

Na univerzite študujú aj takí študenti, ktorí zároveň pôsobia ako pracovníci na čiastočný úväzok napríklad v Centre výpočtových a komunikačných technológií. Prebratý systém autentifikácie využíva údaje z LDAPu, na základe ktorých je možné určiť rolu používateľa. Problémom sú však viaceré roli navzájom – možností je veľa, čo vyplýva z pokusov nad dátami LDAPu rôznych používateľov.

Návrh

Pre študentov ktorých sa to týka bola nájdená vhodná kombinácia parametrov z ich záznamov z LDAPu na základe ktorých je možné ich rozlišovať. Zamýšľané bolo aj nové zjednotenie pridelovania rolí učiteľov. Na to sa bude využívať administrátorská stránka systému a rola učiteľa sa prideluje priamo tam. Pre študentov sa kontrolujú parametre fiit-stud a fiit-zam. Ak má používateľ len záznam zamestnanca, je považovaný za učiteľa. Ak však má aj záznam fiit-stud, je považovaný za zamestnaného študenta.

Riešenie

Vo funkcii ldap_before_save bola upravená podmienka pridelovania rolí. Aktuálna rola pridelená v databáze sa mení len v prípade študentov – je možné že sa stanú učiteľmi. Pre ostatných je nutné meniť rolu len priamo v databáze a v zamýšľanej administrátorskej časti. Podmienka porovnáva parametre záznamu používateľa v univerzitnom ldape – riadi sa podľa navrhnutých pravidiel.

Testovanie

Testovanie prebiehalo využitím používateľských účtov študentov, pre ktorých platia analyzované obmedzenia, rovnako ako aj na tých, pre ktorých neplatia. Výsledkom je prihlásenie v roli študenta.

6.2.2.5 Učiteľ vie vyexportovať (aj ohodnotené) odpovede študentov

Analýza

Dáta získané z odpovedí študentov môže učiteľ využiť aj mimo systému a môže s nimi pracovať. Dáta môže využiť napríklad na tvorbu rôznych štatistických analýz, ktoré v systéme nie je možné urobiť.

Návrh

Na stránku testu je potrebné doplniť možnosť exportu odpovedí študentov. Ak bol test už ukončený, zobrazí sa mu na stránke tlačidlo s možnosťou exportu. Dáta sú stiahnuté do učiteľovho počítača vo formáte csv.

Riešenie

Do funkcionality systému bol doplnený pre ukončený test link na export dát. Export vyvoláva metódu určenú na spracovanie získaných odpovedí do nového formátu. Ten obsahuje informácie dôležité pre určenie správnosti odpovede a na identifikáciu prislúchajúcej otázky podľa jej referencie. Zobrazuje aj dôležité informácie o používateľovi, ktorý odpovedal – meno, prihlasovacie meno a informácie o termíne.

Testovanie

Testovanie bolo vykonané nad simulovanými testami, v ktorých odpovedal rôzny počet študentov, vrátane žiadnych. Výsledkom bol vyexportovaný csv dokument s počtom záznamov v závislosti na počte používateľov a počte otázok, ktoré bol jednému používateľovi vybraté.

6.2.2.6 Cvičiaci vidí len svoje cvičenia a Administrátor vidí všetky cvičenia s menami cvičiacich

Analýza

V zozname termínov sa v tabuľke uvádzajú informácie o plánovanom čase a týždni spolu s prechodmi na dôležité funkcie termínu (napr. spustenie testu). K termínom musí mať prístup aj administrátor. Ten z pozície garanta (vedúceho predmetu) môže pracovať s termínmi ktoréhokoľvek cvičiaceho. Vzhľadom na to, že sa tabuľka dosť rozširuje a je možné, že sa objavia aj termíny v rovnakom čase, je potrebné ich sprehľadniť a odlišiť podľa vedúceho (učiteľa) daného termínu.

Návrh

Do tabuľky termínov je potrebné doplniť ďalší stĺpec, ktorý v prípade prístupu so strany používateľa s rolou administrátora, zobrazuje meno a priezvisko vedúceho termínu. Administrátor tak dokáže odlišovať aj termíny, ktoré prebiehajú v rovnakom dátume a čase.

Riešenie

Na stránku zobrazenia termínov bol doplnený stĺpec obsahujúci meno a priezvisko cvičiaceho. Tento stĺpec sa používateľovi zobrazí len v prípade, že systém vyhodnotí používateľovu rolu ako rolu administrátora.

Testovanie

Prechádzanie na stránku zobrazenia zoznamu termínov bolo testované na používateľoch s rôznymi rolami. Študent k nej prístup nemá. Učiteľ má prístup k zobrazeniu tých termínov, ktoré mu boli pridelené. Administrátorovi sa zobrazujú všetky termíny aj s menami cvičiacich.

6.2.2.7 Učiteľ môže znovuotvoriť test na písanie

Analýza

Učiteľ môže spustiť a ukončiť test. V prípade, že test ukončí omylom, nemá možnosť ho znovuspustiť. Znovuotvorenie testu môže byť využité aj pre dopisovanie testov chýbajúcimi študentmi. Je potrebné, aby učiteľ mal túto možnosť.

Návrh

Na stránke spúšťania testov je možné test spustiť a v prípade, že bol spustený ukončiť. Doplniť je potrebné ďalší možný stav. Ak bol test už ukončený, musí byť možnosť znovuspustiť ho. Znovuotvorenie má byť k dispozícii na stránke zobrazenia informácie testu, rovnako ako aj v zozname termínov.

Riešenie

Pre určenie ukončeného času sa používa časový údaj `real_end`, ktorý do databázy zaznamenáva čas, kedy bol ukončený. Pre vrátenie pôvodného stavu, ktorý dovoľuje prístup k testom, bol tento čas vo funkcii znovuovertorenia testu odstránený, čím sa test dostal akoby do stavu po spustení, hoci čas spustenia ostáva pôvodný. Odovzdané testy zostávajú v platnosti a študenti nemôžu písať test druhýkrát. Funkcia `update` pre zmenu stavu testu bola doplnená o novú podmienku, ktorej obsah sa vykonáva v prípade už skončených testov. V zozname termínov, rovnako ako aj na stránke testu bolo doplnené tlačidlo na znovuspustenie, ktoré vyvoláva danú metódu.

Testovanie

Testované boli rôzne možnosti stavov v databáze. V závislosti na dostupnosti časových údajov `real_start` a `real_end` sa musela nasledujúca stránka testu zobraziť v správnom formáte v závislosti na stave. Znovuspustenie úspešne pracovalo v každom prípade, keďže jeho funkcia je len o odstránení jedného časového údajá zo stĺpca v databáze.

6.2.2.8 Cvičiaci zobrazí test na cvičení

Analýza

Cvičiaci zobrazí test na cvičení. Po prihlásení si cvičiaci môže zobrazíť testy pre svoje cvičenia. Z nich si následne vyberie konkrétny test a spustí ho, pričom študentom poskytne heslo pre vstup testu.

Návrh

Cvičiaci na cvičení spustí test pre svojich študentov. V zozname svojich cvičení si vyhľadá aktuálne cvičenie. Po zobrazení stránky cvičenia môže spustiť pre toto cvičenie určený test. Po spustení testu sa cvičiacemu zobrazia údaje, ktoré potrebujú študenti, aby sa dostali na stránku testu. Na test sa študenti môžu dostať použitím zverejnenej adresy alebo kódu cvičenia.

Riešenie

Do systému bola doplnená funkcionálna zobrazovania zoznamu cvičení. V zozname cvičení sa prirodzene objavujú len testy, ktoré patria danému cvičiacemu. Cvičiaci má k dispozícii všetky záznamy počas celého semestra. Výnimkou je administrátorská rola, ktorá má k dispozícii úplne všetky testy. Zo zobrazenia zoznamu testov je možné sa prekliknúť na konkrétne cvičenie stlačením tlačidla vstúpiť. Stlačením tlačidla sa cvičiaci presmeruje na stránku konkrétneho cvičenia. Na tejto stránke má možnosť test spustiť. Po spustení testu sa mu zobrazí informácia o tom, že test práve prebieha a dostane k dispozícii údaje, ako sa dostať k testu (priama adresa resp. kód cvičenia). Pri snahe o zobrazenie stránky testu, na ktorý nemá daný cvičiaci práva, sa zobrazí informácia o nepovolenom prístupe.

Testovanie

Spúšťanie testu bolo testované použitím viacerých testovacích používateľov z rôznymi rolami. Testoval sa prístup rôznych používateľov na stránku testu, rovnako ako aj overovanie možnosti opakovaného spustenia testu.

6.2.2.9 Cvičiaci ukončí test na cvičení

Analýza

Cvičiaci môže ukončiť svoje začaté a neukončené testy. Testy, ktoré neboli spustené sa nedajú ukončiť a testy, ktoré už boli ukončené nemôže ukončiť opätovne. Študenti sa nemôžu dostať na stránku už ukončeného testu.

Návrh

Cvičiaci si v zozname svojich cvičení vyhľadá test, ktorý chce ukončiť. Po prechode na stránku testu ho, v prípade, že bol spustený, môže ukončiť. Po ukončení testu už študenti nemôžu daný test písať. Cvičiaci

po ukončení testu nemôže test opätovne spúšťať alebo ukončovať. Po ukončení testu môže prejsť zo stránky testu len do sekcie hodnotenia testu.

Riešenie

Na stránke cvičenia sa k možnosti spustenia testu doplnila funkcionálna zobrazenia po spustení testu. Keď je test spustený, na obrazovke cvičenia sa zobrazuje informácia o prebiehajúcom teste spolu s prístupovými údajmi testu (adresa a kód cvičenia). Pod týmito údajmi sa nachádza tlačidlo ukončenia testu. Po kliknutí na toto tlačidlo sa test ukončí. Stránka cvičenia sa obnoví a na nej sa cvičiacemu zobrazí informácia o ukončení testu spolu s tlačidlom na prechod do sekcie hodnotenia testu.

Testovanie

Testovanie prebiehalo spolu s testovaním spúšťania testu. Používatelia rôznych rolí sa podľa stanovených kritérií nesmeli dostať k testom, ku ktorým nemali prístup, resp. pri študentoch k testom, ktoré neboli spustené, alebo boli už ukončené.

6.2.2.10 Cvičiaci si zobrazia zoznam svojich termínov cvičení

Analýza

Prihlásime sa ako administrátor alebo učiteľ. V navigačnom paneli prejdeme na stránku Učenie kde sa nám zobrazí zoznam termínov cvičení so základnými informáciami.

Pre administrátora sa zobrazia všetky cvičenia, pre cvičiaceho sa zobrazia iba tie cvičenia, ktoré sú jemu priradené.

Návrh

Navrhujeme vytvorenie novej stránky Učenie, ktorá bude obsahovať tabuľku cvičení, ktorá bude obsahovať niektoré základné informácie ako aj tlačidlo pre vstup do konkrétneho cvičenia.

Riešenie

Bol vytvorený nový kontrolér ako aj frontend pre novú stránku Učenie. Boli upravené navigačné panely aby zobrazovali adresu iba pre užívateľa s vyžadovanými rolami. Užívateľ bez vyžadovanej roly nemá do týchto častí prístup. Vytvorená tabuľka obsahuje poradové číslo logického týždňa, termín, stav a vstup do cvičenia.

Testovanie

Tabuľka je správne zobrazená pre naše testovacie dáta a reštrikcia vstupu na základe role bola otestovaná a vyhlásená za funkčnú.

6.2.2.11 Učiteľ a admin nemôžu písať test

Analýza

Prihlásime sa ako administrátor alebo učiteľ. Počas pokusu o vstúpenie do testu, ktorý bol otvorený hocikým, nás presmeruje na domovskú stránku so správou, pre používateľa, že k testom nemá prístup.

Písať test je schopnosť, ktorú má iba študent. Učitelia a administrátori túto schopnosť nemajú. Pôvodný systém Alef-TNG pracuje s gemami Devise a CanCan, ktoré umožňujú pridávať používateľom schopnosti a roly. Za ich pomoci sa dá nastaviť reštrikcia prístupu do častí systému na základe schopností a rolí.

Návrh

Náš návrh zahŕňa úpravu práv používateľov, tak aby role učiteľ a administrátor nemali prístup k stránke kde sa píše testy.

Riešenie

Upravili sme časť kódu zodpovednú za pridelovanie schopností rolám, tak aby učiteľ a administrátor nemali prístup k písaniu testu. Na druhú stranu sme túto schopnosť študentovi pridelili.

Bolo nutné upraviť kontrolér pre otázky tak aby metódy pre zobrazenie a odoslanie testu vyžadovali autorizáciu.

Testovanie

Po prihlásení sa pod rozličnými užívateľmi(administrátor, učiteľ, študent) sa testoval prístup na stránku testov. Administrátor a učiteľ boli presmerovaný na domovskú stránku s príslušnou správou. Študent bol schopný písať test.

6.2.2.12 Cvičiaci skontroluje dochádzku študentov na cvičení

Analýza

Po ukončení testu cvičiacim sú študenti, ktorí sa testu zúčastnili, t.j. tí, ktorí boli prítomní vypísaní v tabuľke spolu s ich AIS id a počtom správnych zodpovedaných otázok.

Návrh

Na základe záznamov o odpovediach na testové úlohy zistíme dochádzku.

Riešenie

Vytvorený nové čiastkové zobrazenie, ktoré sa renderuje ako súčasť stránky s informáciami o konkrétnom cvičení.

Zo záznamov o odpovediach patriacich konkrétnemu cvičeniu vyberáme unikátnych používateľov. Informácie o nich sú zobrazené v tabuľke.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.2.13 Učiteľ na domovskej stránke môže spustiť test

Analýza

Učiteľ je po prihlásení presmerovaný na stránku učenia, kde sú všetky cvičenia z iných týždňov ako dnešného skryté. Zobrazia sa až po rozkliknutí.

Bloky testov sú rozdelené na aktuálne cvičenia, cvičenia týždeň po a pred a ostatné cvičenia po a pred aktuálnym cvičením.

Návrh

Pri prihlásení na základe role používateľa presmerovať na jeho domovskú stránku.

Existujúcu tabuľku rozdeliť na sekcie podľa blokov cvičení tak ako boli opísané v analýze. Upraviť kód html tak aby boli tieto sekcie skrývateľné.

Riešenie

Na potrebné miesta pridané riadky, ktoré spúšťajú javascript. Tento javascript sa pozerá na vlastnosť id každého riadku, ktorá označuje číslo týždňa. Argumenty, ktoré určujú, ktoré týždne sa schovávajú sú v názve riadku spúšťajúceho javascript.

Po prihlásení používateľa sa kontroluje či sa jedná o učiteľa ak áno tak ho posiela na stránku učenia.

Testovanie

Pretože sa upravuje už existujúci kód, bola overená pôvodná funkcionálnosť.

6.2.2.14 Učiteľom sa bude objavovať online help

Analýza

Učitelia majú k dispozícii tabuľku ich cvičení/prednášok. V nej sa nachádza veľa informácií o konkrétnych termínoch. V tejto tabuľke sa taktiež nachádza veľa druhov ikoniek pre vykonanie rôznych akcií. Tie sú síce intuitívne, ale rozhodli sme sa, že v tejto úlohe k nim pridáme pomocné informácie pri prejdení myšou po jednotlivých ikonkách. Tým sme zabezpečili ešte zrozumiteľnejšie prostredie pre učiteľov.

Návrh

Učiteľ sa nachádza na stránke „Učenie“, v ktorej sa zobrazuje tabuľka jeho cvičení a prednášok. Pri priložení kurzora na konkrétnu ikonku (ktorej významu si nie je používateľ istý) sa zobrazí pri ikonke informačné okno s názvom ikonky, ktorý opisuje jej funkciu.

Riešenie

Implementovali sme to klasickými tooltipmi, ktoré sme aplikovali na všetky potrebné elementy.

Testovanie

Túto funkčnosť sme testovali na rôznych dostupných prehliadačoch, kde sme nenarazili na žiaden problém.

6.2.3 Modul študent

6.2.3.1 Študent si môže otvoriť test QR kódom

Analýza

Aby sme študentom umožnili rýchlejší a flexibilnejší prístup k testom, budú mať k dispozícii nielen vloženie kódu testu do textového poľa a priamy prístup vložení kódu testu ako linky, ale aj vygenerovaný QR kód. Na jeho generovanie použijeme dostupný gem `qr`. Tento kód bude k dispozícii na obrazovke s prebiehajúcim testom, kde sa nachádza aj kód cvičenia a linka. QR kód okrem uľahčenia prístupu k testu zároveň vyplní prázdne miesto na tejto obrazovke.

Návrh

Po spustení testu je učiteľ presmerovaný na obrazovku s prebiehajúcim testom. Túto obrazovku väčšinou majú študenti k dispozícii na cvičení, alebo prednáške ako projekciu na plátno. Na nej sa nachádza okrem iných informácií o teste aj vygenerovaný QR kód. Obsahuje v sebe zakódovanú adresu vrátane kódu cvičenia. Aby bol QR kód čo najlepšie čitateľný, a teda vyplňal zvyšok voľného miesta, je potrebné aby bol responzívny. Po ukončení testu je potrebné súbor s QR kódom vymazať z dôvodu uvoľnenia zdrojov.

Riešenie

Vytvorili sme generátor QR kódu použitím gemu `qr`. Kód sa po spustení testu vygeneruje do samostatného súboru, do priečinka s obrázkami, a jeho názov obsahuje príponu identifikátoru testu kvôli existencii viacerých súborov súčasne. Po presmerovaní na obrazovku prebiehajúcего cvičenia sa QR kód načíta zo súboru ako obrázok. Aplikovaním CSS pravidiel sme dosiahli jeho responzívny dizajn. Po ukončení testu sa súbor automaticky vymaže.

Testovanie

Počas testovania sme sa zamerali hlavne na otestovanie responzívneho dizajnu obrázku, správnosť informácii obsiahnutej v QR kóde a jeho automatické generovanie a vymazanie po ukončení testu. Počas testovania sme nezistili žiadne negatívne skutočnosti.

6.2.3.2 Študent si môže pozrieť svoje hodnotenie

Analýza

Pokiaľ sú odpovede na test už dostupné musí si byť študent schopný svoje odpovede prezrieť. Študent si musí vedieť vybrať test, ktorého odpovede chce vidieť. Študentovi musí byť vyznačená jeho odpoveď aj správna odpoveď.

Návrh

Vytvorenie novej stránky, ktorá obsahuje zodpovedané testy na, ktoré sa dá kliknúť. Po kliknutí sa študentovi zobrazí zoznam otázok a jeho odpovedí.

V rámci používania tých istých prvkov v rámci stránky je vyznačenie správnych, nesprávnych a študentových odpovedí rovnaké ako v iných častiach alefu.

Testovanie

Bolo vykonané základné modulové testovanie. Bol vytvorený automatický test.

6.2.3.3 Študent si môže pozrieť hodnotenie až po nakonfigurovanom čase

Analýza

Študenti by nemali byť schopní prezeráť si opravený test hneď po odovzdaní. Napríklad aby dávali pozor na hodine.

Návrh

Musí sa pridať informácia o nastavení po akej dobe budú opravené testy dostupné. Táto informácia by mohla byť definovaná pre každé cvičenie zvlášť alebo raz pre predmet.

Pred vpustením študenta sa skontroluje uplynutý čas od uzavretia testu ak je väčší ako nakonfigurovaný čas tak ho pustí ďalej, inak ho vráti.

Riešenie

Pridaný nový stĺpec do tabuľky pre cvičenia nazvaný *cooldown_time_amount*, ktorý značí čas v minútach po uplynutí ktorého je opravený test prístupný.

Testovanie

Funkcionalita bola ručne overená.

6.2.3.4 Študent sa môže zúčastniť viacerých rôznych testov v jednom týždni

Analýza

Charakter dátového modelu nepovoľoval rozličné sety otázok pre jeden týždeň. V praxi to znamenalo, že ak sa uskutoční test aj na prednáške aj na cvičení, nebolo možné zabezpečiť aby nedostal tie isté otázky.

Návrh

Navrhuje rozšírenie dátového modelu tak aby povoľoval concept prepojený na termín prepojením ManyToMany. Pôvodné prepojenie konceptov na týždne ostane zachované. Takto vzniknú dva koncepty, týždňové a termínové koncepty.

Riešenie

Implementácia návrhu si vyžadovala úpravu seedovacích skriptov, formátu csv súborov a recommendera, ktorý teraz je nastaviteľný aké koncepty má brať do úvahy.

Testovanie

Funkcionalita bola ručne overená.

6.2.3.5 Študent môže prísť na hociktoré cvičenie

Analýza

Študent môže písať test na hociktorom cvičení. Na test sa študent zaregistruje heslom, ktorý mu cvičiaci poskytne. Vie otvoriť test priamo adresu aj nepriamo preklikávaním sa v systéme a zadaním hesla.

Študenti nie sú priamo priradení na jednotlivé cvičenia. Táto funkcionalita vychádza z rôznych podmienok, ktoré môžu byť stanovené na niektorých predmetoch. Študent teda príde na cvičenie. Na ňom cvičiaci sprístupní test a oboznámi študentov, ako sa dostať na test (dá im k dispozícii adresu a kód cvičenia). Študent môže využiť na vstup do testu priamu adresu, ktorý zadá do prehliadača, alebo kód cvičenia, ktorý zadá do textového poľa na to určeného.

