

VELEUČILIŠTE U RIJECI
SPECIJALISTIČKI STUDIJ INFORMACIJSKE TEHNOLOGIJE U POSLOVNIM
SUSTAVIMA

Programsko inženjerstvo

„MojIspit“

Jelena Benić
Valentina Klepac
Nikola Rodeš

U Rijeci, 2015.

Sadržaj

1. UVOD	3
2. SASTAV PROJEKTOG TIMA I ULOGE U TIMU	4
3. PREDUVIJETI ZA IZRADU	5
4. SPECIFIKACIJA SUSTAVA	7
4.1. DIJAGRAMI DEKOMPOZICIJE I BAZA PODATAKA	7
4.2. Početni relacijski dijagram baze podataka.....	11
4.3. SQL upiti	12
5. DIZAJN I IMPLEMENTACIJA.....	14
5.1. Aktivnosti dizajna.....	14
5.2. Dizajn podataka, relacijska baza podataka	15
6. DIZAJN SUČELJA	17
7. Faza testiranja.....	21
8. Evolucija.....	22

1. UVOD

Ocjenjivanje znanja u školskim ustanovama provodi se od samog početka školovanja. Provjere znanja pišu sami nastavnici, profesori ili učitelji ovisno o svom planu i programu rada i o kolegiju za koji su odgovorni. U većini slučajeva iste provjere znanja pišu se na papiru. Razvijanje tehnologije i njezino korištenje u svakodnevnicima dovele su do ideje za izgradnjom sustava za provjeru znanja na računalu. Putem sustava bilo bi omogućeno izraditi cijeli sustav za provjeru znanja za jedan predmet. Učitelji bi unosili pitanja i moguće odgovore i kreirali ispite koje bi učenici kada bi došlo vrijeme mogli ispunjavati. Olakšalo bi se i nastavnicima i učenicima ispunjenje obaveze za provjeru znanja, omogućilo bi se brže ocjenjivanje, lakša pohrana ispita i lakša provjera stečenih ocjena.

2. SASTAV PROJEKTOG TIMA I ULOGE U TIMU

Nikola Rodeš- voditelj tima

Jelena Benić

Valentina Klepac

Poslovi unutar tima su raspoređeni ovisno o prethodno stečenom znanju svakog od članova i želji da pojedini član unutar tima nauči nešto novo.

Voditelj tima, Nikola Rodeš, ujedno je bio zadužen i odgovoran za izgradnju djela sustava vezan za autorizaciju i administraciju. Prije početka implementacije bio je zadužen za proučavanje i odlučivanje o okruženju unutar kojeg sustav bude implementiran. Nakon dogovora s timom odlučeno je da se sustav bude izgradio pomoću Yii Frameworka i razvojnog okruženja NetBeans. Uz autorizaciju i administraciju bio je zadužen za vođenje korisničkih računa i pojedinih izgleda unosnih formi.

Jelena Benić, bila je zadužena za izgradnju relacijske baze podataka i sve izmjene koje su se vršile na bazi tokom izgradnje, ovisno o potrebama sustava. Uz samu izgradnju odgovornost je bila i kod kreiranja svih funkcionalnosti vezanih uz bazu koje su se morale same naknadno izgraditi.

