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

0	bsah					
1	Úvod4					
2	Slovník pojmov5					
3	Globáln	e ciele pre zimný semester				
4	Celkový	pohľad				
	4.1 Naš	e zmeny9				
	4.2 Pro	totyp10				
5	Moduly	systému12				
	5.1 Pro	totyp				
	5.1.1	Modul administrátor12				
	5.1.1.1	1 Pridávanie testových otázok12				
	5.1.2	Modul vyučujúci				
	5.1.2.2	1 Učiteľ a admin nemôžu písať test13				
	5.1.2.2	2 Cvičiaci zobrazí prístupový kód aktuálneho testu14				
	5.1.2.3	3 Cvičiaci zobrazí anonymizované odpovede študentov14				
	5.1.2.4	4 Cvičiaci po ukončení testu môže zobraziť základnú štatistiku odpovedí				
	5.1.2.	5 Prihlasovanie cvičiacich cez konfiguráciu16				
	5.1.3	Modul študent				
	5.1.3.2	1 Študent sa prihlási cez AIS LDAP17				
	5.1.3.2	2 Študent odpovedá na otázky s otvorenou odpoveďou18				
	5.1.3.3	3 Študent odpovedá na otázky s viacerými možnosťami18				
	5.1.3.4	4 Študent si zobrazí test s otázkami19				
	5.1.3.	5 Systém neumožní nič okrem povolených akcií20				
	5.1.3.0 štatist	5 Vynechanie prázdnych odpovedí multichoice otázok pri generovaní testu a zobrazení iky 21				
	5.2 Ale	fTNG22				
	5.2.1	Modul Administrátor				

	5.2.1.1	Administrátor pri štarte systému nahrá súbor s cvičeniami	22	
	5.2.1.2	Administrátor vytvorí termíny cvičení počas semestra	23	
	5.2.1.3	Administrátor vloží termíny cvičení a otázky do systému	24	
5	.2.2	Modul vyučujúci	25	
	5.2.2.1	Cvičiaci zobrazí test na cvičení	25	
	5.2.2.2	Cvičiaci ukončí test na cvičení	25	
	5.2.2.3	Cvičiaci si zobrazí zoznam svojich termínov cvičení	26	
	5.2.2.4	Učiteľ a admin nemôžu písať test	27	
	5.2.2.5	Cvičiaci skontroluje dochádzku študentov na cvičení	27	
	5.2.2.6	Učiteľ na domovskej stránke môže spustiť test	28	
5	.2.3	Modul študent	28	
	5.2.3.1	Študent môže prísť na hociktoré cvičenie	28	
	5.2.3.2	Študent po odoslaní odpovede nemôže test písať druhýkrát	29	
	5.2.3.3	Študent odošle svoju odpoveď na test	30	
	5.2.3.4	Študent odpovedá na obrázkovú otázku	30	
	5.2.3.5	Študent odpovedá na otvorené otázky	31	
	5.2.3.6	Študent odpovedá na výberové otázky	32	
	5.2.3.7	Študent si v rámci testu zobrazí pridelené otázky	32	
	5.2.3.8	Študent si zobrazí aktuálny test	33	
	5.2.3.9	Všetci študenti dostanú špeciálnu zhodnú otázku alebo zhodné otázky na teste	33	
	5.2.3.10	0 Študent dostane otázku podľa zadaného kľúča	34	
Ρ	oužívan	ie prototypu – testovanie reálnymi používateľmi	36	
6.1	Prieb	peh testovanie vedomostí študentov	36	
6.2	Obra	izovky prototypu – študent	36	
6.3	Obra	izovky prototypu – Vyučujúci	41	
6.4	4 Štatistiky			

6

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 prispôsobiť 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
ΑΡΙ	Skratka pre Application programming interface
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 Lightweight Directory Access Protocol je protokol na ukladanie a prístup k dátam na adresárovom serveri. Jeho súčasťou je autentifikácia klienta.
import	Prevod elektronicky 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 obrázok

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.

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 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-controler. Dátový model, logika a používateľské rozhranie sú tím 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.



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.

4.1 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 jeden krát. 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. 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.

Administrátori majú možnosť nahrať 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).

Vyučujúci má okrem ovládania prístupu k testu možnosť aj zobraziť 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 zobraziť, ale nie absolvovať. V pôvodnej verzií bol vyučujúci braný rovnako ako študent.