Návrh

Študent sa na cvičení prihlási do systému. V prípade, že cvičiaci spustil test a dal k dispozícii prístupové údaje pre test môže ich študent zadať do políčka adresy v prehliadači, alebo do textového poľa na vrchu

svojej študentskej úvodnej stránky. Textové pole sa v aplikácii objavuje len ak v danom čase, keď sa na stránku dostal existuje nejaký predpokladaný termín cvičenia, alebo je nejaký iný spustený. Po zadaní adresy alebo kódu testu sa študent dostane na stránku písania testu.

Riešenie

V riešení bola prepracovaná študentská úvodná stránka, ktorá obsahuje v základnej verzii len zoznam učiacich tém pre týždne, ktoré už počas semestra prešli, resp. týždeň, ktorý je aktuálny. Doplnené bolo zobrazovanie textového poľa, ktorého zobrazenie závisí od dostupnosti cvičení v čase, kedy si študent zobrazil svoju úvodnú stránku. V prípade existencie termínu, alebo otvoreného testu sa mu zobrazí toto textové pole, do ktorého je potrebné správne zadať kód cvičenia na ktorom sa nachádza (kód, ktorý mu bol daný k dispozícii cvičiacim). Po správnom zadaní údajov systém študenta pošle na stránku písania testu.

Testovanie

Testovanie prebiehalo formou simulovania otvárania a zatvárania rôznych testov, pričom simulovaní testovací študenti sa mali dostať na test zadaním adresy resp. kódu. Pri zadávaní kódu boli zadávané nesprávne hodnoty, hodnoty iných ešte nezačatých alebo naopak ukončených testov a podobne.

6.2.3.6 Študent po odoslaní odpovede nemôže test písať druhýkrát

Analýza

Študent môže písať daný test len raz. Po napísaní testu, odpovedaní na jednotlivé otázky a stlačení tlačidla odovzdaj, sa študent už nemôže dostať opäťovne na stránku písania testu.

Návrh

Študent odpovedá na rôzne otázky. O jeho odpovediach sa vedie v databáze záznam, ktorý je možné priradiť konkrétnemu cvičeniu. Pri snahe o prístup k testu sa prezerajú záznamy na prítomnosť vzťahu k danému cvičeniu, ktorého test sa snaží študent zobrazíť. V prípade, že sa takýmto spôsobom zistí existencia odpovedí, študentovi je zobrazená informácia o tom, že test nemôže písať viac ako raz.

Riešenie

Pri prechode na stránku testu, či už cez priamu adresu alebo využitím textového poľa so zadaným kódom sa v databáze overuje prítomnosť záznamov odpovedí na test tohto cvičenia. V prípade, že sa takéto záznamy v databáze nenachádzajú, študentovi sa zobrazí stránka písania testu. V opačnom prípade ho systém na stránku nepustí, pošle ho späť na úvodnú stránku a informuje ho oznámením o možnosti písať test len raz.

Testovanie

Testovanie prebiehalo za pomoci vytvárania záznamov v databáze, ktoré obsahovali testovacie odpovede študentov na konkrétny test. Následne sa testovala správnosť overovania existencie záznamov o odpovediach.

6.2.3.7 Študent odošle svoju odpoveď na test

Analýza

Študent vstúpi do stránky písania testu. Test vypracuje a stlačí tlačidlo odovzdaj test. Jeho odpovede sú zaslané na server kde sú spracované. Používateľ je presmerovaný na domovskú stránku.

Pôvodný systém Alef-TNG obsahuje funkcie pre opravu pôvodných typov otázok.

Návrh

Celý test je považovaný za jeden veľký formulár, preto je nutné navrhnúť vhodnú štruktúru odpovede aby sa dali jednoducho aplikovať už existujúce metódy.

V prípade otvorených otázok sa tieto označia ako odovzdané ale neopravené. Takéto odpovede musia byť opravené ručne, táto funkcionality nie je zatiaľ implementovaná.

Študentove odpovede a ich stav sú zapísané do databázy.

Riešenie

Odpovede sú zasielané v hashi, ktorý obsahuje odpovede na každú otázku na vlastnom indexe. Takto je jednoduché implementovať vlastnú variantu automatického opravovania testov.

V prípade otvorených otázok sa uloží študentova odpoveď do príslušného stĺpca v databáze, kde čaká na manuálne opravenie.

Testovanie

Testovanie prebiehalo pomocou porovnávania zasielaného hashu a zmien v databáze. Počas tohto procesu test pozostával z testovacích úloh, nakoľko ešte nebol implementovaný vhodný recommender.

6.2.3.8 Študent odpovedá na obrázkovú otázku

Analýza

Študentovi sa po otvorení testu zobrazia otázky, niektoré z nich môžu mať ako súčasť zadania obrázky.

Študent odpovie na otázku spôsobom, ktorý je typický pre typ otázky.

Po vpísaní svojej odpovede, študent môže pokračovať v riešení ďalších otázok.

Akýkoľvek typ otázky môže obsahovať obrázok v zadaní. Táto funkcionlita je podporovaná v pôvodnom systéme Alef-TNG sú obrázky uložené v tabuľke otázok ako binárna hodnota.

Návrh

Analyzovať ako funguje zobrazenie obrázkov v pôvodných častiach frontendu a analogicky implementovať túto funkcionlita do frontendu pre zobrazenie testu.

Riešenie

Pridané zobrazenie obrázku, ak existuje, do zobrazenia otázky v teste. Vytvorenie nazvanej cesty qimages, ktorá vráti obrázok pre zadaný týždeň a identifikátor otázky.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.3.9 Študent odpovedá na otvorené otázky

Analýza

Študentovi sa po otvorení testu zobrazia otázky, niektoré z nich môžu byť vo formáte otvorenej otázky, študent na ne musí byť schopný odpovedať.

Študent zadá do textového poľa svoju odpoveď, ktorá je naďalej editovateľná.

Po vpísaní svojej odpovede, študent môže pokračovať v riešení ďalších otázok.

V prípade, že užívateľ na otázku neodpovie, za jeho odpoveď sa považuje prázdny reťazec.

Návrh

Navrhujeme vytvorenie nového typu otázok, ktoré môžu byť zobrazené v teste vedľa pôvodných typov otázok. Tieto budú okrem textu otázky, obsahovať textové pole, ktorého obsah sa bude zasielať pri odovzdaní testu ako odpoveď.

Riešenie

Vytvorili sme triedu otvorených otázok, ktoré dedia vlastnosti triedy otázka. Vo frontend časti aplikácie bola pridaná vetva schopná vykresliť tieto otázky a odoslať ich textového poľa pri odovzdaní testu.

Testovanie

Testovanie funkčnosti prebiehalo na základe inšpekcie zasielaného hashu pri odovzdaní textu.

6.2.3.10 Študent odpovedá na výberové otázky

Analýza

Študentovi sa po otvorení testu zobrazia otázky, niektoré z nich môžu byť vo formáte výberovej otázky, študent na ne musí byť schopný odpovedať.

Existujú dva typy výberových otázok, otázka kde môže byť viacero správnych odpovedí a otázka kde je iba jedna odpoveď správna. Oboje typy výberových otázok sú implementované v pôvodnom systéme Alef-TNG.

Po zvolení odpovede, študent môže pokračovať v riešení ďalších otázok.

Návrh

Keďže tieto typy otázok už sú implementované, navrhujeme vytvoriť vlastnú vetvu vo frontende, ktorá ich bude správne zobrazovať v teste, na čo pôvodne neboli stavané (Alef-TNG zobrazuje otázky zásadne po jednom).

Riešenie

Upravil sa frontend testu tak aby bol schopný zobrazovať obe typy výberových otázok. Zobrazenie výberových otázok v teste je postavené na základe zobrazenia v pôvodnom systéme Alef-TNG. Pôvodné zobrazenie ostalo nezmenené aby sa zachovala funkcionálnosť pôvodného systému.

Testovanie

Testovanie funkčnosti prebiehalo na základe inšpekcie zasielaného hashu pri odovzdaní textu.

6.2.3.11 Študent si v rámci testu zobrazí pridelené otázky

Analýza

Študentovi sa po vložení správnej adresy, zobrazí stránka s testom. Táto stránka musí zobrazovať všetky otázky.

Otázky v pôvodnom systéme Alef-TNG sú zobrazované v zozname iba s textom otázky a až po kliknutí na kartu otázky sa prechádza na novú stránku, ktorá obsahuje otázku, možnosti, komentáre, tlačidlá.

Návrh

Náš návrh vyžaduje jednu stránku so všetkými otázkami v plnom tvare bez tlačidiel a komentárov a informácií koľkokrát bola otázka videná.

Na základe týchto požiadaviek navrhujeme rozšírenie frontendu pre testy o schopnosť zobraziť spoločné vlastnosti rozličných typov otázok a zobraziť špecifické vlastnosti na základe typu otázky.

Riešenie

Frontend bol rozšírený o cyklus, ktorý zobrazuje otázky. Otázka pozostáva z textu, nepovinného obrázku a špecifickej časti, ktorá pozostáva z komponentov slúžiacich na vybranie či napísanie odpovede.

Celý test je považovaný za jeden formulár. Na konci testu sa nachádza tlačidlo odovzdaj test.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.3.12 Študent si zobrazí aktuálny test

Analýza

Študent po prihlásení do systému prejde na zadá adresu, ktorú zobrazil cvičiaci po spustení testu. Táto cesta ho zoberie na stránku s testom.

V prípade zle zadanej cesty je študent presmerovaný na stránku s chybovou hláškou 404.

Návrh

Návrh pozostáva z vytvorenia novej stránky, ktorá bude obsahovať test na vhodnej adrese. Adresa bude obsahovať číslo týždňa a kód cvičenia.

Riešenie

Bola vytvorená nová metóda v kontroléri pre otázky, ktorá slúži na zobrazenie stránky testu. Táto metóda je mapovaná na adresu, ktorú musí zadať študent. Adresa má tvar `/test/(cislo_tyzdna)/(kod)`.

Bol vytvorený frontend, ktorý v budúcnosti bude zobrazovať test.

Testovanie

Bolo vykonané základné modulové testovanie.

6.2.3.13 Všetci študenti dostanú špeciálnu zhodnú otázku alebo zhodné otázky na teste

Analýza

V niektorých prípadoch sa vyžaduje doplnenie pridelovania otázok pre test podľa ďalších špecifických kritérií. Okrem čisto náhodného pridelenia otázok môže byť vyžadované, aby sa niektoré otázky vybrali pre všetkých študentov, najmä ak má otázka vysokú prioritu, je dôležitá, aby všetci študenti vedeli na ňu odpovedať. V takom prípade sa výber otázok musí upraviť tak, že okrem náhodného výberu otázok doplní množinu vybraných otázok aj o tieto špeciálne otázky.

Návrh

V konkrétnom týždni sa vyberajú otázky podľa princípu popísaného pri úlohe: Študent dostane otázku podľa zadaného kľúča. Pri existencii špeciálnych otázok však množina vybraných otázok musí obsahovať aj

tieto otázky. Medzi vybrané otázky sa teda pridávajú špeciálne otázky a zvyšné otázky na doplnenie celkového počtu otázok sa vyberú náhodne.

Riešenie

Pre recommender popísaný v úlohe: Študent dostane otázku podľa zadaného kľúča bolo potrebné doplniť správu a pridelovanie špeciálnych otázok. Špeciálne otázky sa od bežných otázok odlišujú v identifikátore špeciálnych otázok. Ten určuje, či je otázka z množiny špeciálnych alebo nie. Recommender vyberá najskôr všetky existujúce otázky, ktoré spadajú do množiny špeciálnych a následne zvyšné doplní o testové otázky pre daný týždeň, ktoré nespádajú do tejto množiny. Študent dostane po spustení testu vždy všetky existujúce špeciálne otázky daného týždňa, v ktorom sa píše test.

Testovanie

Odporúčanie otázok so špeciálnymi otázkami bolo testované podobne ako štandardný recommender. Testovanie sa doplnilo o overenie výberu všetkých dostupných špeciálnych otázok pre test.

6.2.3.14 Študent dostane otázku podľa zadaného kľúča

Analýza

Študent na cvičení po spustení testu dostane otázky. Tie musia spĺňať niekoľko kritérií. Musia byť z množiny testových otázok pre daný týždeň, respektíve pre konkrétne cvičenie. Študentovi sa prideluje konkrétny počet otázok k . Týchto k otázok sa vyberá z množiny otázok pre cvičenie m (platí samozrejme, že k je menšie alebo rovné m). Otázky pre cvičenie sa vyberajú zo všetkých testových otázok pre daný týždeň, tie označme n . Vzťah pre počty otázok je $k \leq m \leq n$. Otázky pre cvičenie sú vždy zhodné a vyberú sa vždy zhodne, bez ohľadu na študenta. Študent následne vypracuje test, ktorého obsahom sú jemu pridelené otázky.

Návrh

V konkrétnom týždni existuje určité množstvo testových otázok. Z nich sa pre rôzne cvičenia vyberá konkrétny počet. Pri každom výbere otázok pre jednotlivé cvičenie sa vygenerujú tie isté otázky. Z nich sa následne vyberajú otázky pre jednotlivých študentov, ktorí na danom cvičení píše test. Študent po spustení testu dostáva na svojej obrazovke výstup v podobe zadania testu, ktorý obsahuje jednotlivé otázky vybrané na základe kritérií spomenutých vyššie. Otázky pre test sa nemenia ani pri obnovení stránky testu (refresh). Test môže študent následne vypracovávať.

6.2.3.15 Študent odpovedá na test na mobile

Analýza

Jednou z najväčších výhod našej aplikácie je jej responzivnosť na rôzne typy zariadení. V tejto úlohe sa táto responzivnosť na konkrétnych obrazovkách aplikácie pridávala, upravovala a testovala tak, aby bola čo najlepšia.

Riešenie a návrh

Počítačným bodom, sú všetky stránky aplikácie, ktorými študent prechádza pri procese písania testov. Cieľom sú správne zobrazené stránky na všetkých (hlavne mobilných) zariadeniach.

Tento problém sme vyriešili vhodným použitím CSS, ktoré umožní správne zobrazenie stránok aplikácie na rôznych zariadeniach.

Testovanie

Testovali sme túto pridanú funkcionálnosť na rôznych typoch mobilných zariadení, tabletov, klasických počítačov a taktiež aj v rôznych prehliadačoch. Zistili sme, že sa naša aplikácia zobrazuje veľmi dobre vo všetkých situáciách.

Riešenie

Do aplikácie bol doplnený recommender, ktorý má za úlohu vyberať otázky podľa spomenutých kritérií pre študenta. Recommender vyberá otázky pomocou náhodného generovania čísel. Generovanie rovnakých otázok pre cvičenie pri každom behu recommendera je pri generovaní otázok pre cvičenie zabezpečené úvodným inicializovaním funkcie náhodného výberu čísel pomocou vstupného kódu testu. Otázky pre študenta sa vyberajú obdobne, za použitia inicializácie využívajúcej id používateľa (študenta). Generované čísla sa mapovali na indexy v poli testových otázok týždňa.

Testovanie

Testovanie prebiehalo priradovaním rôznych hodnôt potenciálnych cvičení, pre ktoré sa vždy očakávala rovnaká postupnosť čísel určujúca výber otázok pre cvičenie. Pre rozdielne testy sa pritom môžu vyskytnúť zhody otázok. Vybraté otázky boli vždy z rozsahu možných otázok. Pre študenta sa otestovalo obnovenie stránky, pri ktorom boli študentovi vždy pridelené rovnaké otázky. Pri splnení všeobecných kritérií o veľkosti množín otázok sa recommender správa podľa očakávaní.

6.2.3.16 Študentovi sa zobrazí upozornenie pri zlom zadaní testovacieho kľúča

Analýza

V tejto úlohe sa pridala funkcionálnosť upozornenia používateľa (študenta) pri chybnom zadaní testovacieho kľúča. Toto upozornenie je vytvorené ako upozorňujúce okno s chybovou hláškou. Toto okno má vzhľad ladiaci s témou celej aplikácie a samo zmizne po pár sekundách, aby ďalej nevyrušovalo používateľa.

Návrh

Začiatkom tohto používateľského príbehu, je zadanie zlého testového kľúča študenta. Po kliknutí na potvrdenie kódu sa tento kód skontroluje. Keďže je neplatný, zobrazí sa používateľovi spomínaná upozorňujúca hláška o zlom testovom kľúči.

Riešenie

Túto úlohu sme riešili tak, že sme skontrolovali platnosť kódu. Keď je kľúč zadaný študentom správny, presmerujeme ho na konkrétny test. Ak ale taký kľúč nepatrí žiadnemu testu, tak sa použije upozorňovacie okno oznamujúce chybovú situáciu.

Testovanie

Túto funkcionality sme riadne otestovali na rôznych prípadoch zadávania kódov na rôznych prehliadačoch a stále to fungovalo bez problémov.

6.2.3.17 Systém informuje používateľa o zmenách stavu

Analýza

V našej aplikácii chceme, aby používateľ stále vedel, kde je a či aplikácia vykonala všetky operácie, ktoré v nej spravil. Preto sme v tejto úlohe implementovali upozorňujúce hlášky, ktoré používateľa informujú o akciách, ktoré vykonal. Tie sú v tvare notifikácií, ktoré po určitom čase samé elegantne zmiznú.

Návrh

Používateľ vykoná funkciu, na ktorú chce získať potvrdenie, že sa vykonala správne. Napríklad odhlásenie, prihlásenie, odovzdanie testu... Keď takúto činnosť vykoná, systém mu zobrazí upozorňujúcu správu o úspešnosti danej operácie. To isté platí pri chybovom zaobchádzaní s aplikáciou. To znamená napríklad zlé prihlásenie, nesprávne zadanie testového kľúča... V tom prípade sa používateľovi taktiež zobrazí potrebná upozorňujúca správa hlásiaca chybovú udalosť.

Riešenie

Tento problém sme riešili za pomoci upozorňujúcich okien. Použili sme ich rovnaký štýl, ako obsahoval aj pôvodný systém Alef. Všetky tieto správy sme lokalizovali to jedného súboru, pre každý jazyk a tým sme zjednotili prístup k lokalizácií upozornení. To nám uľahčí prácu pri zmenách jazykov a pridávaní nových upozornení.

Testovanie

Túto funkcionality sme testovali na všetkých možných prípadoch, kde sa táto funkcionality nachádza. Všetko funguje tak, ako je potrebné.

6.2.3.18 Systém upozorní pri odovzdávaní študenta, ak neodpovedal na všetky otázky

Analýza

Z podnetu od študentov, ktorí našu aplikáciu používali, sme zistili, že sa študentom občas podarí omylom odoslať test s tým, že nevyplnia všetky testové otázky. Aby sme tomu zabránili, pridali sme v tejto úlohe funkcionality, ktorá kontroluje, či študent odpovedal na všetky otázky.

Návrh

Študent sa nachádza na stránke testu. Ak odpovie na všetky otázky a klikne na tlačidlo odovzdania testu, test sa bez problémov odošle. V prípade, že študent nevyplní aspoň jednu otázku v teste, zobrazí sa mu po stlačení odovzdania testu dialógové okno. Upozorní študenta na nezodpovedanie všetkých otázok. Ak študent chce aj napriek tomu odovzdať test, odklikne možnosť v dialógovom okne, ktorá test odošle. Ak ale test nechce v takomto stave odoslať, klikne na príslušnú možnosť v dialógovom okne a dostane sa späť na jeho test.

Riešenie

Implementácia tejto funkcionality spočívala v kontrolovaní všetkých testovacích otázok resp. ich odpovedí. Kontrolujeme, či študent odpovedal na všetky (rôzne typy) otázok. Ak áno, testové odpovede uložíme do databázy. Ak nie, použijeme dialógové okno, ktoré sa pýta používateľa, či chce tento test aj napriek nevyplneniu všetkých otázok odovzdať. Ak chce, odpovede taktiež uložíme. V prípade, že nie, tak používateľa nepresmerujeme a len zavrieme dialógové okno, pričom používateľ môže pokračovať v písaní jeho testu.

Testovanie

Túto funkcionality sme testovali na rôznych prehliadačoch a zariadeniach a nenašli sme žiadny problém vo fungovaní.

7 Používanie prototypu – testovanie reálnymi používateľmi

Táto kapitola opisuje používanie a reálne dáta z prototypu použitého na predmete Operačné systémy.

6.1 Pribeh testovania vedomostí študentov

Používanie prototypu je spojené s testovaním vedomostí.

1. Vyučujúci začína prihlásením sa a vygenerovaním prístupového kódu
2. Vyučujúci následne zobrazí kód pre prístup študentom
3. Študent po prihlásení zadá zobrazený prístupový kód
4. Študent odpovedá na otázky
5. Študent odovzdáva odpovede
6. Vyučujúci ukončuje test a zobrazuje štatistiku a odpovedá na otázky testu

6.2 Obrazovky prototypu – študent

≡ MENU Prototyp

Operačné systémy

Prihlásenie

Zadaj svoje prihlasovacie údaje z AIS

AIS login

heslo

AIS Login

Vitajte v našej aplikácii pre Operačné systémy 1.0

Otestuj sa a nazbieraj body!

je to strašná zábava

Zadaj kód cvičenia

kód

Over a prejdí na otázky

Vitajte v našej aplikácii pre Operačné systémy 1.0

Otestuj sa a nazbieraj body!

je to strašná zábava

Zadaj kód cvičenia

60377119

Over a prejdí na otázky

Otázka číslo 1.

Vyberte kľucovy pojem, pre ktory je dane tvrdenie typicke.
Hlavnym cieľom je udržať procesor v stave "busy" počas doby prítomnosti procesov pripravených na vykonanie.

- Multiprocessing
- Multitasking
- Multithreading
- Multiprogramming
- Žiadne z uvedených

Otázka číslo 2.

Otázka číslo 3.

Otázka číslo 4.