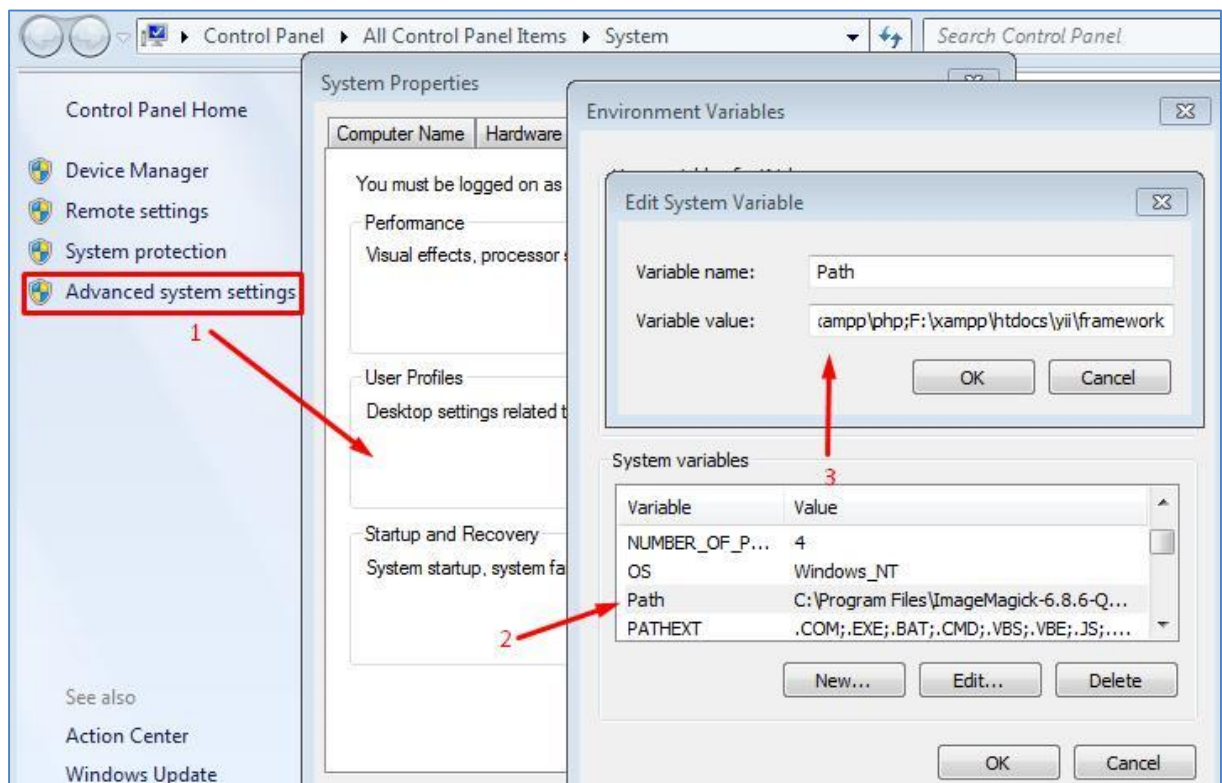
Valentina Klepac, prvobitna zaduženost bila je generirati sve potrebne modele i crude operacije putem gii generatora. Generiranje joj je omogućila sama baza postavljena na lokalni server. Nakon generiranja i početka izgradnje imala je odgovornost urediti i izgraditi cjelokupni dizajn sučelja. Također je formirala izgled svih formi koje su se nalazile unutar aplikacije.

Zajednički rad sastojao se od testiranja rada sustava, pisanja projektne dokumentacije, ispuna baze podataka, i odlučivanja o mogućim nadogradnjama sustava. Svako je testirao pojedini dio sustava, ujedno popunio pojedini dio baze podataka te nakon testiranja napisao projektne dokumentaciju.