4.2 Prototyp

Prototyp nášho tímového projektu slúži na overenie si funkcionalít a správnosti našich biznis plánov skutočnými používateľmi. Pomocou neho chceme získať primárne spätnú väzbu od študentov a učiteľov. Taktiež ním cielime overiť, či je používateľské rozhranie dostatočne jednoduché a ľahko používateľné. V neposlednom rade ale potrebujeme zistiť ako bude naša aplikácia reálne vplývať 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 naň 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 PosgreSQL. Informácie medzi týmito



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 <u>https://tesa.fiit.stuba.sk/os</u>.

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 prihlasovanie 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:

 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 neprebieha cvičenie a nezobrazí sa mu ani kód testu. 2. 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:

- 1. Po prihlásení do aplikácie sa presmeruje na obrazovku na zadanie kódu testu resp. cvičenia, ktorý pre nich cvičiaci zverejnil.
- 2. 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

5 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.

5.1 Prototyp

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

5.1.1 Modul administrátor

5.1.1.1 Pridávanie testových otázok

Analýza

Prototyp má za úlohu testovať študentov počas cvičenia. Administrátor predmetu v spolupráci s cvičiacimi každý týždeň pripravia 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 pridelí 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 querry.sql

Generátor predpokladá prítomnosť otázky s id (stĺpec name) v tvare steinmul-NN-01, kde NN je číslo týždňa, ktorú pridelí 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 vyraz "|" výrazom "\n" v texte otázky a v odpovediach.

Testovanie

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

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

5.1.2 Modul vyučujúci

5.1.2.1 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.

5.1.2.2 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.

5.1.2.3 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.

5.1.2.4 Cvičiaci po ukončení testu môže zobraziť 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 prototype tak môžeme odstrániť zistené nedostatky a doplniť chýbajúcu funkcionalitu 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 prototype 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 prototype. 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.

5.1.2.5 Prihlasovanie cvičiacich cez konfiguráciu

Analýza

Cvičiaci, ktorý 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 obídenie 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.

5.1.3 Modul študent

5.1.3.1 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 chybnej adresy primárneho servera.

5.1.3.2 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 správnemu zobrazeniu v aplikácii. Odpovede na otázky boli vkladané taktiež s rôznou dĺžkou, špeciálnymi znakmi, prípadne prázdne.

5.1.3.3 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 ktorý 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 správnemu zobrazeniu v aplikácii a rôznym počtom vybraných možností. Odpovede na otázky boli vyberané v rôznom poradí a následne porovnávané s dátami uloženými do databázy.

5.1.3.4 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.

5.1.3.5 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.

5.1.3.6 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 funkcionalitu 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.

5.2 AlefTNG

5.2.1 Modul Administrátor

5.2.1.1 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 pridelený 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 dispozícií 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

- 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.

5.2.1.2 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.

5.2.1.3 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 prideľujú týždňom podľa pravidla z pôvodného prototypu – jedna téma na konkrétny týžděň. 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.

5.2.2 Modul vyučujúci

5.2.2.1 Cvičiaci zobrazí test na cvičení

Analýza

Cvičiaci zobrazí test na cvičení. Po prihlásení si cvičiaci môže zobraziť 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 funkcionalita zobrazovania zoznamu cvičení. V zozname cvičení sa prirodzene objavujú len testy, ktoré patria danému cvičiacemu. Cvičiaci má k dispozícií všetky záznamy počas celého semestra. Výnimkou je administrátorská rola, ktorá má k dispozícií 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 opätovného spustenia testu.

5.2.2.2 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 funkcionalita 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.

5.2.2.3 Cvičiaci si zobrazí 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.

5.2.2.4 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 píšu testy.

Riešenie

Upravili sme časť kódu zodpovednú za prideľovanie 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.

5.2.2.5 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.

5.2.2.6 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 funkcionalita.

5.2.3 Modul študent

5.2.3.1 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 priraďovaní 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 aplikácií 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 úž 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.

5.2.3.2 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 opätovne 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 zobraziť. 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.

5.2.3.3 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 funkcionalita 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.

5.2.3.4 Študent odpovedá na obrázkovú otázku

Analýza

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

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 funkcionalita 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 funkcionalitu 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.

5.2.3.5 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.

5.2.3.6 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 správa. 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 funkcionalita pôvodného systému.

Testovanie

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

5.2.3.7 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.

5.2.3.8 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.

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

Analýza

V niektorých prípadoch sa vyžaduje doplnenie prideľovania 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 vybratých 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 vybratých otázok musí obsahovať aj tieto otázky. Medzi vybraté 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 prideľovanie 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 nespadajú 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.