Odovzdaj

Otázka číslo 1.

Otázka číslo 2.

Otázka číslo 3.

Otázka číslo 4.

Vyjadrite svoj vzťah k aplikácii prototypu TESA


- tesim sa ked ju opat uvidim
- uz by som bez nej nedokazal zit
- slovami sa vyjadrit neda
- chcem si s nou stale pisat
- s nou je moj den krajsi

Odovzdaj

☰ MENU Otázky pre 8. týždeň

Otázka číslo 1.

Otázka číslo 2.



The page at https://tesa.fiit.stuba.sk says:
Úspešne si odovzdal test a automaticky si sa odhlásil.

OK

- slovami sa vyjadrit neda
- chcem si s nou stale pisat
- s nou je moj den krajsi

Odovzdaj

☰ MENU

Operačné systémy

< O nás

< Otázky

< Oblasť as

Σ

nihol 2IA

WVCSBJKOV

olserl

.....

■

Vitajte v našej aplikácii pre operačné systémy

1.0

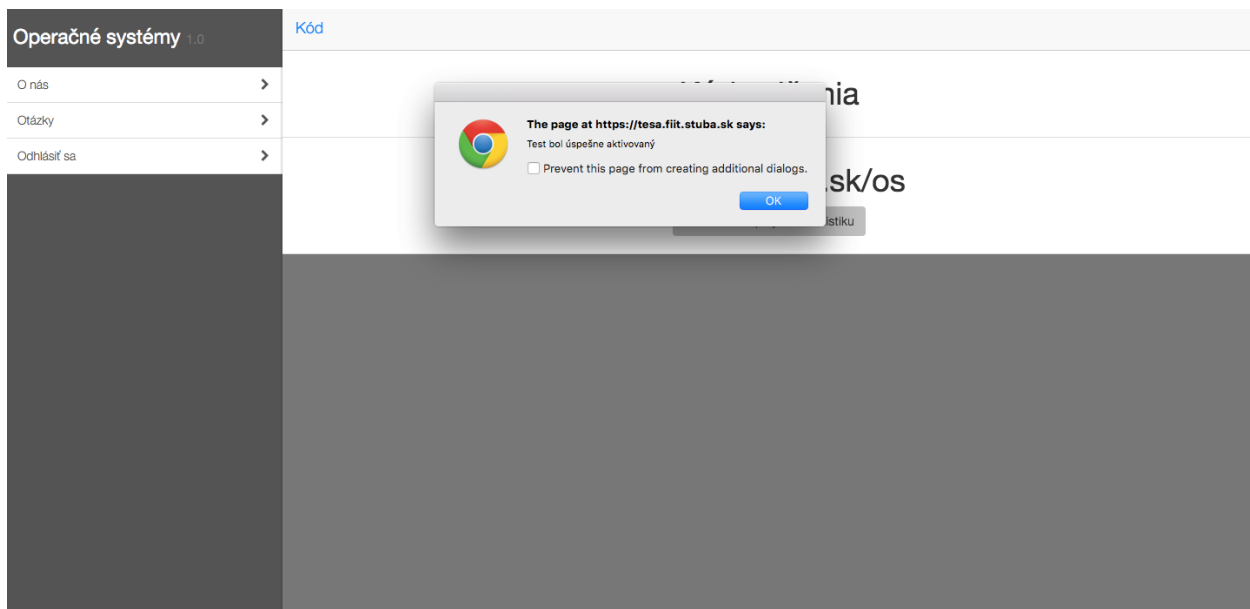
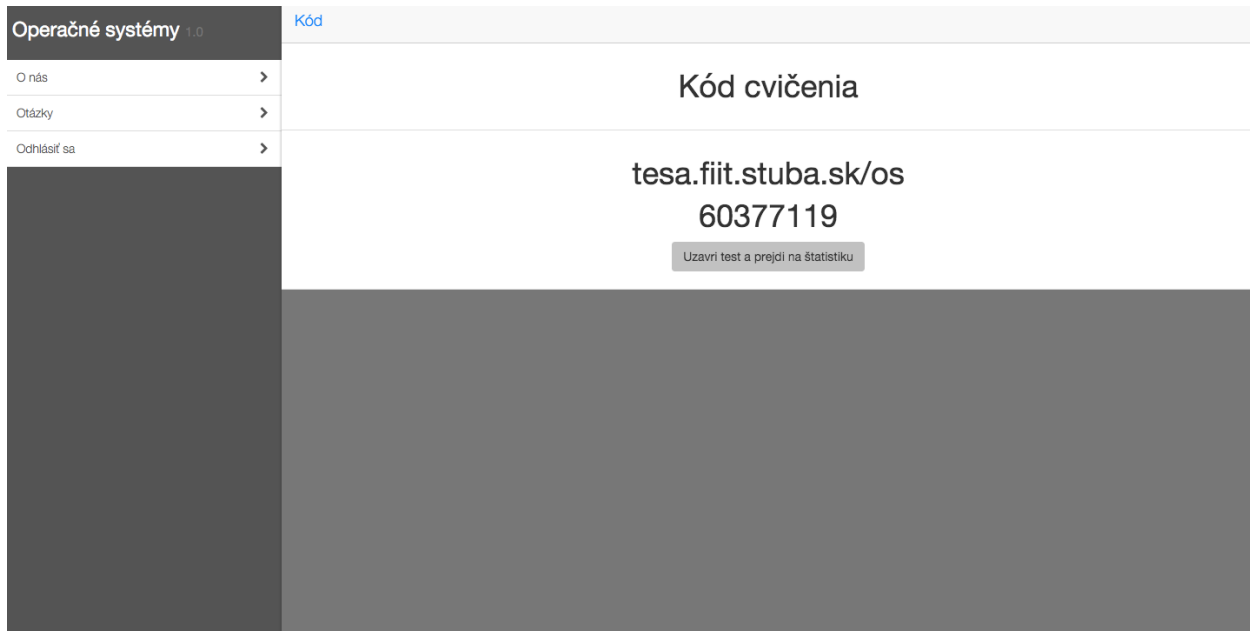
História prototypu

Tento prototyp bol vytvorený prvotne špeciálne pre účely predmetu Operačné systémy. Neskôr sa jeho poslanie zmenilo a stal sa z neho funkčný prototyp pre oficiálny produkt tímového projektu tímu 17 (Osičky) pre školský rok 2015/2016.

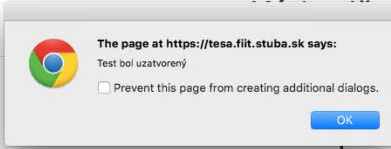
Stvoritelia

O vytvorenie sa postarali tri vtedy ešte len osie larvičky : Lukáš Csóka, Silvia Macejková a Roman Pikna, pod dohľadom sršňa Ing. Branislava Steinmüllera. Tento produkt mal a má aj napriek jeho muškám (možno aj vďaka nim) pre nich veľký význam. Bol vytvorený s láskou a s cieľom vylepšiť a zjednodušiť súčasnú situáciu cvičení.

6.3 Obrazovky prototypu – Vyučujúci



- O nás >
- Otázky >
- Odhlásiť sa >

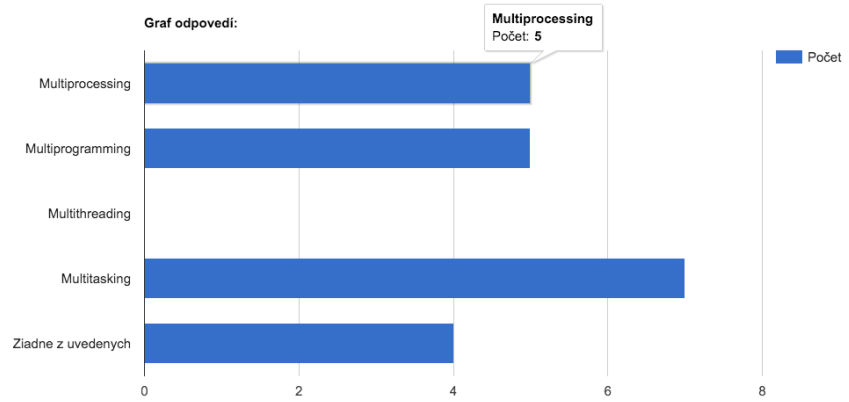


Uzavri test a prejdi na štatistiku

- O nás >
- Otázky >
- Odhlásiť sa >

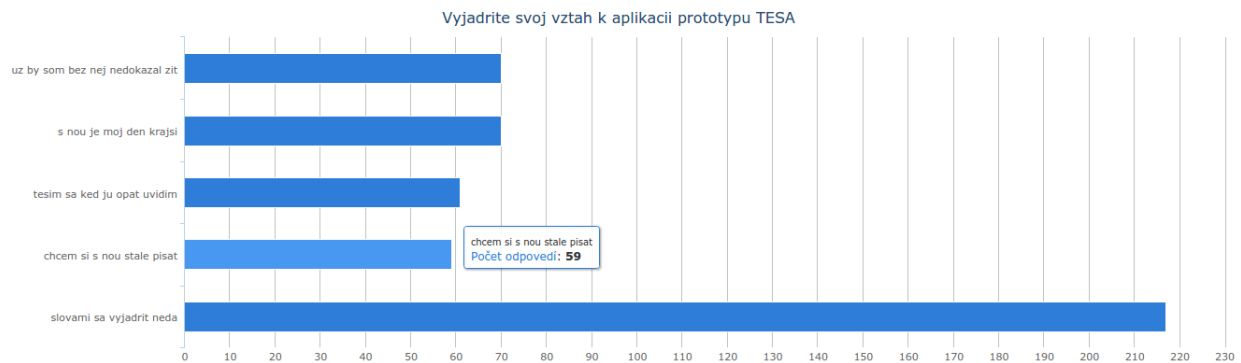
Otázka 1 Otázka 2 Otázka 3 Otázka 4

Vyberte kľucovy pojem, pre ktorý je dane tvrdenie typicke. Hlavnym cieľom je udržať procesor v stave "busy" počas doby prítomnosti procesov pripravených na vykonanie.



6.4 Štatistiky

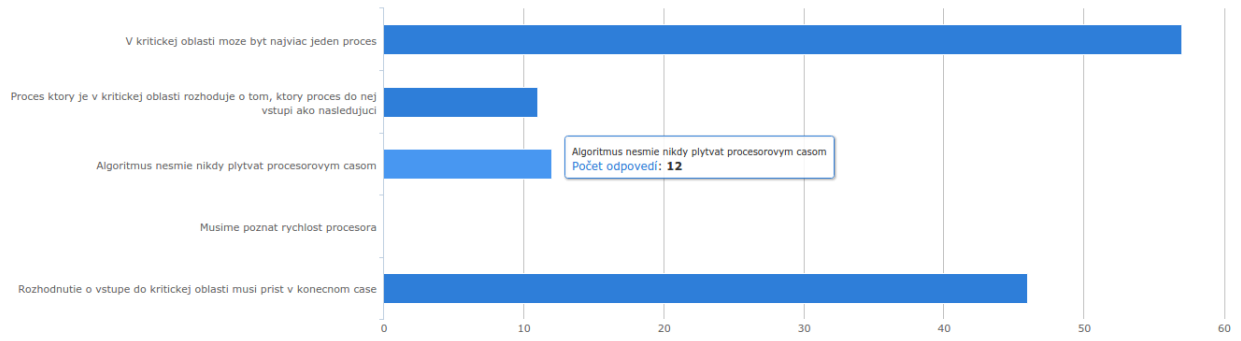
Nasledovné štatistiky pochádzajú z konkrétnych odpovedí študentov na testoch.



Doplňte spravne do tvrdenia: Pri pouziti algoritmu vyberu obeť LRU je z hl. pamate odstranena _____ stranka.

Počet	Text odpovede
2	nepouzivana
2	posledna
2	najmenej pouzivana v blizkej minulosti
1	najdavnejsie pouzita
1	najpouzivanejsia
1	najstarsie pouzita
1	Posledna nedavno pouzita
1	Prva
1	najdlhsie nevyuzitu
1	poskodena
1	Nepouzivana
1	najdlhsie nepuzita
1	posledna pouzita
1	najmenej pouzivatelná
1	aktuálna
1	naposledy pouzita
1	Pracovna

Vyberte nutne podmienky na riesenie vzajomneho vylucovania



Príloha A: Príručky

V nasledovnej prílohe sú príručky pre import dát, používateľov a administráciu.

Používateľská príručka

Používanie našej aplikácie je navrhnuté tak, aby bolo čo najintuitívnejšie. Taktiež má minimálne požiadavky na hardvér a softvér. Je podporovaná všetkými dostupnými prehliadačmi s podporou AJAX-u. Je taktiež responzívna, čo znamená, že sa dá používať na rôznych zariadeniach s rôznymi rozlíšeniami (mobilné zariadenia, tablety, počítače...). V tejto príručke opíšeme pridanú funkcionálnosť v našom projekte TESA oproti pôvodnému systému – Alef.

Naša aplikácia momentálne slúži primárne na testovanie študentov. Študenti a ich učitelia predstavujú naše základné používateľské role. Každá z týchto rolí má iné právomoci a používateľskú funkcionálnosť. Preto sa naša používateľská príručka skladá z dvoch základných častí – študentskej a učiteľskej.

Používateľská príručka pre študenta

Pri študentoch sa počíta za primárne používateľské zariadenie našej aplikácie mobilné zariadenie. Preto aj ukážky z nej budú práve v tomto rozlíšení. Nasledujúci - základný prípad použitia dokážu študenti ovládať bez akejkoľvek inej znalosti o aplikácii.

Napísanie a odovzdanie testu



Alef

Prihláste sa AIS údajmi.

Prihlasovacie meno

aisID

Heslo

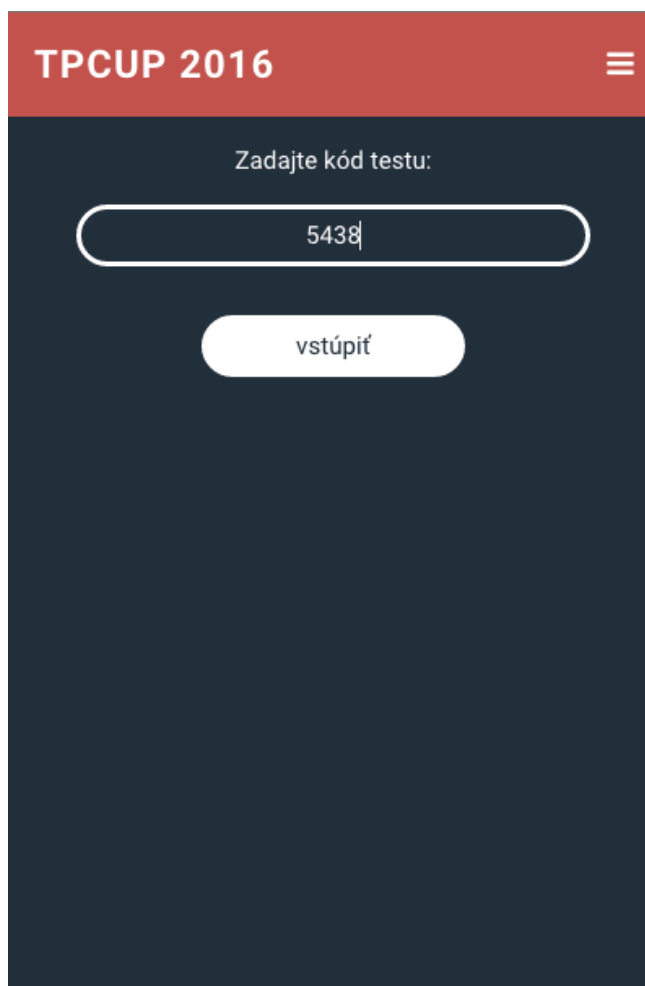
.....

Zapamätať prihlásenie

Prihlásiť

Obr. 1: Prihlasovací formulár

Na Obr. 1 je zobrazená prvá obrazovka (za predpokladu, že používateľ ešte nie je prihlásený) po otvorení inštancie našej aplikácie. Je ňou prihlasovací formulár. V ňom je na úspešné prihlásenie potrebné vyplniť prihlasovacie údaje zo systému AIS.



The image shows a mobile application interface for 'TPCUP 2016'. At the top, there is a red header bar with the text 'TPCUP 2016' in white and a white menu icon on the right. Below the header, the background is dark blue. The text 'Zadajte kód testu:' is centered in white. Below this text is a white rounded rectangular input field containing the number '5438'. Underneath the input field is a white rounded rectangular button with the text 'vstúpiť' in dark blue.

Obr. 2: Zadanie kódu

V prípade úspešného prihlásenia študenta môže pokračovať v používaní aplikácie. Ak v čase prihlásenia je k dispozícii aspoň jeden test, študent je hneď po prihlásení presmerovaný na stránku zobrazenú na Obr. 2 (ak nie je dostupný žiadny test, študent je presmerovaný na stránku „Učenie“). Používateľ má k dispozícii textové okno, do ktorého môže napísať kód testu, ktorý má k dispozícii na cvičení/prednáške, na ktorej sa nachádza. Tlačidlom „vstúpiť“ potom tento kód potvrdí na vstup do konkrétneho testu.

Študent sa môže dostať k zobrazeniu a písaniu konkrétneho testu aj pomocou odfovtenia QR kódu alebo zadaním konkrétnej URL adresy testu. Tieto informácie sú zobrazené učiteľom pri spustení testu (Obr. 9).

TPCUP 2016 - test

Ktorý z nasledujúcich tímových projektov/tímov vyhrá?

- Tesa
- Tím 17**
- Osičky
- tento

Napište vaše pocity a nápady, ako vylepšiť náš projekt.

Sem napíš svoju odpoveď

TPCUP 2016 - test

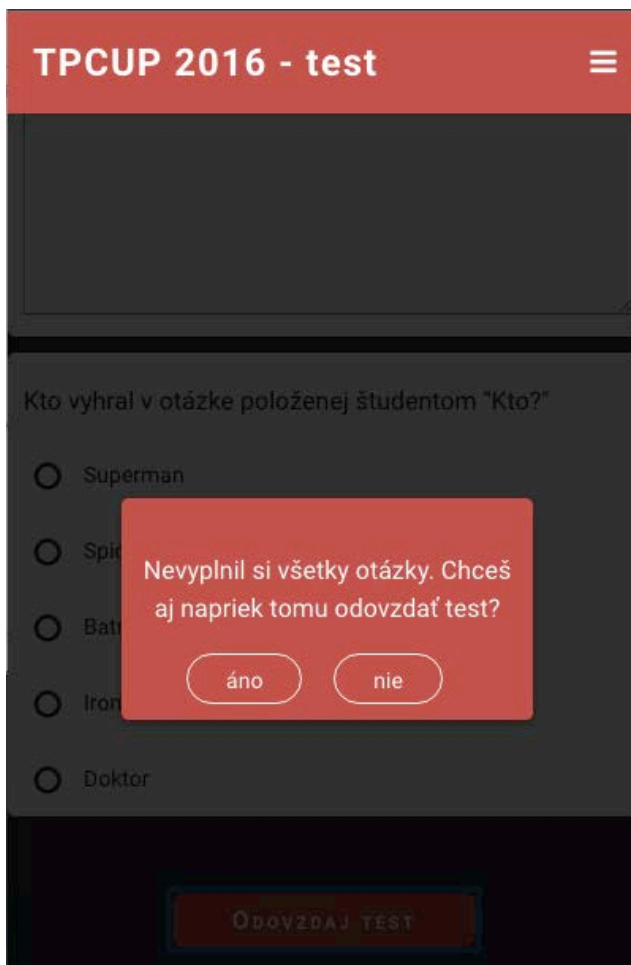
Kto vyhral v otázke položenej študentom "Kto?"

- Superman
- Spiderman
- Batman
- Ironman
- Doktor

ODOVZDAJ TEST

Obr. 3: Písanie testu

Keďže študent zadal správny kód z cvičenia/prednášky, na ktorej sa nachádza, bol presmerovaný na stránku testu ukázanú na Obr. 3. Test s jednoduchosťou vyplní a tlačidlom „odovzdaj test“ svoj test odovzdá. V prípade, že nevyplnil všetky testové otázky, mu bude zobrazené nasledujúce okno pre potvrdenie alebo odmietnutie odovzdania, zobrazené na Obr. 4. Tento kontrolný bod slúži k prevencii proti neúmyselnému odovzdaniu nevyplneného testu.



Obr. 4: Kontrola nevyplneného testu



Obr. 5: Úspešné odovzдание testu a prechod na obrazovku „Učenie“

Po úspešnom odovzdaní testu je študent presmerovaný na stránku „Učenie“ s prehľadom týždňov a pôvodnou funkcionalitou systému Alef. Tá je zobrazená na Obr. 5. Tým sa končí tento priamočiary prípad použitia.

Zobrazenie hodnotenia testov

Na zobrazenie hodnotenia testov sa dostaneme cez menu v hornej lište. To sa otvorí po kliknutí na tlačidlo vyznačené zeleným štvorcikom v prvej časti Obr. 6. Menu je dostupné z ktorejkoľvek stránky aplikácie. V ňom si vyberieme možnosť „Odobzdané testy“ – zobrazené v druhej časti Obr. 6.

The screenshot displays the TPCUP 2016 application interface. On the left, a red header bar contains the text "TPCUP 2016". Below it, a list of weeks is shown, with the 10th week highlighted as "AKTUÁLNY". A blue arrow points to a red square menu icon in the top right of the header. On the right, a dark grey sidebar menu is open, showing the user's name "Silvia Macejkova" and several options: "Učenie", "Spustiť test", "Odobzdané testy" (highlighted), "Spätná väzba", and "Odhlásiť". Below the sidebar, the same week list is repeated, also with the 10th week highlighted.