3. PREDUVIJETI ZA IZRADU

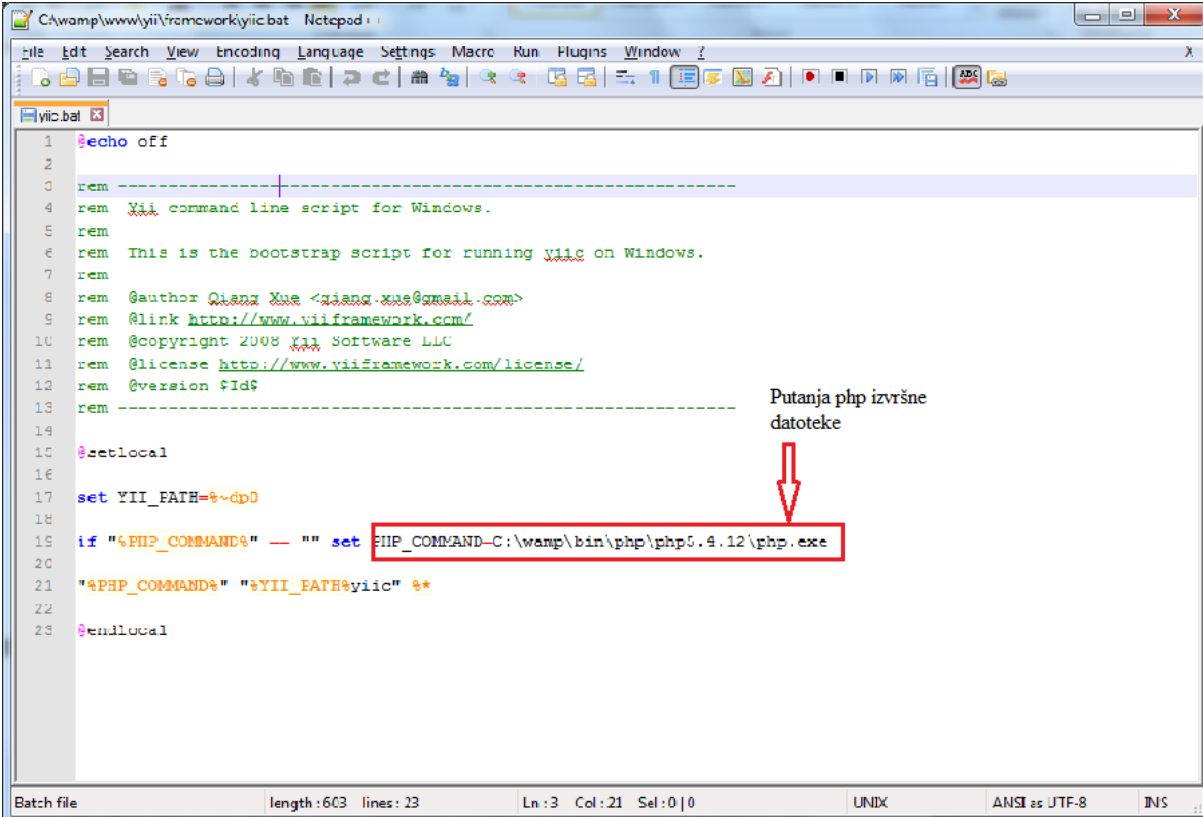
Za izgradnju sustava za provjeru znanja bilo je potrebno proučiti što je sve potrebno za izgradnju sustava, odabrati okruženje za rad i upoznati se s načinom na koji će se sam sustav izgraditi. Prilikom početka izrade prvobitno je bilo potrebno osmisliti sam sustav funkcionalnosti koje će se koristiti unutar sustava, te putem toga izgraditi i samu bazu podataka. Baza podataka izgrađena je unutar MySQL Workbench. Prikaz svih funkcionalnosti i odnos između procesa te dijagrami aktivnosti izgrađeni su putem online servisa za crtanje dijagrama, draw.io.

Nakon izgradnje sustava unutar baze podataka, putem istraživanja odabran je yii framework, xampp server, Netbeans za programsko okruženje za rad. Prvobitno je bilo potrebno instalirati xampp server i omogućiti rad na lokalnom računalu. Nakon instalacije servera, bilo je potrebno preuzeti paket za instalaciju yii frameworka na računalo. Paket koji se preuzme putem službenih stranica je arhiviran te ga je potrebno raspakirati unutar datoteke htdocs unutar xampp datoteke. Također je potrebno i dodati još jednu sistemsku varijablu unutar postavki za računalo.