5.2.3.10 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 prideľuje 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≤m≤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 píšu test. Študent po spustení testu dostáva na svojej obrazovke výstup v podobe zadania testu, ktorý obsahuje jednotlivé otázky vybraté na základe kritérií spomenutých vyššie. Otázky pre test sa nemenia ani pri obnovení stránky testu (refresh). Test môže študent následne vypracovávať.

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 priraďovaním 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 Používanie prototypu – testovanie reálnymi používateľmi

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

6.1 Priebeh testovanie 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

xmacejkova

heslo

AIS Login

-			
	ΝЛ		
	1 1 1		S

Prototyp

Vitajte v našej aplikácii pre Operačné systémy 1.0

Otestuj sa a nazbieraj body!

je to strašná zábava

Zadaj kód cvičenia

kód

Over a prejdi na otázky

≡ MENU

Prototyp

Vitajte v našej aplikácii pre Operačné systémy 1.0

Otestuj sa a nazbieraj body!

je to strašná zábava

Zadaj kód cvičenia

60377119

Over a prejdi na otázky

≡ MENU (Otázky	pre 8. ty	ýždeň
----------	--------	-----------	-------

Otázka číslo 1.
Vyberte klucovy pojem, pre ktory je dane tvrdenie typicke. Hlavnym cielom je udrzat procesor v stave "busy" pocas doby pritomnosti procesov pripravenych na vykonanie. Multiprocessing Multitasking Multithreading Multiprogramming Ziadne z uvedenych
Otázka číslo 2.
Otázka číslo 3.
Otázka číslo 4.
Odovzdaj

	Otázky pre 8. týždeň
Otázka číslo 1.	
Otázka číslo 2.	
Otázka číslo 3.	
Otázka číslo 4.	
Vyjadrite svoj vzta v tesim sa ked ju uz by som bez slovami sa vyja c chcem si s nou s nou je moj de	h k aplikacii prototypu TESA opat uvidim nej nedokazal zit drit neda i stale pisat en krajsi

Odovzdaj

≡ MENU	Otázky pre 8. týždeň
Otázka číslo	01.
Otázka číslo	2.
0	The page at https://tesa.fiit.stuba.sk says: Úspešne si odovzdal test a automaticky si sa odhlásil. OK
 ✓ slovami sa ✓ chcem si s ✓ s nou je mo 	vyjadrit neda nou stale pisat ij den krajsi

Odovzdaj

Operačné systémy 1.0		≡ MENU
O nás	>	On
Otázky	>	Οp
Odhlásiť sa	>	
		Zi
		AIS login
		xmacejkova
		heslo
		•••••

Prototyp

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.



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 súčastnú situáciu cvičení.

6.3 Obrazovky prototypu – Vyučujúci

Operačné systémy 1.0	Kód				
O nás 👂	Kód ovičenie				
Otázky >	Kou cvicenia				
Odhlásiť sa 📏					
	tesa.fiit.studa.sk/os				
	60377119				
	Uzavri test a prejdi na štatistiku				
Operačné systémy 10	Kod				
O nás 📏	lia				
Otázky	The page at https://tesa.fiit.stuba.sk says:				
Odhlásiť sa >	Prevent this page from creating additional dialogs.				
	Jouru				

Operačné systémy		Kód
O nás	>	
Otázky	>	The page at https://tesa.fiit.stuba.sk says:
Odhlásiť sa	>	Tect bol uzatvorný Prevent this page from creating additional dialogs. OK I Izravri test a preirlí na štatistiku

Operačné systémy 🗤		Graf	
O nás	>	Otázka 1 Otázka 2 Otázka 3 Otázka 4	
Otázky	>	Vyberte klucovy pojem, pre ktory je dane tvrdenie typicke. Hlavnym cielom je udrzat procesor v stave	
Odhlásiť sa	>	busy pocas doby pritomnosti procesov pripravených na vykonánie.	
		Graf odpovedí: Multiprocessing	



6.4 Štatistiky



Nasledovné štatistiky pochádzajú z konkrétnych odpovedí študentov na testoch.

DopInte spravne do tvrdenia: Pri pouziti algoritmu vyberu obete 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	najdlhšie nepužitá
1	posledná použitá
1	najmenej používatelná
1	aktuálna
1	naposledy pouzita
1	Pracovna



Vyberte nutne podmienky na riesenie vzajomneho vylucovania