Week	Period
10. TÝŽDEŇ (AKTUÁLNY)	9.5. - 15.5.2016
9. TÝŽDEŇ	2.5. - 8.5.2016
8. TÝŽDEŇ	25.4. - 1.5.2016
7. TÝŽDEŇ	18.4. - 24.4.2016
6. TÝŽDEŇ	11.4. - 17.4.2016
5. TÝŽDEŇ	4.4. - 10.4.2016
4. TÝŽDEŇ	28.3. - 3.4.2016
3. TÝŽDEŇ	21.3. - 27.3.2016

Obr. 6: Otvorenie menu

Po výbere „Odobzdané testy“ položky v menu sa zobrazí stránka (prvá časť Obr. 7) obsahujúca zoznam odobzdaných testov konkrétneho prihláseného študenta. Hodnotenie konkrétneho testu zo zoznamu sa zobrazí, len ak ho učiteľ sprístupnil. Po výbere konkrétneho sprístupneného testu študentom sa zobrazí hodnotenie tohto testu (druhá časť Obr. 7).

TPCUP 2016	
1. TÝŽDEŇ - TEST(NEDOSTUPNÝ)	7.3. - 13.3.2016
1. TÝŽDEŇ - TEST	7.3. - 13.3.2016

TPCUP 2016 - hodnotenie testu

Napíšte vaše pocity a nápady, ako vylepšiť náš projekt.

Nie sú podporované automatickým hodnotením. Otázka sa hodnotí ručne.

Kto vyhral v otázke položenej študentom "Kto?"

- Superman
- Spiderman
- Batman
- Ironman
- Doktor

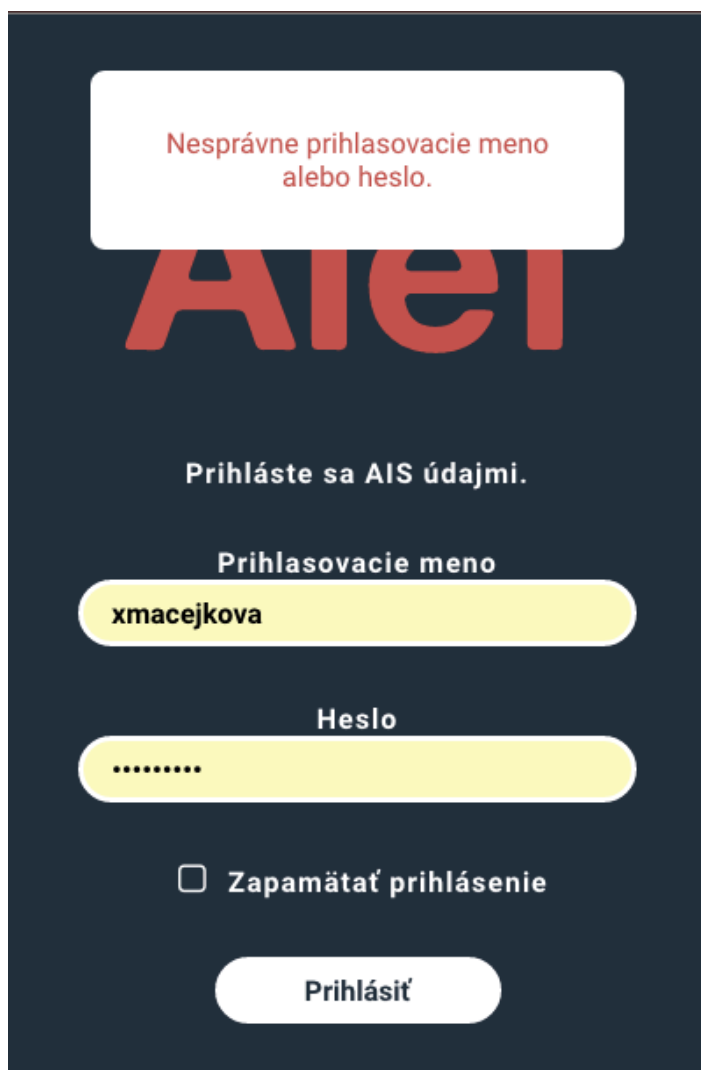
Obr. 7: Výber a zobrazenie hodnotenia testu

Hodnotenie zobrazené v druhej časti Obr. 7 ukazuje test po automatickom hodnotení systémom. Keďže tento test obsahuje aj otvorenú otázku s písaním voľného textu, je hodnotenie tejto otázky ešte nedostupné. Bude dostupné až po manuálnom hodnotení učiteľom.

Špeciálne prípady pre študenta

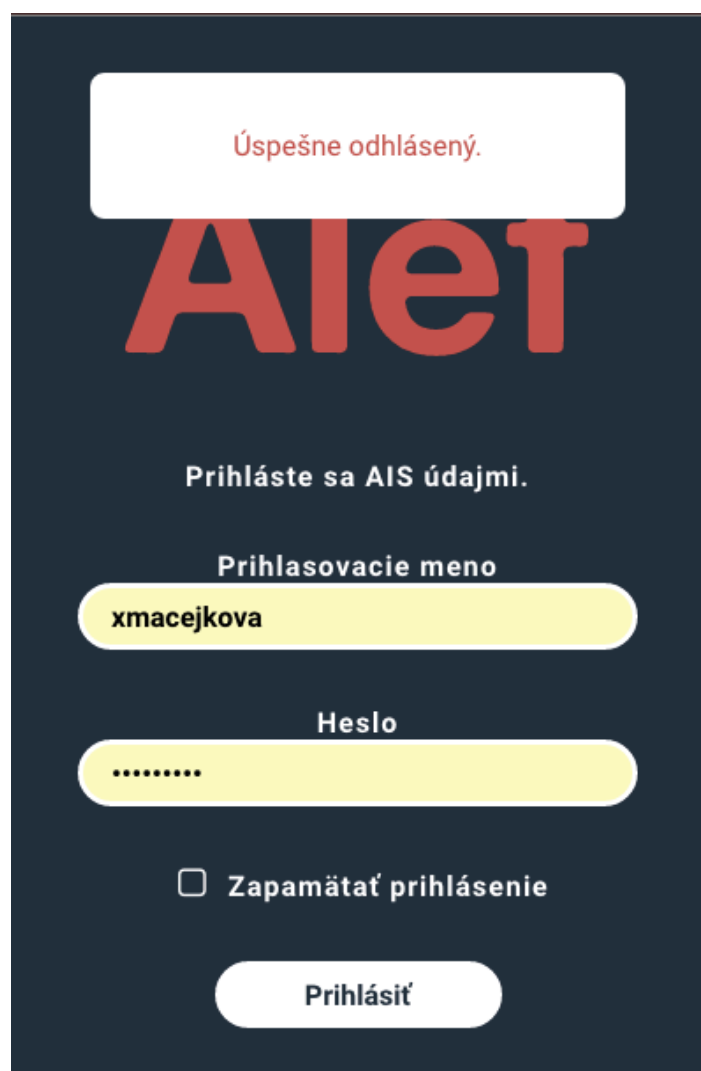
Pri zadaní nesprávneho prihlasovacieho mena alebo hesla používateľ nedostane prístup k aplikácii a je mu vypísaný nasledujúca chybová hláška (Obr. A).

Pri úspešnom odhlásení je používateľ presmerovaný na prihlasovaciu stránku aplikácie s informujúcou hláškou o úspešnom odhlásení (Obr. B).



The screenshot shows a dark blue login page for 'AIET'. At the top, a white rounded rectangle contains the error message 'Nesprávne prihlasovacie meno alebo heslo.' in red. Below this is the 'AIET' logo in red. The text 'Prihláste sa AIS údajmi.' is centered. There are two input fields: 'Prihlasovacie meno' with the text 'xmacejkova' and 'Heslo' with masked characters '.....'. Below the fields is a checkbox labeled 'Zapamätať prihlásenie' and a white 'Prihlásiť' button.

Obr. A: Zlé meno alebo heslo

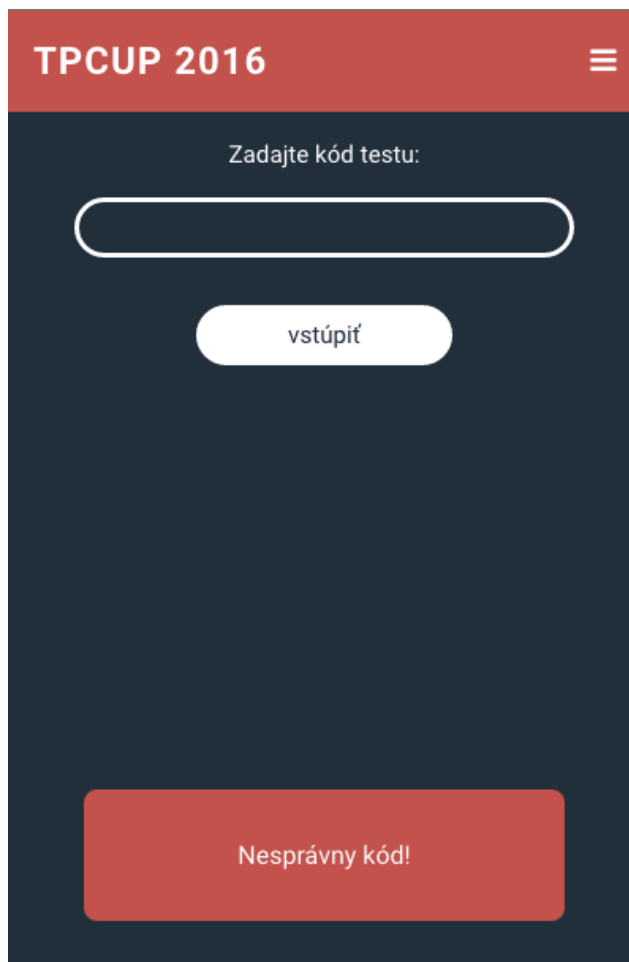


The screenshot shows the same dark blue login page for 'AIET'. At the top, a white rounded rectangle contains the success message 'Úspešne odhlásený.' in red. Below this is the 'AIET' logo in red. The text 'Prihláste sa AIS údajmi.' is centered. There are two input fields: 'Prihlasovacie meno' with the text 'xmacejkova' and 'Heslo' with masked characters '.....'. Below the fields is a checkbox labeled 'Zapamätať prihlásenie' and a white 'Prihlásiť' button.

Obr. B: Odhlásenie

Na Obr. C je zobrazený scenár, ktorý sa udeje pri nesprávnom zadaní kódu cvičenia.

Obr. D nastane, keď študent nemá prístupenie hodnotenia konkrétneho testu (na obrázku prvý v poradí) a chce sa tam dostať kliknutím na tento test.



The screenshot shows a dark blue mobile interface for 'TPCUP 2016'. At the top, there is a red header with the text 'TPCUP 2016' and a hamburger menu icon. Below the header, the text 'Zadajte kód testu:' is displayed above a white rounded rectangular input field. Underneath the input field is a white rounded rectangular button with the text 'vstúpiť'. At the bottom of the screen, a large red rounded rectangular box contains the text 'Nesprávny kód!'.

Obr. C: Zle zadaný kód testu



The screenshot shows a white mobile interface for 'TPCUP 2016'. At the top, there is a red header with the text 'TPCUP 2016' and a hamburger menu icon. Below the header, there is a table with two rows of test information, separated by horizontal lines.

Test Name	Period
1. TÝŽDEŇ - TEST(NEDOSTUPNÝ)	7.3. - 13.3.2016
1. TÝŽDEŇ - TEST	7.3. - 13.3.2016

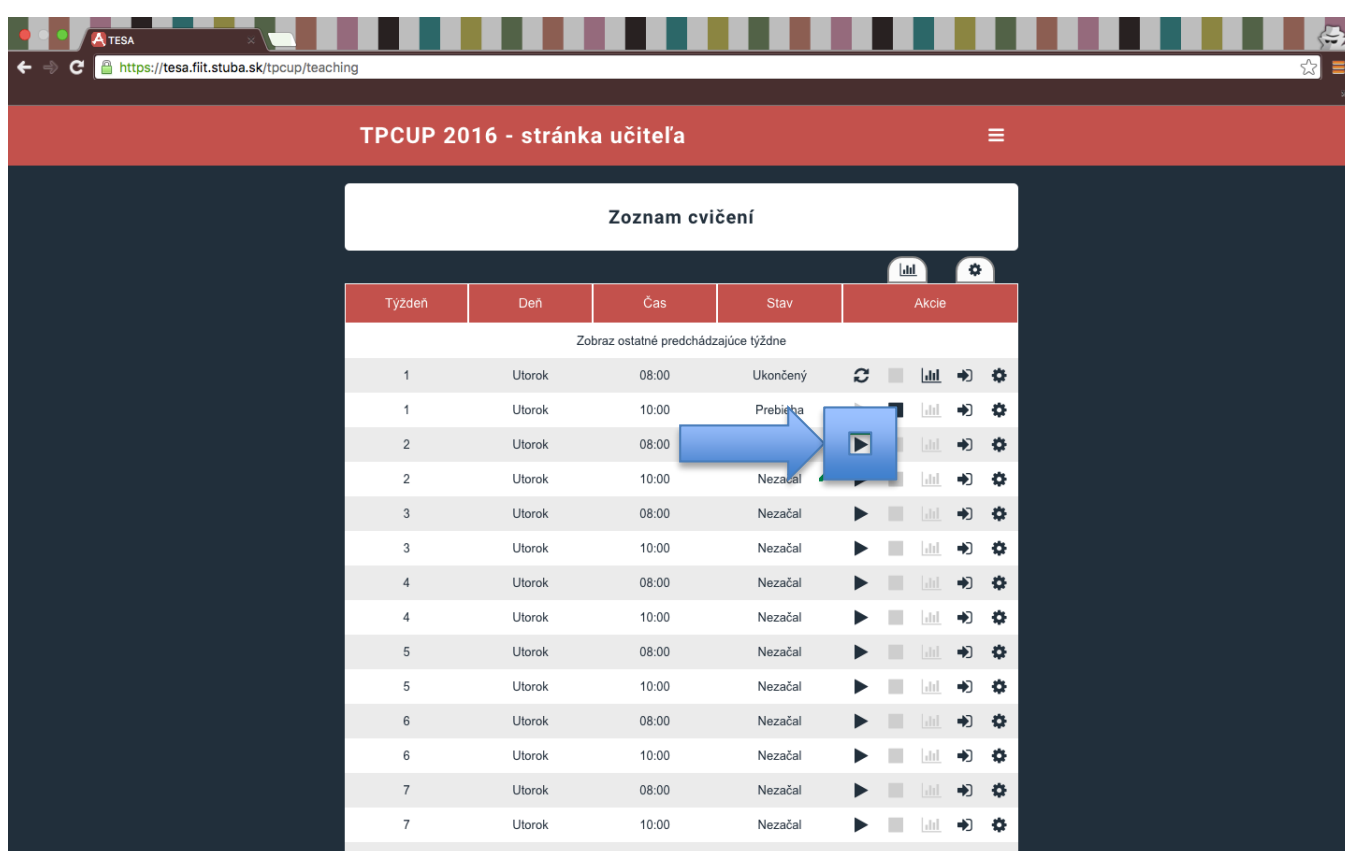
Odpovede ešte nie sú dostupné!

Obr. D: Nedostupné hodnotenie testu

Používateľská príručka pre učiteľa

Keďže sa počíta s tým, že učitelia používajú našu aplikáciu na počítačoch, je rozlíšenie ukážok zobrazené v danom rozlíšení. Učiteľ sa prihlási do systému rovnako, ako to bolo opísané v študentskej časti na Obr. 1 svojimi prihlasovacími údajmi z AIS. Po úspešnom prihlásení je učiteľ presmerovaný na stránku so zoznamom všetkých jeho dostupných cvičení/prednášok zobrazenej na Obr. 8. Sú v nej všetky potrebné informácie, ktoré učiteľ potrebuje vedieť. Pre lepšie pochopenie ikoniek akcií, sú dostupné taktiež tooltipy (stručná informácia o danej ikonke) pri ukázaní kurzora na konkrétnu ikonku (hover).

Spustenie testu a zobrazenie jeho výsledkov



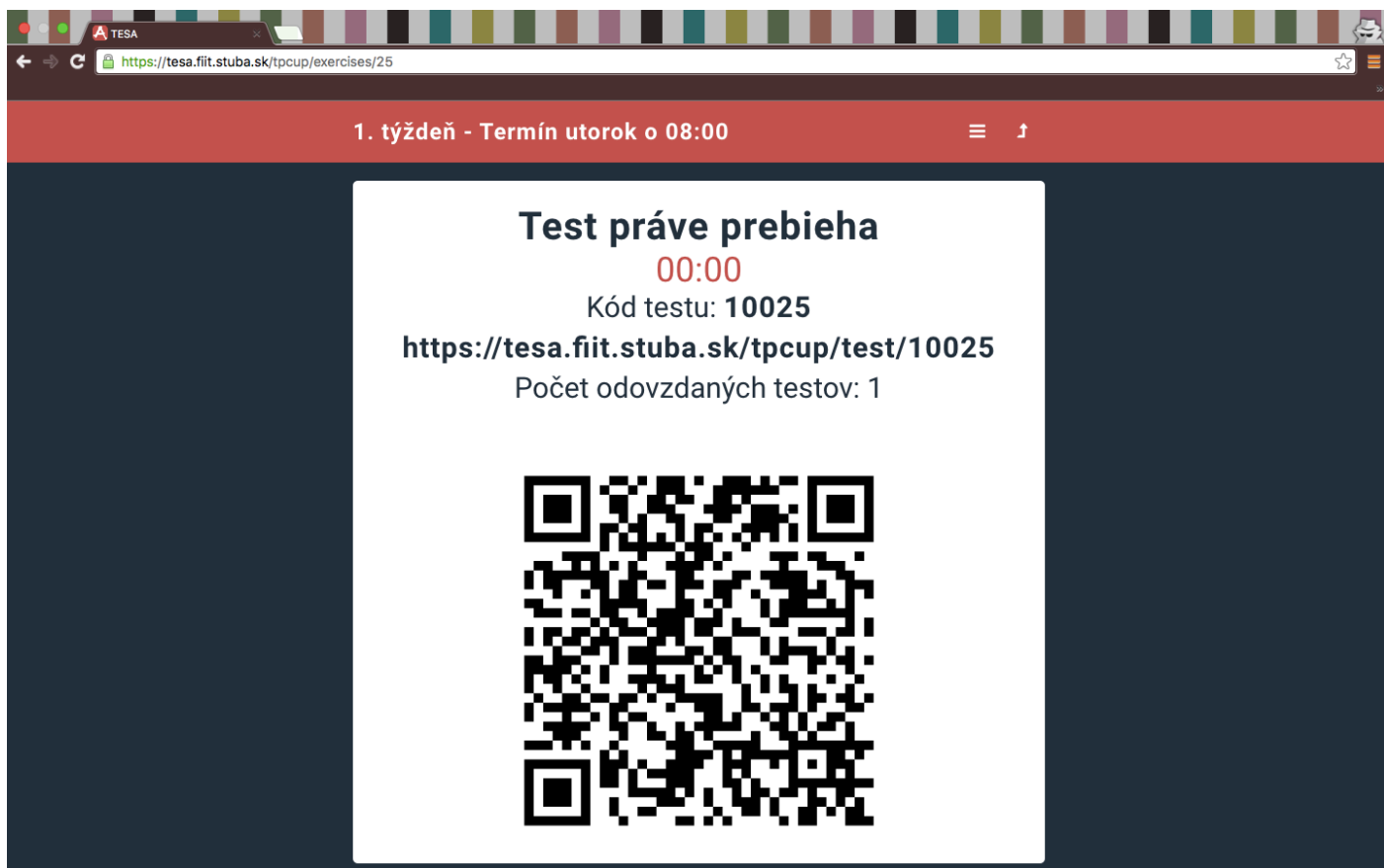
The screenshot shows a web browser window with the URL <https://tesa.fkit.stuba.sk/tpcup/teaching>. The page title is "TPCUP 2016 - stránka učiteľa". The main content is a table titled "Zoznam cvičení". The table has columns: "Týždeň", "Deň", "Čas", "Stav", and "Akcie". The table contains 14 rows of data. A blue arrow points to a play button icon in the "Akcie" column of the second row.

Týždeň	Deň	Čas	Stav	Akcie
Zobraz ostatné predchádzajúce týždne				
1	Utorok	08:00	Ukončený	↺ ⏪ ⏩ ↻ 📊 ⚙️
1	Utorok	10:00	Prebieha	⏪ ⏩ 📊 ⚙️
2	Utorok	08:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
2	Utorok	10:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
3	Utorok	08:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
3	Utorok	10:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
4	Utorok	08:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
4	Utorok	10:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
5	Utorok	08:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
5	Utorok	10:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
6	Utorok	08:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
6	Utorok	10:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
7	Utorok	08:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️
7	Utorok	10:00	Nezačal	▶️ ⏪ ⏩ 📊 ⚙️

Obr. 8: Zoznam cvičení

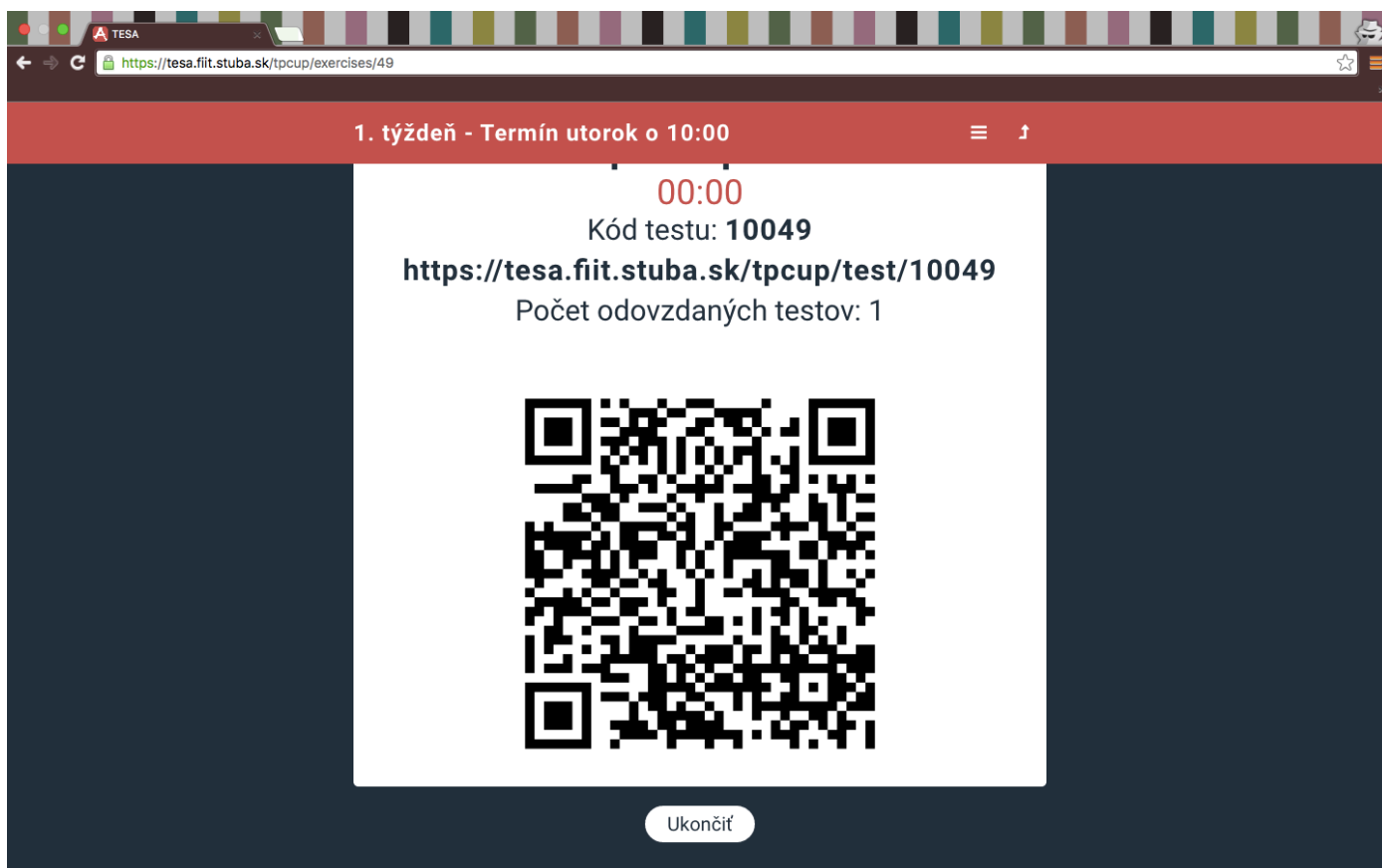
V tabuľke cvičení/prednášok si učiteľ vyberie ten termín, ktorý chce spustiť a klikne na ikonku šípky zobrazenú modrým štvorcem v Obr. 8.

Po spustení testu, sa učiteľovi zobrazí nasledujúca stránka (Obr. 9) zobrazujúca potrebné informácie o teste. Táto obrazovka je odporúčaná na zobrazenie premietačkou, aby tieto informácie boli viditeľné pre študentov.

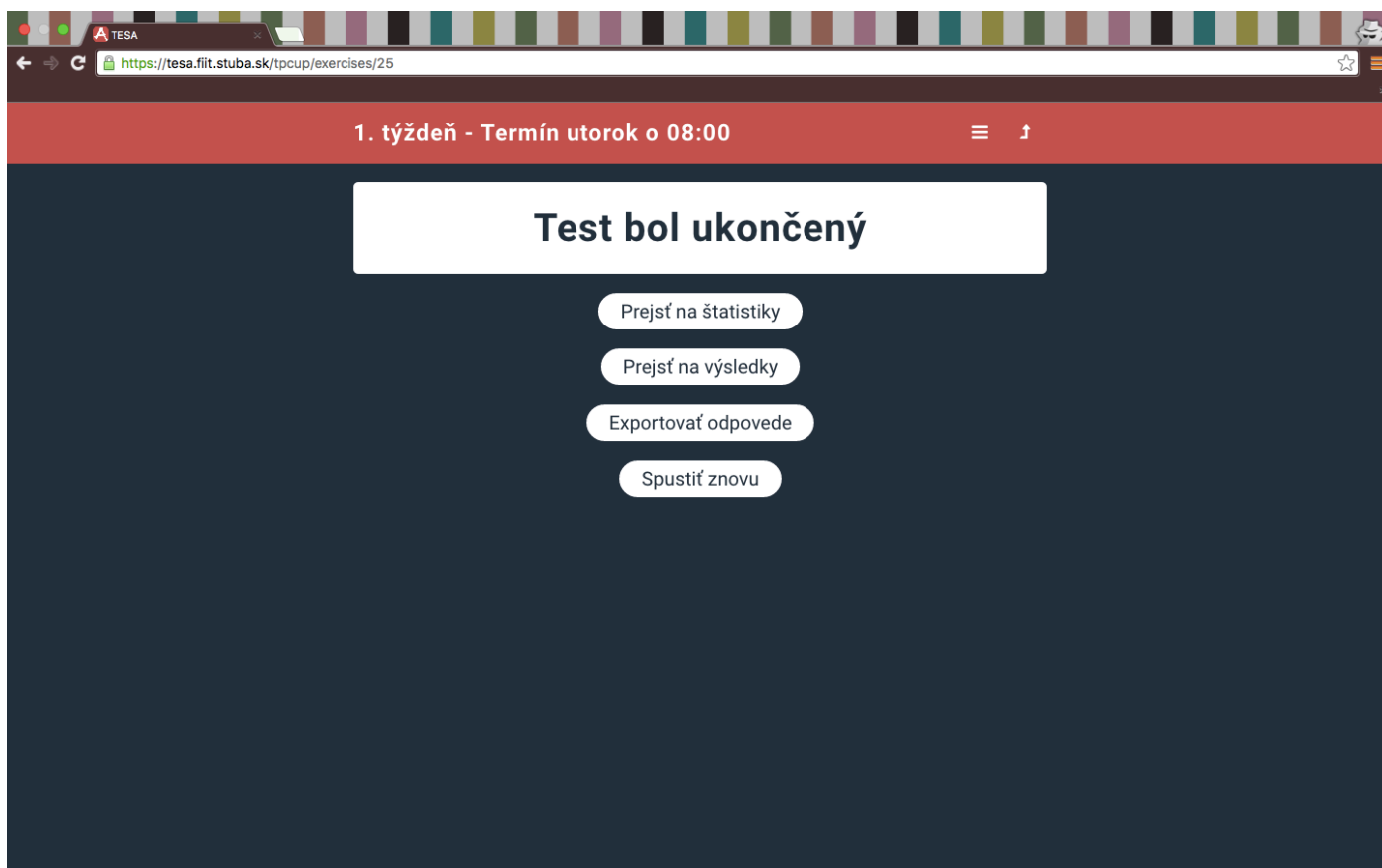


Obr. 9: Informácie o spustenom teste

Keď sa učiteľ rozhodne, že je čas test ukončiť, vykoná to kliknutím na tlačidlo „Ukončiť“ zobrazené na Obr. 10. Následne sa zobrazí stránka s informáciou o ukončení testu a možnosťami pokračovania (Obr. 11).

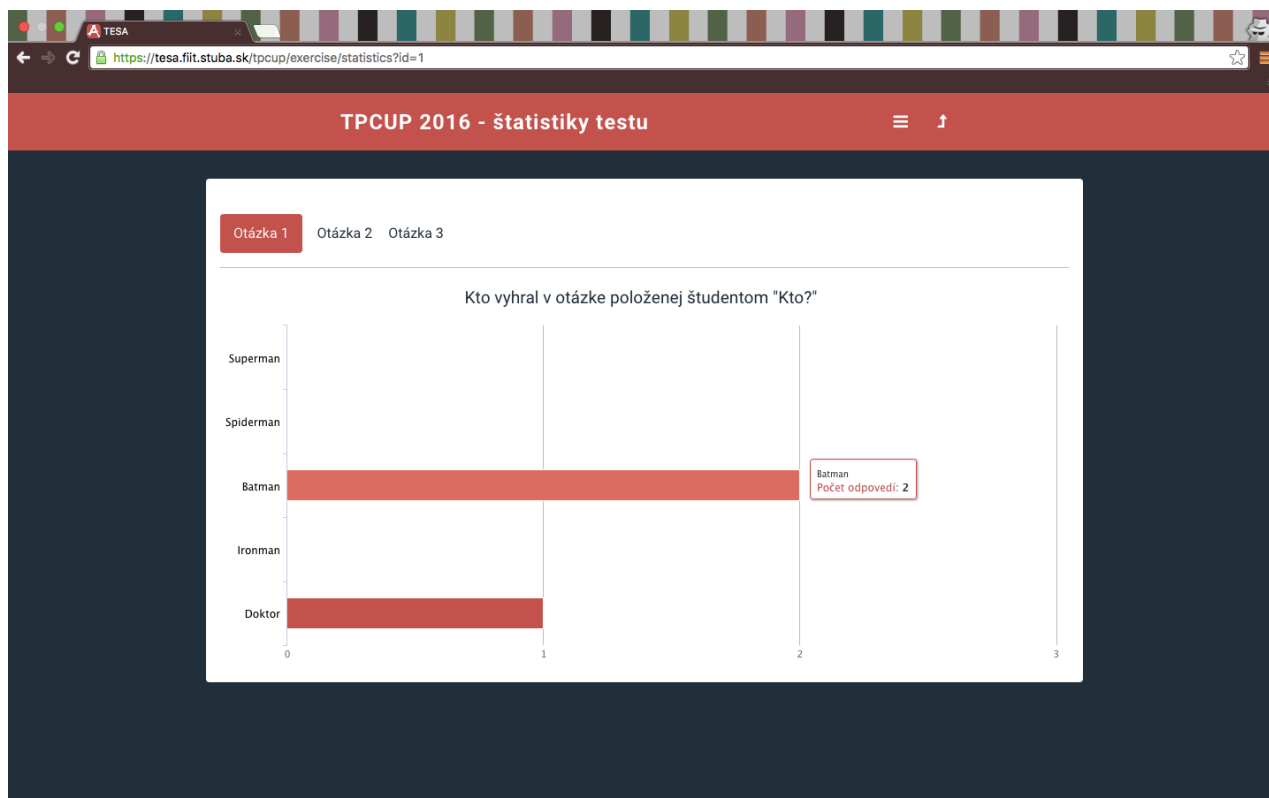


Obr. 10: Tlačidlo na ukončenie testu



Obr. 11: Ukončený test

Po ukončení testu zvyčajne nasleduje zhodnotenie testu a to zobrazením anonymnej štatistiky. K nej sa učiteľ dostane kliknutím na tlačidlo „Prejsť na štatistiky“ na Obr. 11. Následne sa zobrazí štatistika otázok zobrazená na Obr. 12. Taktiež si učiteľ môže zo zobrazených možností (Obr. 11) exportovať odpovede alebo prejsť na jednotlivé výsledky testov (Obr. 12).



Obr. 12: Štatistika

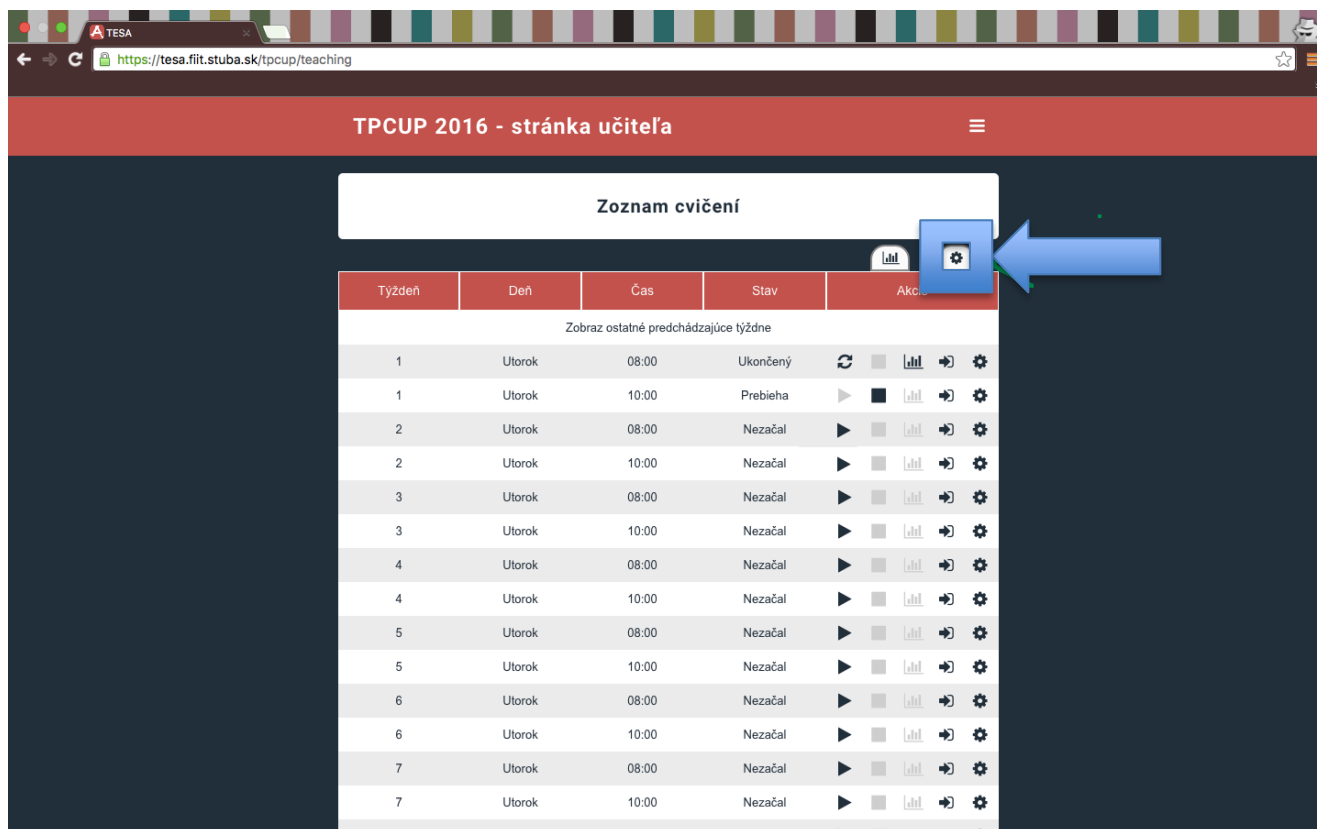
The screenshot shows a web browser window with the URL <https://tesa.fit.stuba.sk/tpcup/exercise/results?id=25>. The page title is "TPCUP 2016 - výsledky testu". The main content area displays a table with the following data:

Meno študenta	Ais id	Správnych odpovedí	Výsledky
Juraj Petrik	5731	1	Prezrieť test
Michal Farkas	52003	0	Prezrieť test
Silvia Macejkova	52626	1	Prezrieť test

Obr. 13: Jednotlivé výsledky testov

Hodnotenie testov

Testy podliehajú automatickému hodnoteniu. Avšak otvorené otázky musia byť skontrolované učiteľom. K tomuto hodnoteniu sa učiteľ dostane cez tabuľku cvičení/prednášok ikonkou zobrazenou na Obr. 14 modrým štvorcom.

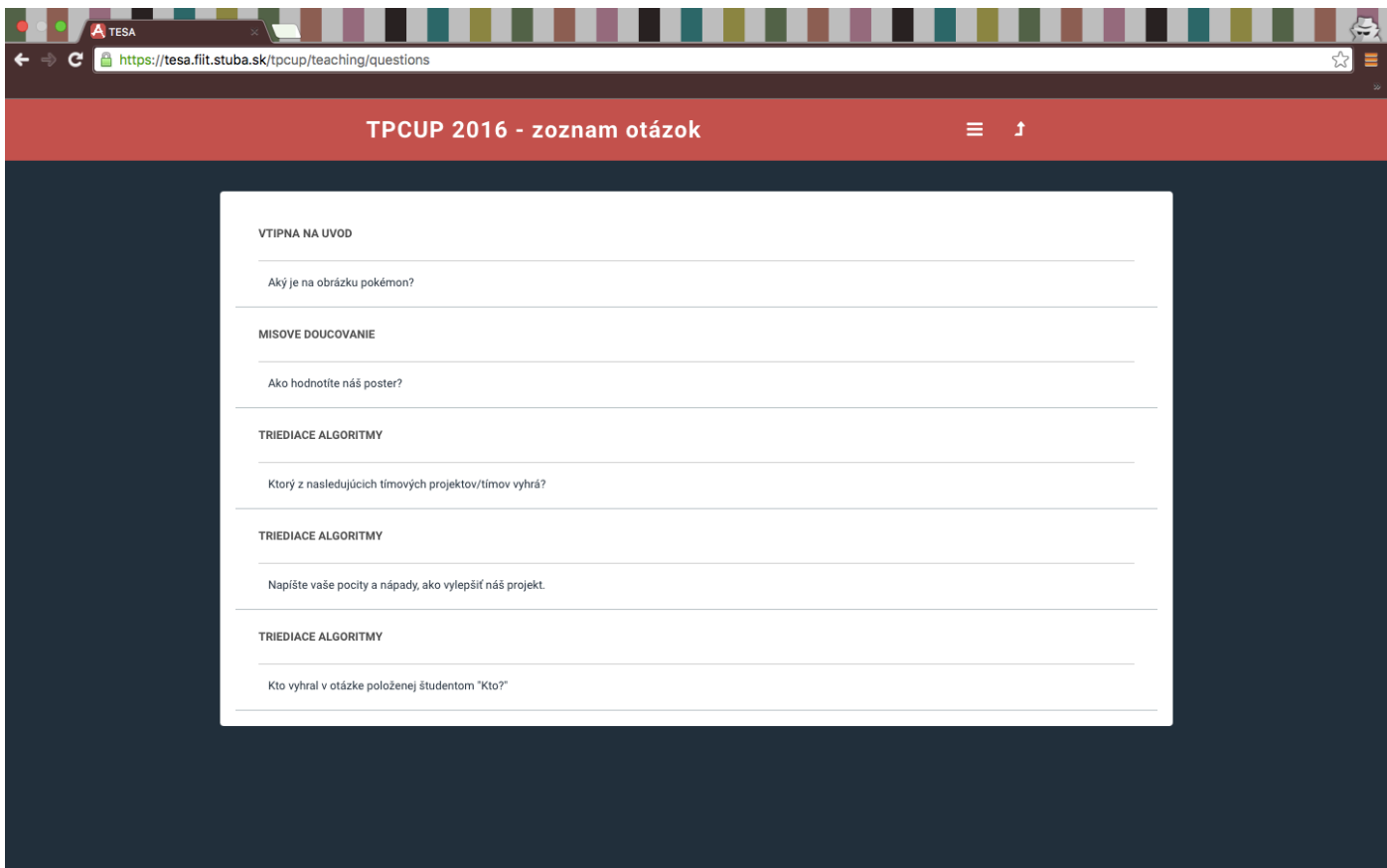


The screenshot shows a web browser window with the URL <https://tesa.fit.stuba.sk/tpcup/teaching>. The page title is "TPCUP 2016 - stránka učiteľa". The main content area is titled "Zoznam cvičení" and contains a table with columns: "Týždeň", "Deň", "Čas", "Stav", and "Akcie". A blue box highlights a gear icon in the "Akcie" column of the first row, with a blue arrow pointing to it from the right. Below the table, there is a link that says "Zobraz ostatné predchádzajúce týždne".

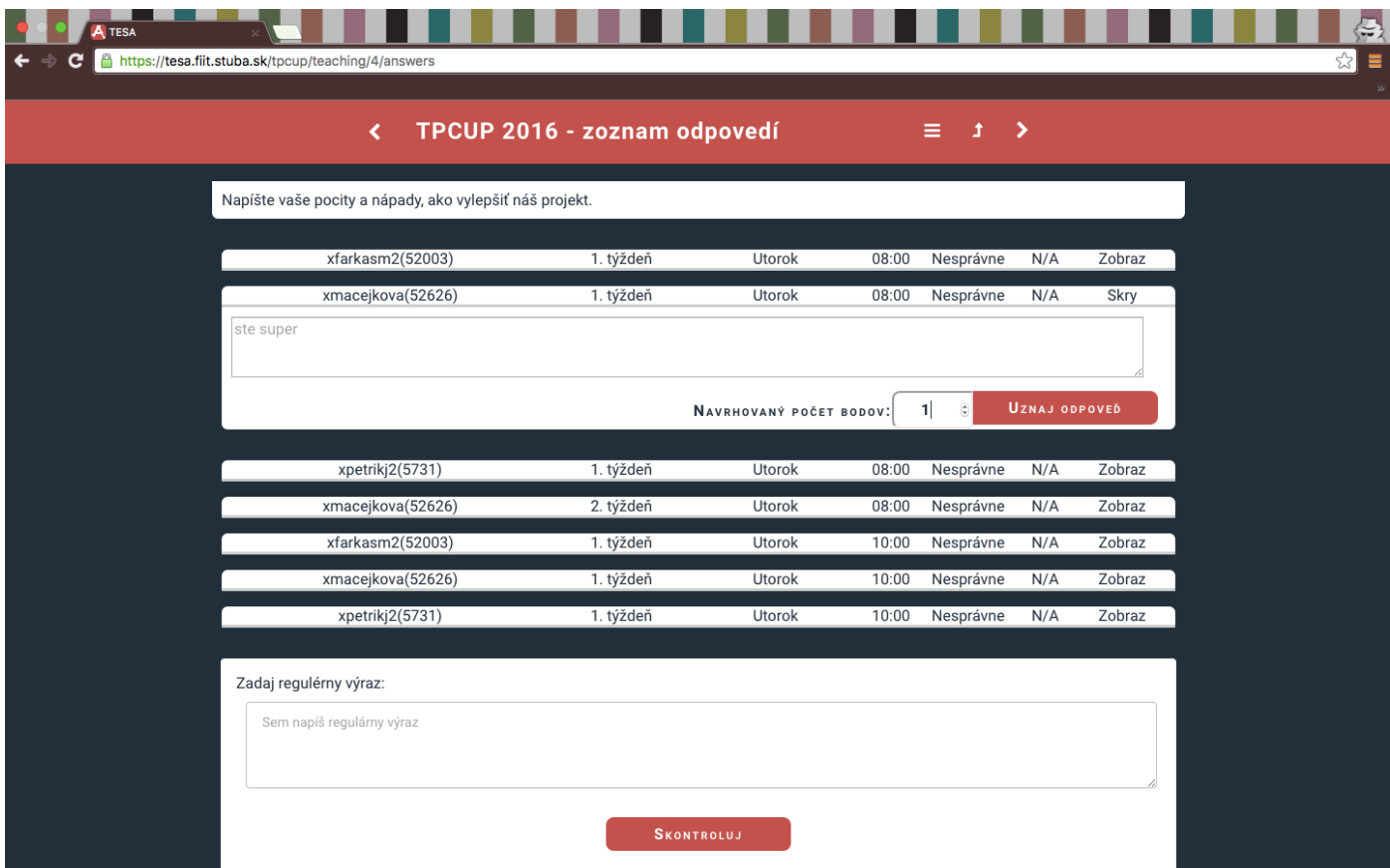
Týždeň	Deň	Čas	Stav	Akcie
Zobraz ostatné predchádzajúce týždne				
1	Utorok	08:00	Ukončený	🔄 📊 ➡ ⚙️
1	Utorok	10:00	Prebieha	▶ 📊 ➡ ⚙️
2	Utorok	08:00	Nezačal	▶ 📊 ➡ ⚙️
2	Utorok	10:00	Nezačal	▶ 📊 ➡ ⚙️
3	Utorok	08:00	Nezačal	▶ 📊 ➡ ⚙️
3	Utorok	10:00	Nezačal	▶ 📊 ➡ ⚙️
4	Utorok	08:00	Nezačal	▶ 📊 ➡ ⚙️
4	Utorok	10:00	Nezačal	▶ 📊 ➡ ⚙️
5	Utorok	08:00	Nezačal	▶ 📊 ➡ ⚙️
5	Utorok	10:00	Nezačal	▶ 📊 ➡ ⚙️
6	Utorok	08:00	Nezačal	▶ 📊 ➡ ⚙️
6	Utorok	10:00	Nezačal	▶ 📊 ➡ ⚙️
7	Utorok	08:00	Nezačal	▶ 📊 ➡ ⚙️
7	Utorok	10:00	Nezačal	▶ 📊 ➡ ⚙️

Obr. 14: Ikonka hodnotenia testov

Učiteľovi sa zobrazí stránka so všetkými otázkami (Obr. 15). Z nich si vyberie tú, ktorú chce hodnotiť. Po vybraní jednej konkrétnej otvorenej otázky sa zobrazí obrazovka na Obr. 16. Tam vie učiteľ ohodnotiť jednotlivé odpovede, ale aj napísať všeobecný regulárny výraz pre danú otázku na uľahčenie opravovania.



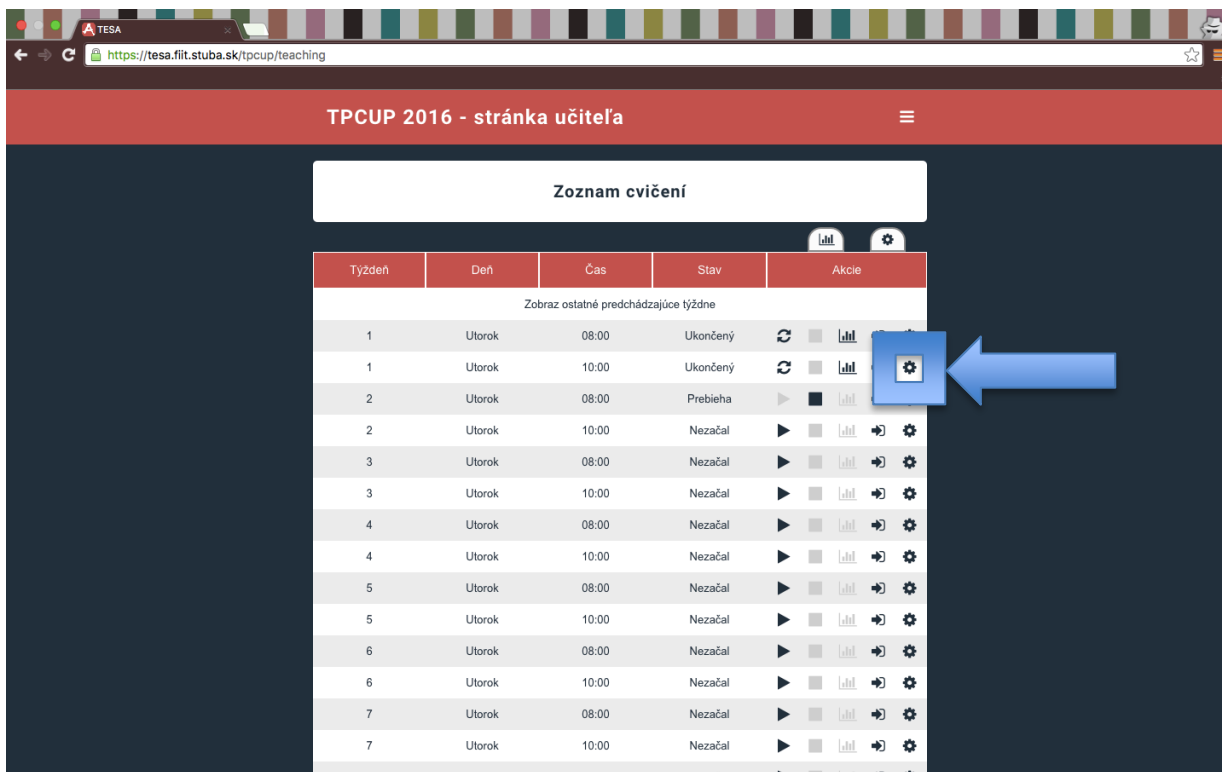
Obr. 15: Zoznam otázok na hodnotenie



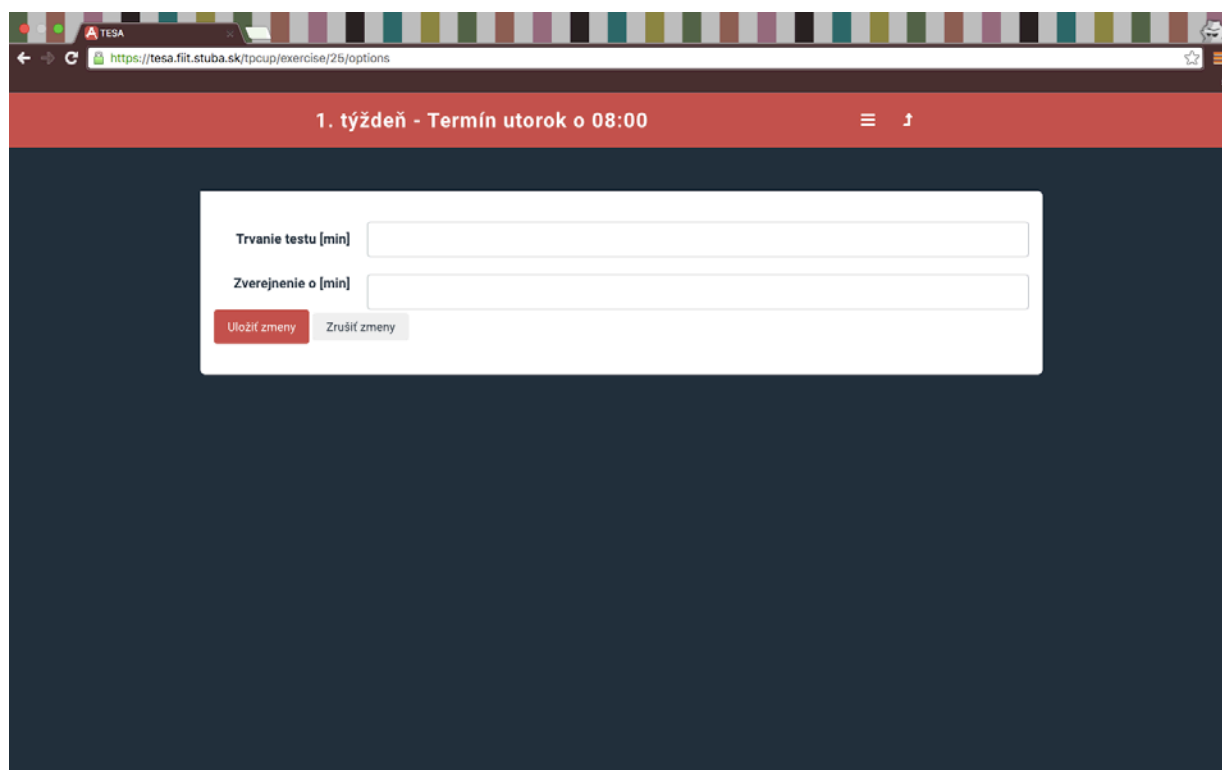
Obr. 16: Hodnotenie otvorenej otázky

Nastavenia zverejnenia a trvania testov

Testy môže učiteľ zverejniť buď ešte pred manuálnym ohodnotením otvorených otázok alebo až po ňom. Taktiež má možnosť nastaviť stanovenú dĺžku testu (Obr. 18). To všetko je možné v nastaveniach, ku ktorým sa dostane pomocou ikonky znázornenej v Obr. 17 modrým štvorcom.



Obr. 17: Ikonka nastavenia zverejnenia a trvania testov



Obr. 18: Obrazovka nastavenia zverejnenia testov

Vysvetlivky pre učiteľa

Jan Mazak
Tabuľky cvičení/prednášok

Učenie → Spätná väzba → Spätná väzba pre vývojárov
Odhlásiť → Odhlásenie používateľa

TPCUP 2016 - stránka učiteľa Menu

Zoznam cvičení

Celková štatistika
Nastavenia hodnotenia otázok

Týždeň	Deň	Čas	Stav	Akcie
Zobraz ostatné predchádzajúce týždne				
1	Utorok	08:00	Ukončený	🔄 📊 ➡ ⚙️ → Štatistika testu
1	Utorok	10:00	Ukončený	🔄 📊 ➡ ⚙️ → Znovu-spustenie testu
2	Utorok	08:00	Prebieha	▶️ 📊 ➡ ⚙️ → Zastavenie testu
2	Utorok	10:00	Nezačal	▶️ 📊 ➡ ⚙️ → Podrobnosti testu
3	Utorok	08:00	Nezačal	▶️ 📊 ➡ ⚙️ → Nastavenia testu
3	Utorok	10:00	Nezačal	▶️ 📊 ➡ ⚙️ → Spustenie testu
4	Utorok	08:00	Nezačal	▶️ 📊 ➡ ⚙️
4	Utorok	10:00	Nezačal	▶️ 📊 ➡ ⚙️

Štatistika testu
Znovu-spustenie testu
Zastavenie testu
Podrobnosti testu
Nastavenia testu
Spustenie testu

Príručka k importom dát

Táto príručka zachytáva kroky potrebné k importu dát do systému Tesa Aleftng. Zároveň obsahuje popis štruktúry potrebných csv súborov. Import dát je možné využiť na vloženie používateľov, termínov, otázok a iných. Pomocným zdrojom pre vytváranie termínov je skript na generovanie rozvrhu na celý semester, ktorého opis je rovnako súčasťou tejto príručky.

1 Import termínov

1.1 Formát súboru CSV

Pre importovanie termínov je potrebné použiť csv súbor v nasledujúcom formáte:

Exercise start, Exercise end, Code, Lecturer, Week, Concepts

Exercise start – definuje čas a dátum štandardného začiatku cvičenia. Využíva sa štandardne formát YYYY-MM-DD HH:mm:ss (+timezone data).

Exercise end – definuje čas a dátum štandardného konca cvičenia. Využíva sa štandardne formát YYYY-MM-DD HH:mm:ss (+timezone data).

Code – definuje unikátny kľúč pre vstup do testu. Pre každé cvičenie musí byť rozdielny. Použiť je možné celé číslo veľkosti 4 bytov. Optimálne je využívať kladné čísla.

Lecturer – definuje používateľa, ktorý vedie daný termín. V tomto stĺpci je nutné uviesť identifikátor (ID stĺpec) zo záznamu konkrétneho používateľa, ktorému sa priraduje termín.

Week – definuje týždeň, počas ktorého je štandardne plánovaný daný termín. Obsahuje id týždňa, na ktorý sa odkazuje.

Concepts – definuje koncepty, ktoré využíva daný termín. Jednotlivé koncepty sú uvádzané ich plným názvom a sú oddelené bodkočiarkou.

Súčasťou súboru CSV je aj hlavička.

1.2 Spustenie rake tasku

Import termínov je spúšťaný štandardne z príkazového riadku spustením:

```
rake tesa:data:import_exercises[filename.csv]
```

kde sa namiesto filename.csv uvedie konkrétny názov importovaného súboru csv (pre súbory mimo root adresáru aj cesta k nemu).

1.3 Implementácia

Implementácia sa nachádza v: /lib/tasks/csvsimple.rake

2 Import používateľov

2.1 Formát súboru CSV

Pre importovanie termínov je potrebné použiť csv súbor v nasledujúcom formáte:

Login, Role, First name, Last name, Password, Type

Login – definuje prihlasovacie meno používateľa. Prihlasovacie meno musí byť unikátne pre každého používateľa.

Role – definuje rolu používateľa. Možné je využiť jednu z troch rolí: *student* – pre študenta, *teacher* – pre učiteľa a *administrator* – pre administrátora (vedúceho predmetu)

First name – definuje krstné meno používateľa.

Last name – definuje priezvisko používateľa

Password – definuje prihlasovacie heslo pre vytvárané používateľské konto. Potrebné je len pri type používateľa *LocalUser* (viď nižšie).

Type – definuje typ používateľa. Možné je využiť dva typy: *LocalUser* – lokálny používateľ, ktorého prihlasovanie je zabezpečené len využitím dát z databázy a *LdapUser* – používateľ z AIS, ktorého prihlasovanie zabezpečuje univerzitný LDAP. Pre typ *LdapUser*, z dôvodu využitia externej autorizácie, nie je potrebné definovať heslo (nevyužije sa).

Súčasťou súboru CSV je aj hlavička.

2.2 Spustenie rake tasku

Import používateľov je spúšťaný štandardne z príkazového riadku spustením:

```
rake tesa:data:import_users[filename.csv]
```

kde sa namiesto filename.csv uvedie konkrétny názov importovaného súboru csv (pre súbory mimo root adresáru aj cesta k nemu).

2.3 Implementácia

Implementácia sa nachádza v: `/lib/tasks/csvsimple.rake`

3 Import týždňových konceptov

3.1 Formát súboru CSV

Pre importovanie týždňových konceptov je potrebné použiť csv súbor v nasledujúcom formáte:

Week, Concept, week_id_or_number

Week – definuje týždeň, ktorému sú pridelované koncepty. Týždeň sa určuje jedným číslom, ktoré určuje číslo týždňa v poradí počas semestra, alebo id záznamu týždňa (v závislosti od obsahu stĺpca *week_id_or_number* – viď nižšie).

Concept – definuje koncepty, ktoré sú pridelené konkrétnemu týždňu určenému v stĺpci Week. Obsahuje jeden alebo viac konceptov. Pri použití viacerých je potrebné, aby boli navzájom oddelené bodkočiarkou.

week_id_or_number – určuje podľa akého stĺpca tabuľka týždňov (weeks) sa budú koncepty priradovať. Týždne obsahujú vlastný identifikátor (ID) a číslo týždňa určujúce poradové číslo počas semestra. V importeri je možné využiť oba tieto identifikátory. Pre použitie ID je potrebné využiť hodnotu *id* a pre číslo týždňa využiť hodnotu *number*

Súčasťou súboru CSV je aj hlavička.

3.2 Spustenie rake tasku

Import týždňových konceptov je spúšťaný štandardne z príkazového riadku spustením:

```
rake tesa:data:import_weekly_concepts[filename.csv]
```

kde sa namiesto filename.csv uvedie konkrétny názov importovaného súboru csv (pre súbory mimo root adresáru aj cesta k nemu).

3.3 Implementácia

Implementácia sa nachádza v: /lib/tasks/csvsimple.rake

4 Import testových otázok

4.1 Formát súboru CSV

Pre importovanie testových otázok je potrebné použiť csv súbor v nasledujúcom formáte:

Title,Question,Local Concepts,Global Concepts,Type,Answers,Difficulty,Image,Is special,Week

Title – definuje univerzálny identifikátor otázky. Pre každú otázku by mal byť odlišný (nutná podmienka to nie je, vhodné je to pre prehľadnosť). Určuje názov (popis) otázky, ktorý je využívaný v zobrazení otázky.

Question – definuje text zadania otázky. Obsahovať môže voľný reťazec znakov, prípadne súčasťou môže byť aj vzorec v latex formáte.

Local Concepts – definuje koncepty, do ktorých patrí otázka. Malo by ísť o koncepty, ktoré sú pridelené konkrétnym termínom. Definuje sa vymenovaním názvov konceptov, ktoré sú oddelené bodkočiarkou.

Global Concepts – definuje koncepty, do ktorých patrí otázka v rámci využívania týždňových konceptov. Definuje sa vymenovaním názvov konceptov, ktoré sú oddelené bodkočiarkou.

Type – definuje typ otázky. Môže to byť: *single-choice* (otázka s jednou správnou odpoveďou), *multi-choice* (otázka s viacerými správnymi odpoveďami), *answer-validator* (otázka na ohodnotenie správnosti), *open-question* (otázka s voľnou odpoveďou)

Answers – vymenováva odpovede na otázku. Odpovede sú navzájom oddelené bodkočiarkou. Pre *open-question* nie je potrebné definovať žiadne, pre *answer-validator* sa definuje jedna. Pre otázky *single-choice* a *multi-choice* typu je možné definovať ľubovoľný vhodný počet odpovedí.

V závislosti na type, je potrebné určiť správnosť odpovedí ohraňením pomocou XML tagov: `<correct>odpoveď</correct>`. Pre *single-choice* otázku musí byť takto vyznačená len jedna odpoveď. Pre *multi-choice* otázku ich môže byť takto vyznačených ľubovoľný počet.

Difficulty – definuje náročnosť otázky. Tá môže byť: *trivialne, lahke, stredne, tazke, impossible*.

Image – definuje názov súboru obrázka, ktorý je použitý pre doplnenie otázky. Je potrebné ho uvádzať v tvare *imagename.imageformat*. Pri importe sa musí nachádzať vo zvolenom adresári obrázkov.

Is special – definuje či je otázka špeciálna, t. j. že ju pri výbere otázok odporúčaním systém vygeneruje každému študentovi termínu, ktorému prislúcha otázka. Určuje sa ako pravdivostná hodnota. Pre zabezpečenie toho, že otázka bude vybratá všetkým študentom, je potrebné uviesť *TRUE* (1).

Week – definuje týždeň otázky. Týždeň je použitý pre mapovanie *Global concepts* na konkrétne týždne. Obsahuje číslo daného týždňa (*week_number* v databáze).

Súčasťou súboru CSV je aj hlavička.

4.2 Spustenie rake tasku

Import testových otázok je spúšťaný štandardne z príkazového riadku spustením:

```
rake tesa:data:import_tests[filename.csv, imagedir]
```

kde sa namiesto *filename.csv* uvedie konkrétny názov importovaného súboru csv (pre súbory mimo *root* adresáru aj cesta k nemu) a namiesto *imagedir* sa uvedie cesta k adresáru obsahujúcemu obrázky pre otázky. Ak žiadne otázky neobsahujú obrázky je potrebné určiť ktorýkoľvek adresár.

4.3 Implementácia

Implementácia sa nachádza v: `/lib/tasks/testdata.rake`

5 Generátor termínov

5.1 Formát súboru CSV

Vstupný csv súbor musí byť v nasledujúcom formáte:

Exercise start, Exercise end, Lecturer

Exercise start – definuje čas a dátum štandardného začiatku cvičenia. Využíva sa štandardne formát `YYYY-MM-DD HH:mm:ss (+timezone data)`.

Exercise end – definuje čas a dátum štandardného konca cvičenia. Využíva sa štandardne formát `YYYY-MM-DD HH:mm:ss (+timezone data)`.

Lecturer – definuje používateľa, ktorý vedie daný termín. V tomto stĺpci je nutné uviesť identifikátor (ID stĺpec) zo záznamu konkrétneho používateľa, ktorému sa priraduje termín.

Súčasťou súboru CSV je aj hlavička. Do súboru je potrebné vložiť rozpis termínov cvičení a prednášok pre predmet na prvý týždeň semestra.

5.2 Spustenie skriptu generovania

Skript generovania termínov je spúšťaný štandardne z príkazového riadku spustením:

```
ruby exercisecsv2csv.rb filename.csv term_length
```

kde sa namiesto filename.csv uvedie konkrétny názov importovaného súboru csv (pre súbory mimo root adresáru aj cesta k nemu) a namiesto term_length je definovaný počet týždňov semestra.

Skript vygeneruje v adresári aplikácie súbor exercise.csv obsahujúci rozvrh na celý semester.

5.3 Implementácia

Implementácia sa nachádza v: exercisecsv2csv.rb

1. Inštalačná príručka - TESA

verzia: 1.0 zostavené: 13.5.2016 vypracoval: Lukáš Csóka, xcsokal@stuba.sk

2. Úvod

Táto príručka udáva podmienky a spôsob inštalácie systému TESA - Priebežné overovanie prípravy študentov na cvičeniach. Tento systém je založený na open-source riešení Alef-tng (<https://github.com/PewePro/alef-tng>). Kompletné zdrojové kódy sú dostupné tu: https://github.com/mrtvolka/tesa_alef-tng.

3. Podmienky inštalácie

TESA nie je náročná na výpočtový výkon, vyžaduje ale splnenie viacerých podmienok na funkčný beh:

- Minimálne virtuálny server (2GB operačnej pamäte, 10GB diskového miesta, 2 virtualizované jadrá)
- Server má operačný systém Ubuntu Server 14.04.4 LTS alebo obdobný operačný systém
- Na serveri je nainštalovaný PostgreSQL vo verzii 9.3.11 (presne táto verzia je nutná)
- Na serveri je nainštalovaný Apache2 (otestované s verziou 2.4.7) (ak sa použije iný webservice, treba upraviť konfiguráciu pre nainštalovaný webservice)
- TCP port 80 a 443 sú voľné a prístupné (respektíve využívané v Apache2)
- Na inštaláciu je nutné mať užívateľa s právami root

4. Postup inštalácie

Jeden server je možné používať na viacero predmetov, pričom ale na každý predmet je vždy nutná vlastná inštancia – vlastný zdrojový kód, vlastná databáza. V tomto návode budeme predpokladať inštaláciu WebTestu tak, aby súčasne bežalo viacero inštancií, pričom každá inštancia beží na vlastnej sub-uri adrese k root doméne, napríklad pre predmet AZA beží inštancia na doméne `tesa.fiit.stuba.sk/aza`.

Každá inštancia by mala mať vlastného vytvoreného používateľa. Pre ukážky inštalácie predpokladáme používateľa `user_aza` s domovským priečinkom `/home/user_aza`.

Najprv je udávaný postup inštalácie len pre TESA, následne aj voliteľná integrácia s webservice Apache2, vrátane inštalácie passengera. Následne príprava databázy, príklady konfiguračných súborov a časté chyby.

4.1. Inštalácia inštancie TESA

1. Nainštalovať RUBY VERSION MANAGER (RVM) - <https://rvm.io>. V čase tvorby tejto príručky (13.5.2016) sa RVM inštaloval v domovskom

adresári užívateľa pomocou príkazov:

```
gpg --keyserver hkp://keys.gnupg.net --recv-keys \  
409B6B1796C275462A1703113804BB82D39DC0E3 \  
\curl -sSL https://get.rvm.io | bash -s stable
```

2. Zmeniť RUBY verziu na 2.2.0 pomocou príkazov:

```
rvm install 2.2.0  
rvm --default 2.2.0
```

3. Do zvoleného priečinka, napríklad v našom prípade `/home/user_aza/aza` nahrať zdrojový kód aplikácie. Napríklad v domovskom adresári pomocou príkazu:

```
git clone -b osicky_develop https://git@github.com/mrtvolka/tesa_alef-  
tng.git aza
```

4. Vytvoríť databázového používateľa, ktorý má právo vytvárať databázy (privilégium `CREATEDB`).

5. V zdrojovom kóde skopírovať a upraviť súbor `/home/user_aza/aza/config/database.yml.example`. Je nutné upraviť minimálne názov produkčnej databázy, aby mala každá inštancia jedinečný názov databázy a prihlasovacie údaje. Príklad úpravy je v časti ukážok konfiguračných súborov.

```
cp /home/user_aza/aza/database.yml.example /home/user_aza/aza/database.yml  
nano database.yml
```

6. Do `/home/user_aza/aza/config/application.rb` pridať riadok:

```
config.relative_url_root = "/XXXX"
```

,kde `XXXX` značí sub-uri. V našom prípade to bude vyzerať takto:

```
config.relative_url_root = "/aza"
```

7. V `/home/user_aza/aza/config/environments/production` upraviť smtp spojenie v časti súboru označenom ako `#Mail`. Príklad úpravy je v časti konfiguračných súborov.
8. V adresári zdrojového kódu `/home/user_aza/aza` spustiť príkaz `gem install bundler` na nainštalovanie bundler - rails utility na manažovanie rails gemov.
9. V adresári zdrojového kódu `/home/user_aza/aza` spustiť príkaz `bundle install` na nainštalovanie všetkých potrebných rails gemov (gemy sú definované v `/home/user_aza-aza/Gemfile`).
10. Spustiť príkaz `rake assets:precompile` v adresáry zdrojového kódu - v našom prípade `/home/user_aza/aza`.
11. (Voliteľné) TESA podporuje LDAP na prihlasovanie užívateľov, pričom im pridá aj práva podľa nastavenia v LDAP, respektíve AIS. Je nutné upraviť súbor `/home/user_aza/aza/config/ldap.yml`. Príklad úpravy je v časti ukážok konfiguračných súborov.

4.2. Integrácia s Apache2

Pre integráciu s Apache2 sa predpokladá prístup len cez https. Tiež sa predpokladá DocumentRoot nastavený na /www/data/.

1. Doinštalovať knižnice, ak chýbajú:

```
apt-get install libcurl4-openssl-dev
apt-get install apache2-threaded-dev
```

2. Nainštalovať Passenger.

```
gem install passenger
```

a zvoliť Ruby a následne postupovať podľa inštrukcií.

3. Do /etc/apache2/mods-available/passenger.conf a /etc/apache2/mods-available/passenger.load.conf doplniť:

```
LoadModule passenger_module <<CESTA K RUBY>>@aleftng/gems/passenger-5.0.21/buildout/apache2/mod_passenger.c
<IfModule mod_passenger.c>
PassengerRoot <<CESTA K RUBY>>@aleftng/gems/passenger-5.0.21
PassengerDefaultRuby <<CESTA K RUBY>>@aleftng/wrappers/ruby
</IfModule>
```

Výraz <<CESTA K RUBY>> nahradiť cestou k ruby, cesta môže vyzerat takto /home/user_aza/.rvm/gems/ruby-2.2.0. Ako je vidno z časti passenger-5.0.21, Passenger bol nainštalovaný vo verzii 5.0.21, ktorý bol v čase našej inštalácie posledná stabilná verzia. Môže byť nutné zmeniť verziu. Verziu Passengera je možné zistiť príkazom passenger -v.

4. Do /etc/apache2/sites-enabled/000-default.conf doplniť:

```
RewriteCond %{HTTPS} !=on
RewriteRule ^/aza(.*) https://webtest.sk/aza$1 [R,L]
Alias /aza /var/www/aza/public
```

a uistiť sa, že RewriteEngine je nastavený na On.

5. Do /etc/apache2/sites-enabled/default-ssl.conf doplniť:

```
Alias /aza /var/www/aza/public
<Location /aza>
    PassengerBaseURI /aza
    PassengerAppRoot /var/www/aza
    PassengerResolveSymlinksInDocumentRoot on
    SetEnv ALEFTNG_SECRET_KEY_BASE <<SECRET>>
</Location>

<Directory /var/www/aza/public>
```

```
    AllowOverride all
    Options -MultiViews
</Directory>
```

Výraz <<SECRET>> nahraď klúčom - tajomstvom, napríklad vygenerovaným pomocou príkazu `rake secret` spustenom v adresári zdrojového kódu.

6. V priečinku nastavenom ako `DocumentRoot` vytvorí symbolickú linku na priečinok so zdrojovým kódom inštalácie. Napríklad takto: `ln -s /home/user_aza/aza aza`.
7. Reštartovať webserver `/etc/init.d/apache2 restart`.

5. Upgrade inštalácie

Pokiaľ je `WebTest` nainštalovaný a chcete ho povýšiť, postupujte nasledovne:

1. Vypnúť webserver `/etc/init.d/apache2 stop`.
2. Updatnúť zdrojové súbory, napríklad zo vzdialeného úložiska pomocou príkazov spustených v adresári so zdrojovými kódmi:

```
git fetch origin
git reset --hard origin/osicky_develop
```
3. Do `/config/application.rb` treba doplniť: `config.relative_url_root = "/XXXX"`, kde `XXXX` značí sub-uri.
4. Zmigrovať databázu na novú verziu príkazom `rake db:migrate RAILS_ENV=production`.
5. Spustiť: `rake assets:precompile`.
6. Zapnúť webserver `/etc/init.d/apache2 start`.

6. Nastavenie databázy

Ak je správne nakonfigurované spojenie na databázu súborom `home/user_aza/aza/config/database.yml`, je možné manažovať databázu pre inštalácie pomocou rails príkazov, tzn. `rake`.

Pre konkrétnu inštaláciu sa spúšťajú príkazy v adresári zdrojového kódu inštalácie. Rails pozná štandardne tri prostredia: `development`, `production` a `test`. Na koniec každého príkazu je možné doplniť, ktorého prostredia sa daný príkaz týka, pomocou `RAILS_ENV=production` sa napríklad príkaz viaže na produkčné prostredie. Ak nie je doplnené `RAILS_ENV`, tak príkaz sa viaže na vývojové prostredie.

6.1. Zrušenie databázy

Príkaz `rake db:drop` zruší databázu.

6.2. Vytvorenie databázy

Príkaz `rake db:create` vytvorí databázu. Obvykle sa používa dokopy s migračným príkazom.

6.3. Migrácia databázy

Príkaz `rake db:migrate` zmigruje databázu. Migrácia databázy znamená, že budú do databázy nahrané všetky chýbajúce tabuľky a prepojenia, ktoré napríklad mohli byť pridané v novšej verzii. Tiež sa používa po vytvorení databázy na načítanie databázovej schémy ako inicializačný skript.

6.4. Vytvorenie a inicializácia databázy pre inštanciu

Pre produkčné prostredie treba najprv vytvoriť a zmigrovať databázu.

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

Následne treba inicializovať niektoré tabuľky.

6.4.1. Tabuľka `courses`

Táto tabuľka obsahuje údaje o inštancii. Aktuálne do tabuľky `courses` je nutné zadať názov inštalácie a časy vytvorenia a updatovania záznamu inštalácie. Je možné túto tabuľku naplniť príkazom `rake tesa:data:aza_setup` pre testovacie účely, pričom sa naplní aj tabuľka `weeks`.

6.4.2. Tabuľka `weeks`

Tabuľka `weeks` obsahuje týždne semestra. Je možné túto tabuľku naplniť príkazom `rake tesa:data:aza_setup` pre testovacie účely.

6.4.3. Tabuľka `users`

Tabuľka `users` obsahuje používateľov systému. Odporúča sa administrátora predmetu pridať manuálne, pričom zvyšní používatelia budú pridaní pri ich prvom prihlásení AIS účtom. Tiež je možné pridať používateľov csv súborom pomocou príkazu

```
rake tesa:data:import_users[<<CESTA_K_SUBORU>>]
```

Ukážka súboru je v časti úkážok konfiguračných súborov.

6.4.4. Tabuľka `learning_objects` a `answers`

Každý riadok tabuľky `learning_objects` obsahuje otázku, pričom ak sa jedná o otázku so zadanými odpoveďami, odpovede sa nachádzajú v tabuľke `answers`. Tabuľku je možné plniť pomocou administratívneho rozhrania v aplikácii alebo pomocou príkazu:

```
rake tesa:data:import_tests[<<CESTA_K_SUBORU>>,<<CESTA_K_POUZITYM_OBRAZKOM>>]
```

6.4.5. Tabuľka exercises

V tejto tabuľke sú zadané termíny testov. Termíny testov je možné vytvoriť v konfiguračnom rozhraní TESA alebo príkazom:

```
rake tesa:data:import_exercises[<<CESTA_K_SUBORU>>]
```

7. Ukážky konfiguračných súborov

V tejto časti sú ukážky súborov používaných na konfiguráciu a import.

7.1. Databáza - database.yml

V súbore `/home/aza_user/aza/config/database.yml` sa upravujú prístupové údaje do databázy. Pre nakonfigurovanie prístupu kvôli produkčnému behu stačí upraviť časť týkajúcej sa produkcií:

```
production:
  <<: *default
  database: aleftng_production_aza
  username: aza_db_username
  password: aza_db_password
```

Časť `default` môže byť nakonfigurovaná takto:

```
default: &default
adapter: postgresql
encoding: unicode
# For details on connection pooling, see rails configuration guide
# http://guides.rubyonrails.org/configuring.html#database-pooling
pool: 5
```

Viac informácií je v rails [dokumentácii](#).

7.2. Ldap - ldap.yml

V súbore `/home/aza_user/aza/config/ldap.yml` sa upravujú prístupové údaje na ldap spojenie. Na začiatku je vhodné si zdefinovať prístupové údaje na všetky ldap serveri, napríklad takto:

```
stuba_ldap: &STUBA_LDAP
host: ldap.stuba.sk
port: 636
attribute: uid
base: ou=People,dc=stuba,dc=sk
ssl: simple_tls
```



```
stuba_ldap2: &STUBA_LDAP2
  host: ldap2.stuba.sk
  port: 636
  attribute: uid
  base: ou=People,dc=stuba,dc=sk
  ssl: simple_tls
```

Následne je možné použiť tieto údaje pre hociktoré prostredie, napríklad pre produkciu takto:

```
production:
-
  <<: *STUBA_LDAP
-
  <<: *STUBA_LDAP2
```

Ak sa do časového limitu nepodarí spojiť s ldap serverom označeným ako STUBA_LDAP, tak sa TESA pokúsi spojiť s STUBA_LDAP2.

7.3. Mail - production.yml

V súbore `/home/aza_user/aza/config/enviroments/production.yml` sa upravujú prístupové údaje na spojenie s mail serverom. Pre napríklad vývojárske prostredie treba upraviť súbor `/home/aza_user/aza/config/enviroments/development.yml` a podobne.

Posielanie mailov je riešené pomocou SMTP protokolu. Všetky nastavenia sa nachádzajú v časti `#Mail`. Príklad súboru:

```
#Mail
#Determines whether deliveries are actually carried out.
config.action_mailer.perform_deliveries = true
# Ignore bad email addresses and do not raise email delivery errors.
# Set this to true and configure the email server for immediate delivery
#to raise delivery errors.
config.action_mailer.raise_delivery_errors = true
# Defines a delivery method.
config.action_mailer.delivery_method = :smtp
#SMTP config
config.action_mailer.smtp_settings = {
  address:           'mail.fiit.stuba.sk',
  port:              25,
  domain:            'fiit.stuba.sk',
  enable_starttls_auto: false }

#Allows you to set default values for the mail method options
config.action_mailer.default_options = {
  from: 'team17@fiit.stuba.sk',
  to: 'team17@fiit.stuba.sk' }
```

Ak sa vyžaduje, do `config.action_mailer.smtp_settings` je možné zadať aj `user_name`, `password` a `authentication` (autentifikačná metóda, napríklad `login`). Podrobné vysvetlenie, aj ako použiť inú metódu ako SMTP je rails [dokumentácii](#).

7.4. CSV súbor na prídanie používateľov

Príklad súboru:

```
Login,Role,First name,Last name>Password,Type
lucka,administrator,Maria,Lucka, '',LdapUser
qbecka,teacher,Martin,Becka, '',LdapUser
qlukotka,teacher,Robert,Lukotka, '',LdapUser
qmazak,teacher,Jan,Mazak,qmazak,LocalUser
```

Vysvetlenie formátu:

1. Login - používateľské meno
2. Role - rola používateľa, sú podporované 3 roly: administrator, teacher, student
3. First name - prvé meno používateľa
4. Last name - priezvisko používateľa
5. Password - heslo používateľa, v databáze je uložená len jeho hash hodnota
6. Type - typ používateľského účtu, môže byť LocalUser (vtedy sa neoveruje heslo pomocou LDAP) a LdapUser (správnosť hesla sa overuje pomocou Ldap)

7.5. CSV súbor s definíciami otázok

Príklad súboru, konce riadkov sú označené pomocou “\n”:

```
Title,Question,Local Concepts,Global Concepts,Type,Answers,Difficulty,Image,
Is special,Week\n
Vtipna,Aký je na obrázku pokémon?,offtopic,prednaska,single-choice,pikachu;
venusaur;<correct>charizard</correct>;digglet;persian;hitmonlee,lahke,
/pokemon.png,TRUE,1\n
Misova,Ako hodnotíte náš poster?,offtopic,prednaska,answer-validator,
Poster je skvelý,lahke,,TRUE,1Triediace alg,Ktorý z nasledujúcich
tímov vyhrá?,sorts,cvicenie,multi-choice,<correct>Tesa</correct>;
<correct>Tím 17</correct>;<correct>Osičky</correct>;iný,lahke,,TRUE,1\n
Triediace alg,"Napíšte vaše pocity a nápady, ako vylepšiť náš projekt."
,sorts,cvicenie,open-question,,lahke,,TRUE,1\n
Triediace alg,Kto vyhral v otázke položenej študentom Kto?,sorts,
cvicenie;prednaska,single-choice,Superman;Spiderman;<correct>Batman
</correct>;Ironman;Doktor,lahke,,TRUE,1\n
```

Vysvetlenie formátu:

1. Title - názov otázky
2. Question - samotný text otázky
3. Local Concepts - koncept viazaný na termín
4. Global concepts - koncept viazaný na týždne
5. Type - typ otázky, môže byť: single-choice, multi-choice, answer-validator, open-question
6. Answers - odpovede (oddelené pomocou “;”) na otázku pri otázke typu single-choice a multi-choice; pomocou tagu odpoved je možné označiť správne odpovede
7. Difficulty - náročnosť otázky, môže byť: trivialne, lahke, sredne, tazke, impossible
8. Image - ku každej otázke je možné zadať jeden obrázok, v tejto položke treba zadať názov otázky a pri importe cestu k priečinku so všetkými obrázkami
9. Is special - ak označené ako TRUE, tak túto otázku dostanú všetci študenti, inak FALSE
10. Week - id týždňa z tabuľky week

Rovnice: Do otázok a odpovedí (Question a Answers) je možné formou tex výrazov zadávať rovnice. Rovnica sa vkladá medzi párový tag `/[a /]`.

7.6. CSV súbor s termínmy testov

Príklad súboru:

```
Exercise start,Exercise end,Code,Lecturer,Week,Local Concepts
2016-04-04 14:00:00 +0200,2016-02-15 15:40:00 +0100,10008,3,8,prednaska
2016-04-11 14:00:00 +0200,2016-02-15 15:40:00 +0100,10009,3,9,prednaska
2016-04-18 14:00:00 +0200,2016-02-15 15:40:00 +0100,10010,3,10,prednaska
2016-04-25 14:00:00 +0200,2016-02-15 15:40:00 +0100,10011,3,11,prednaska
2016-05-02 14:00:00 +0200,2016-02-15 15:40:00 +0100,10012,3,12,prednaska
2016-02-15 16:00:00 +0100,2016-02-15 17:40:00 +0100,10013,3,1, cvicenie
2016-02-22 16:00:00 +0100,2016-02-15 17:40:00 +0100,10014,3,2, cvicenie
2016-02-29 16:00:00 +0100,2016-02-15 17:40:00 +0100,10015,3,3, cvicenie
2016-03-07 16:00:00 +0100,2016-02-15 17:40:00 +0100,10016,3,4, cvicenie
```

Vysvetlenie formátu:

1. Exercise start - štart termínu vo formáte YYYY-MM-DD HH-MM-SS zzz
2. Exercise end - koniec termínu vo formáte YYYY-MM-DD HH-MM-SS zzz
3. Code - kód pre študentov na vstup do termínu
4. Lecturer - id učiteľa v tabuľke users
5. Week - id týždňa v tabuľke weeks
6. Local Concepts - koncept viazaný na termíny

7.7. Zmena počtu otázok, ktoré dostane študent na teste

V súbore `home/user_aza/aza/lib/tesa_simple_recommender.rb` označujú riadky:

```
exercise_questions_count = 7
student_questions_count = 4
```

počet otázok pre termín a počet otázok, ktoré dostane študent.

8. Časté chyby

8.1. Spojenie s db

Chyba `PG::ConnectionBad (FATAL: Peer authentication failed for user v /home/user_aza/aza/log/production.log`. Problém je s overovaním usera do databázy, buď je nesprávne heslo alebo nesprávna autentifikačná metóda nastavená. V druhom prípade v súbore `/etc/postgresql/9.3/main/pg_hba.conf` je databázový používateľ najčastejšie nastavený, že používa `local` nastavenie, teda `peer` autentifikačnú metódu, čo z dôvodu bezpečnosti nepodporuje `gem pg`, ktorý vyžaduje MD5 autentifikačnú metódu.

8.2. Pád pri migrácii databázy (update databázy)

Ak nastane chyba

```
ERROR: could not open extension control file "/usr/share/postgresql/9.3/extension/hstore.control": No such file or directory.
```

pri migrácii databázy (napríklad príkazom `rake db:migrate`), je nutné najprv doinštalovať knižnicu `apt-get install postgresql-contrib-9.3` a následne spustiť databázový príkaz `CREATE EXTENSION HSTORE` nad danou databázou.

Free Software, Hell Yeah!

Príloha B

Táto príloha obsahuje vygenerovanú technickú dokumentáciu. Táto technická dokumentácia v plnej forme sa nachádza aj na priloženom médiu.

```
#add_question_concept  
#delete_question_concept  
#download_statistics  
#index  
#question_concept_config  
#setup_config  
#setup_config_attributes  
#setup_config_relations
```

class AdministrationsController

Public Instance Methods

add_question_concept()

delete_question_concept()

download_statistics()

Defines action for downloading statistics get 'admin/setup_config/:setup_id/download_statistics'

```
<tt>params[:setup_id]</tt> => id of setup
```

returns file containing statistics

index()

Specifies action for administration setting main page get 'admin'

question_concept_config()

Specifies action for course settings page get 'admin/question_concept_config/:course_id'

```
<tt>params[:course_id]</tt> => id of course
```

setup_config()

Defines action for setup setting in administration get 'admin/setup_config/:setup_id'

```
<tt>params[:setup_id]</tt> => id of setup
```

setup_config_attributes()

Action that implements submitting new settings for specified setup post 'admin/setup_config/:setup_id/setup_attributes'

```
<tt>params[:setup_id]</tt> => id of setup  
<tt>params[:setup]</tt> => hash containing following attributes  
<tt>params[:setup][:week_count]</tt> => number of weeks in semester  
<tt>params[:setup][:first_week_at]</tt> => date of first week of semester  
<tt>params[:setup][:show_all]</tt> => not implemented
```

setup_config_relations()

Specifies submitting new concept to weeks relations post 'admin/setup_config/:setup_id/setup_relations'

```
<tt>params[:setup_id]</tt> => id of setup  
<tt>params[:relations]</tt> => hash containing set of  
pairs concept-weeks for which relationships have to be saved
```

Parent

ApplicationController

Methods

#create
#destroy
#update

Public Instance Methods

create ()

Specifies action for creating new answer for learning object post 'administrations/learning_objects/:learning_object_id/answers' rescues problems with visibility and correct answers count For example: singlechoice question must have only one correct answer answer is created for specified learning object using params: params[:answer][:answer_text] - answer text params[:answer][:is_correct] - answer correctness info

```
# File app/controllers/administrations/answers_controller.rb, line 13
def create
  begin
    ActiveRecord::Base.transaction do
      @learning_object.answers.create!({
        answer_text: params[:answer][:answer_text],
        is_correct: params[:answer][:is_correct] == "1"
      })
      @learning_object.validate_answers!
    end
  rescue AnswersCorrectnessError
    return redirect_to edit_administrations_learning_object_path id: @learning_object.id, :alert => 'Otázka nesmie mať viac ako jednu správnu odpoveď.'
  rescue AnswersVisibilityError
    return redirect_to edit_administrations_learning_object_path id: @learning_object.id, :alert => 'Otázka nesmie mať viac ako jednu viditeľnú odpoveď.'
  end
  redirect_to edit_administrations_learning_object_path id: @learning_object.id, flash[:notice] => t('global.answers.added')
end
```

destroy ()

Specifies action for deleting learning object answer delete 'administrations/learning_objects/:learning_object_id'

```
# File app/controllers/administrations/answers_controller.rb, line 57
def destroy
  Answer.find_by_id(params[:answer_id]).destroy!
  redirect_to edit_administrations_learning_object_path(id: @learning_object.id), flash[:notice] = t('global.answers.deleted')
end
```

update ()

Specifies action for updating answer of learning object put 'administrations/learning_objects/:learning_object_id/answers/:id' rescues problems with correct answers count

```
# File app/controllers/administrations/answers_controller.rb, line 33
def update
  route = edit_administrations_learning_object_path id: @learning_object.id

  begin
    ActiveRecord::Base.transaction do
      @learning_object.answers.each do |a|
        a.update!({
          is_correct: params["correct_answer_#{a.id}"] == "1",
          answer_text: params["edit_answer_text_#{a.id}"]
        })
      end
      @learning_object.validate_answers!
    end
  rescue AnswersCorrectnessError
    return redirect_to route, :alert => 'Otázka nesmie mať viac ako jednu správnu odpoveď.'
  rescue AnswersVisibilityError
    return redirect_to route, :alert => 'Otázka nesmie mať viac ako jednu viditeľnú odpoveď.'
  end

  redirect_to route, flash[:notice] => t('global.answers.changes_saved')
end
```

class ScoringSystem::CompleteTestScoring

Public Class Methods

doScoring(exercise_id, user_id)

Grants points so that student can get 1 point or none, excluding open questions.

```
# File lib/scoring_system/complete_test_scoring.rb, line 5
def self.doScoring(exercise_id, user_id)
  rels=UserToLoRelation.where("user_id= (?) AND exercise_id= (?)",user_id,exercise_id)

  correct= true
  rels.each do |relation|
    if relation.type == "UserFailedLoRelation"
      correct=false
    end
  end
  if correct
    rels.each do |relation|
      if relation.type== "UserSolvedLoRelation"
        relation.points= 1.0/rels.size
        relation.save
      end
    end
  else
    rels.each do |relation|
      relation.points=0
      relation.save
    end
  end
end
end
```


Parent

LearningObject

Methods

#get_solution
#right_answer?

class EvaluatorQuestion

Public Instance Methods

get_solution(user_id)

Computes solution of evaluator question for specified student. This is done by averaging all answers, which are not his.

```
# File app/models/evaluator_question.rb, line 5
def get_solution(user_id)
  rels = UserSolvedLoRelation.where('learning_object_id = (?) AND user_id <> (?)', self.id, user_id).order(created_at: :desc)
  values = Hash.new
  rels.each do |r|
    if values[r.user_id].nil?
      values[r.user_id] = r.interaction.to_i
    end
  end

  return nil if values.empty?

  values.values.reduce(:+).to_f / values.size
end
```

right_answer?(answer, solution)

class LdapUser

Constants

LDAP_AIS_EMAIL

LDAP_EMAIL

'mail' entry FOR A SINGLE USER can look like this (in any order):

```
name.surname@stuba.sk, 2007123456@is.stuba.sk, 2007123456@stuba.sk, xloginn@stuba.sk, xloginn@is.stuba.sk, name.surname@is.stuba.sk, 67890@stuba.sk, 67890
```

extract first non-numeric @stuba.sk and @is.stuba.sk address

Public Instance Methods

ldap_before_save()

```
# File app/models/users/ldap_user.rb, line 5
def ldap_before_save
  ldap_entry = Devise::LDAP::Adapter.get_ldap_entry(self.login)

  # sadly, Net::LDAP::Entry doesn't have #map
  entry = {}
  ldap_entry.each do |param|
    x = ldap_entry[param]
    entry[param] = (x.length == 1 ? x.first : x)
  end

  self.last_name = entry[:sn]
  self.first_name = entry[:givenname] # 'givenName' before symbol lowercasing
  self.aisid = entry[:uisid] # 'uisID' before symbol lowercasing
  if self.student?
    self.role = (( !entry[:host].include?('fiit-stud') && entry[:host].include?('fiit-zam')) ? ROLES[:TEACHER] : ROLES[:STUDENT] )
  end

  mails = extract_ldap_mails(entry[:mail])
  self.email = mails[:email]
  self.ais_email = mails[:ais_email]
end
```

class MultiChoiceQuestion

Public Instance Methods

construct_righ_hash()

Constructs hash for right answer, which has same format as hash posted in student response.

```
# File app/models/multi_choice_question.rb, line 9
def construct_righ_hash
  #{"8"=>"8", "10"=>"10", "11"=>"11"}
  answers= Answer.where("learning_object_id = (?) AND is_correct = true",self.id)
  response= ''
  answers.each do |key, val|
    response << "\"#{key.id}\"=>\"#{key.id}\"", "
  end
  response=response.chomp(',')
  response << ''
  return response
end
```

get_solution(user_id)

Returns correct options

```
# File app/models/multi_choice_question.rb, line 4
def get_solution(user_id)
  self.answers.where(is_correct: true).ids
end
```

right_answer?(answer, solution)

```
# File app/models/multi_choice_question.rb, line 21
def right_answer?(answer, solution)

  if answer == nil
    return solution.empty?
  end

  answer = answer.values.map { |n| n.to_i }
  solution == answer

end
```

Parent

LearningObject

Methods

#construct_righ_hash
#get_solution
#right_answer?

class ScoringSystem::SatanScoring

Parent

ScoringSystem::Scoring

Methods

doScoring

Public Class Methods

doScoring(exercise_id, user_id)

Grants points while penalizing guessing by subtracting point for incorrect options

```
# File lib/scoring_system/satan_scoring.rb, line 5
def self.doScoring(exercise_id, user_id)
  rels=UserToLoRelation.where("user_id= (?) AND exercise_id= (?)",user_id,exercise_id)

  rels.each do |relation|
    of_correct_options= Answer.where("learning_object_id=(?) AND is_correct= true",relation.learning_object.id).count
    of_incorrect_options= Answer.where("learning_object_id=(?) AND is_correct= false",relation.learning_object.id).count
    correct_answers= Answer.where("learning_object_id=(?) AND is_correct= true",relation.learning_object.id)

    if(relation.learning_object.type!= "MultiChoiceQuestion")
      if relation.type== "UserSolvedLoRelation"
        relation.points= 1.0
      elsif relation.type== "userFailedLoRelation"
        relation.points= 0
      end
    else
      points = 0.0
      relhash= eval(relation.interaction)
      relhash.each do |key,array|
        if correct_answers.where("id= (?)",key).empty?
          points+= -1.0/of_incorrect_options
        else
          points+= 1.0/of_correct_options
        end
      end
      if points > 0
        relation.points= points
      else
        relation.points= 0
      end
    end
    relation.save
  end
end
```

class ScoringSystem::Scoring

Parent

Object

Methods

doScoring
doScoringForExercise

Public Class Methods

doScoring(exercise_id, user_id)

Computes points for specific test. If answer is entirely correct it grants one point if not it grants no points

```
# File lib/scoring_system/scoring.rb, line 5
def self.doScoring(exercise_id, user_id)
  rels=UserToLoRelation.where("user_id= (?) AND exercise_id = (?)",user_id,exercise_id)

  rels.each do |relation|
    if relation.type == "UserFailedLoRelation"
      relation.points= 0
    elsif relation.type == "UserSolvedLoRelation"
      relation.points= 1.0
    end
    relation.save
  end
end
```

doScoringForExercise(exercise_id)

Method that scores test for all students. It invokes doScoring for every student

```
# File lib/scoring_system/scoring.rb, line 19
def self.doScoringForExercise(exercise_id)
  rels=UserToLoRelation.where("exercise_id= (?)",exercise_id)

  uids= rels.select(:user_id).distinct

  uids.each do |uid|
    doScoring(exercise_id,uid.user_id)
  end
end
```

Parent

Devise::SessionsController

Methods

```
#create
#destroy
#new
#redirect_away
```

Public Instance Methods

create ()

POST /resource/sign_in

```
# File app/controllers/users/sessions_controller.rb, line 14
def create
  # super

  user_class = User.guess_type(params['local_user']['login'], request)
  request.params['ldap_user'] = request.params['local_user']
  self.resource = warden.authenticate! scope: user_class, recall: "#{controller_path}#new"
  log_info request.params['ldap_user']['login'].to_s + " signed in successfully"
  set_flash_message(:notice, :signed_in) if is_flashing_format?
  sign_in(user_class, self.resource)
  respond_with self.resource, :location => after_sign_in_path_for(self.resource)
end
```

destroy ()

DELETE /resource/sign_out

Calls superclass method

```
# File app/controllers/users/sessions_controller.rb, line 27
def destroy
  user = current_user
  super
  log_info user.login + " signed out successfully"
end
```

new ()

GET /resource/sign_in

```
# File app/controllers/users/sessions_controller.rb, line 9
def new
  self.resource = User.new
end
```

redirect_away ()

```
# File app/controllers/users/sessions_controller.rb, line 4
def redirect_away
  redirect_to root_path
end
```

Parent

[LearningObject](#)

Methods

[#construct_righ_hash](#)
[#get_solution](#)
[#right_answer?](#)

class SingleChoiceQuestion

Public Instance Methods

construct_righ_hash()

Constructs hash for right answer, which has same format as hash posted in student response.

```
# File app/models/single_choice_question.rb, line 9
def construct_righ_hash
  return Answer.where("learning_object_id = (?) AND is_correct= true",self.id).first.id
end
```

get_solution(user_id)

Returns correct options

```
# File app/models/single_choice_question.rb, line 4
def get_solution(user_id)
  self.answers.where(is_correct: true).ids
end
```

right_answer?(answer, solution)

```
# File app/models/single_choice_question.rb, line 13
def right_answer?(answer, solution)
  answer.to_i == solution[0]
end
```

```
#admit_student_answer
#extract_names
#list_answers
#list_questions
#process_legend
#process_multi_choice_answers
#process_open_answers
#process_single_choice_answers
#show
#statistics
#submit_regexp
```

class TeachingsController

Public Instance Methods

admit_student_answer()

Defines action to admit points to student answer post 'teaching/:id/admit_answer'

```
<tt>params[:id]</tt> => id of specified question
<tt>params[:student_answer_id]</tt> => id of student answer
<tt>params[:student_answer_points]</tt> => points to give
```

extract_names(legend)

list_answers()

Action for list of all answers for specified question get 'teaching/:id/answers'

```
<tt>params[:id]</tt> => id of specified question
```

list_questions()

Defines action for list of questions get 'teaching/questions'

process_legend(question, answers)

process_multi_choice_answers(question, legend)

process_open_answers(question)

process_single_choice_answers(question, legend)

show()

Specifies action for teachers main page get 'teaching'

statistics()

Specifies action for statistics of a single exercise get 'exercise/statistics'

```
<tt>params[:id]</tt> => exercise id
```

submit_regexp()

Specifies action for posting regexp and subsequent evaluation of open questions post 'teaching/:id/submit_regexp'

```
<tt>params[:id]</tt> => id of specified question
<tt>params[:regexp]</tt> => string with regular expression
```


class RecommenderSystem::TesaSimpleRecommender

Parent

RecommenderSystem::Recommender

Methods

```
::setup  
#access_key  
#get_list  
#random_value  
#random_value_seed  
#use_global_concepts  
#use_local_concepts
```

Public Class Methods

```
setup(user_id, week_id, access_key, use_local_concepts, use_global_concept)
```

Calls superclass method [RecommenderSystem::Recommender.setup](#)

Public Instance Methods

```
access_key()
```

```
get_list()
```

Specifies method for returning generated list of test questions questions are selected using 1 of 3 strategies and their combinations questions are selected from sets of questions mapped with week's or term's concepts. on each term specified questions set is generated from which students get questions each student gets specified number of these questions - using random with `user_id`

```
random_value(range)
```

```
random_value_seed(range)
```

```
use_global_concepts()
```

```
use_local_concepts()
```

class Administrations::UsersController

Parent

[ApplicationController](#)

Methods

[#create](#)
[#destroy](#)
[#edit](#)
[#index](#)
[#new](#)
[#update](#)

Public Instance Methods

`create()`

Specifies action for saving new created user post 'administrations/users/:user_id/create' checks if filled password are identical and if user login is not already used checks some empty fields too uses params posted in `params[:user]`

`destroy()`

`edit()`

Specifies action for editing existing user account get 'administrations/users/:user_id/edit' user for editing is selected using `params[:user_id]`

`index()`

Specifies action for listing all users get 'administrations/users'

`new()`

Specifies action for new user get 'administrations/users/new'

`update()`

Specifies action for updating user account patch 'administrations/users/:user_id/update' updates user using params in `params[:user_id]` checks if filled password are identical

Parent

ApplicationController

Methods

#enter_test
#index
#list
#show
#test_list

class WeeksController

Public Instance Methods

`enter_test()`

`index()`

Specifies action for showing home page in different situations get 'i' home page is different for users with different role for student role, home page is set according to existence of available tests

`list()`

Specifies action for listing weeks info in weeks page get 'w'

`show()`

`test_list()`

Specifies action for listing tests already written by student get 'tests'

class ExercisesController

Public Instance Methods

answers ()

Specifies action for exporting answers to csv get 'exercises/:id/answers' param[:id] - id of exercise responds with csv file if any answers exist

options ()

Specifies action for options of exercise get 'exercise/:id/options' used for setting exercise times: test length and cooldown time

refresh ()

Specifies action for ajax call refreshing counter and test timer get 'exercises/event/refresh' called from view's js

results ()

Specifies action for exercise results get 'exercise/results' shows results for exercise with id from params[:id]

show ()

Specifies action for exercise show page get 'exercises/show' shows exercise with id from params

update ()

Specifies action for exercise update get 'exercises/edit' called from different actions in exercise show view params[:stats] - if defined redirects to statistic page <tt>params<tt> - if defined redirects to exercise results page when any params posted in request - change state of exercise if test not yet started: start test if test started: stop test

update_options ()

Specifies action for saving updated options for exercise post 'exercise/:id/update_options' updates options from params params[:exercise][:options] - params

Parent

ApplicationController

Methods

#answers
#options
#refresh
#results
#show
#update
#update_options

Parent

ActiveRecord::Base

Included Modules

Exceptions

Methods

```
#allow_new_correctness?
#allow_new_visibility?
#construct_righ_hash
#link_concept
#next
#next_by_hybrid
#previous
#seen_by_user
#successfulness
#url_name
#validate_answers!
```

class LearningObject

Constants

DIFFICULTY

TYPE

Public Instance Methods

allow_new_correctness?()

Checks if new answer can be tagged as correct

allow_new_visibility?()

Checks if new answer can be tagged as visible

construct_righ_hash()**link_concept(concept)****next(week_number)****next_by_hybrid(week_number, user_id)**

Gets next week according to HybridRecommender

previous(week_number)**seen_by_user(user_id)****successfulness()**

Returns average success rate for question

url_name()**validate_answers!()**

Validates answers For single choice question it checks if there is only one correct For evaluator question it checks if only one answer is visible

class Administrations::LearningObjectsController

Parent

ApplicationController

Methods

[#create](#)
[#csv_question_import](#)
[#destroy](#)
[#edit](#)
[#index](#)
[#new](#)
[#update](#)

Public Instance Methods

create()

Specifies action for saving created learning object post 'administrations/learning_object/:learning_object_id/create' rescues invalid records - properties not filled in correctly

csv_question_import()

Specifies action for importing questions from csv file and images uploaded using zip post 'import_question_csv' opens zip and imports question using import tasks
params[:test_data] - zip file containing subfolder img with images and one csv

destroy()

Specifies action for deleting learning object delete 'administrations/learning_objects/:learning_object_id/destroy' learning object is selected using its id in param[:id]

edit()

Specifies action for editing existing learning object get 'administrations/learning_objects/:learning_object_id/edit'

index()

Specifies action for showing list of available learning objects get 'administrations/learning_objects'

new()

Specifies action for creating new learning object get 'administrations/learning_object/new'

update()

Specifies action for update of learning objects patch 'administrations/learning_objects/:learning_object_id/update' updates all attributes of learning object specified in learning_object_params rescues problems with not filled required attributes

Parent

ApplicationController

Methods

```
#check_code
#evaluate
#log_time
#next
#show
#show_answers
#show_image
#show_test
#submit_test
```

class QuestionsController

Public Instance Methods

check_code ()

Specifies question for checking test access code post 'check_code' checks test state: exists and not finished

evaluate ()

Specifies evaluation of question in learning post 'w/:week_number/:id/evaluate_answers' params[:week_number] - week number info <tt>params<tt> - id of learning object evaluates questions submitted answers for autoevaluating types of questions

log_time ()

next ()

Specifies action for redirecting to next question get 'w/:week_number/next' next question is determined using recommender system params[:week_number] - week number info

show ()

Specifies action for showing questions in learning get 'w/:week_number/:id' shows one learning object for week preparation params[:week_number] - week number info <tt>params<tt> - id of learning object calls functions for specifying next and previous questions

show_answers ()

Specifies action for showing evaluated answers get 'test/:exercise_code/answers' questions are showed only after specified cooldown time

show_image ()

Specifies action for showing images in question text get 'learning_objects/:id/image' params[:id] - learning object id sends image data created from database bytea

show_test ()

Specifies action action for showing test get 'test/:exercise_code' params[:exercise_code] - access key for test test has to be started and not submitted by same student questions are generated using recommender system for test questions

submit_test ()

Specifies action for submitting test post 'test/:exercise_code/submit' params[:exercise_code] - access key for test before submitting: checks correct params, test state and previous submitting autoevaluating types of questions are evaluated and user relations are updated with result information

[Home](#)

[Pages](#) [Classes](#) [Methods](#)

Search

Parent

ActiveRecord::Base

Methods

[#change_concepts](#)
[#generatecode](#)
[#test_length_defined?](#)
[#test_time_left](#)
[#unavailable_answers?](#)

class Exercise

Public Instance Methods

`change_concepts (concepts)`

`generatecode ()`

Generates random, unique code for test

`test_length_defined? ()`

`test_time_left ()`

Calculate time left for test

`unavailable_answers? (user)`

Tests whether answers to this test are available. Checks for ending of test and cooldown period

Validate

Generated by RDoc 4.2.0.

Based on Darkfish by Michael Granger.