Slika 1 dodavanje sistemske varijable

Unutar datoteke koja sadrži sve potrebne dokumente za rad yii – a, potrebno je konfigurirati dokument yiic.bat, unutar dokumenta potrebno je dodati putanje za korištenje php izvršne datoteke.



```
1 @echo off
2
3 rem -----
4 rem Yii command line script for Windows.
5 rem
6 rem This is the bootstrap script for running yiic on Windows.
7 rem
8 rem @author Qiang Xue <qiang.xue@gmail.com>
9 rem @link http://www.yiiframework.com/
10 rem @copyright 2008 Yii Software LLC
11 rem @license http://www.yiiframework.com/license/
12 rem @version $Id$
13 rem -----
14
15 %setlocal
16
17 set YII_PATH=%~dp0
18
19 if "%PHP_COMMAND%" == "" set PHP_COMMAND=C:\wamp\bin\php\php5.4.12\php.exe
20
21 "%PHP_COMMAND%" "%YII_PATH%yiic" %*
22
23 %endlocal
```

Putanja php izvršne datoteke

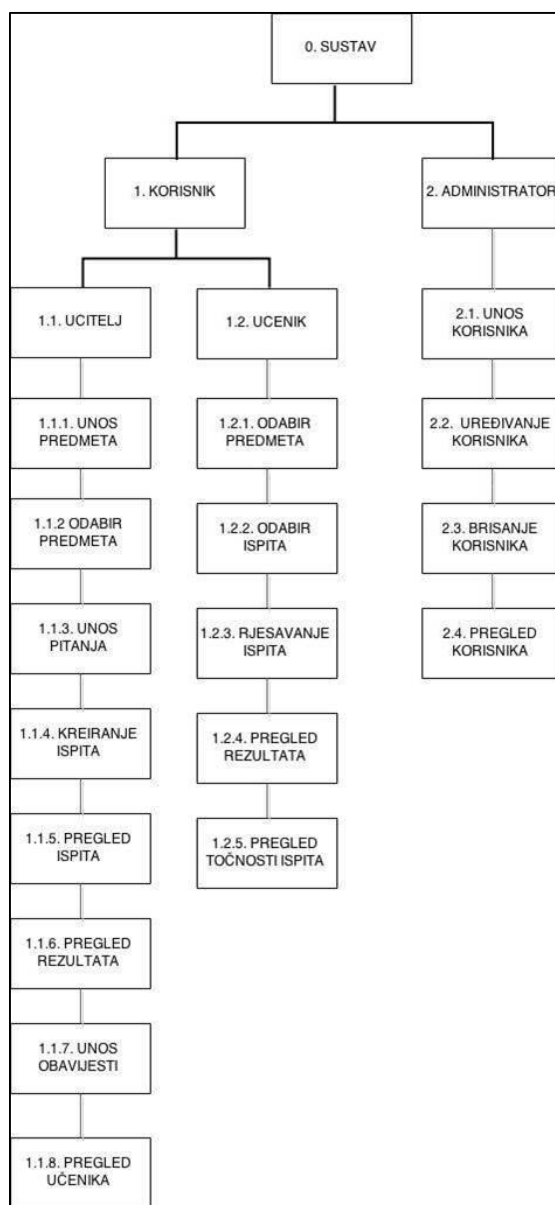
Slika 2 Dodavanje putanje za php.exe

Nakon odrađenih koraka za instalaciju i korištenje yii frameworka, moguće je putem komandne konzole kreirati novi projekt, aplikaciju. Potrebno je unesti putanju i lokaciju na kojoj se projekt bude kreirao i samim time kreirati početni projekt. Unutar daljnjeg projekta koristilo se programsko okruženje NetBeans. Instalacija samog okruženja nije drugačije od instalacije bilo kojeg drugog softvera. Međutim, nakon instalacije, potrebno je preuzeti nadogradnju putem koje se NetBeans spaja s yii frameworkom. Nakon spajanja tih dviju alata, moguće je započeti s daljnjim radom na projektu.

4. SPECIFIKACIJA SUSTAVA

4.1. DIJAGRAMI DEKOMPOZICIJE I BAZA PODATAKA

Daljnji dijagrami prikazuju funkcionalnosti sustava i sve procese i aktivnosti koji će se unutar sustava događati.



Slika 3 Dijagram toka podataka nulte razine

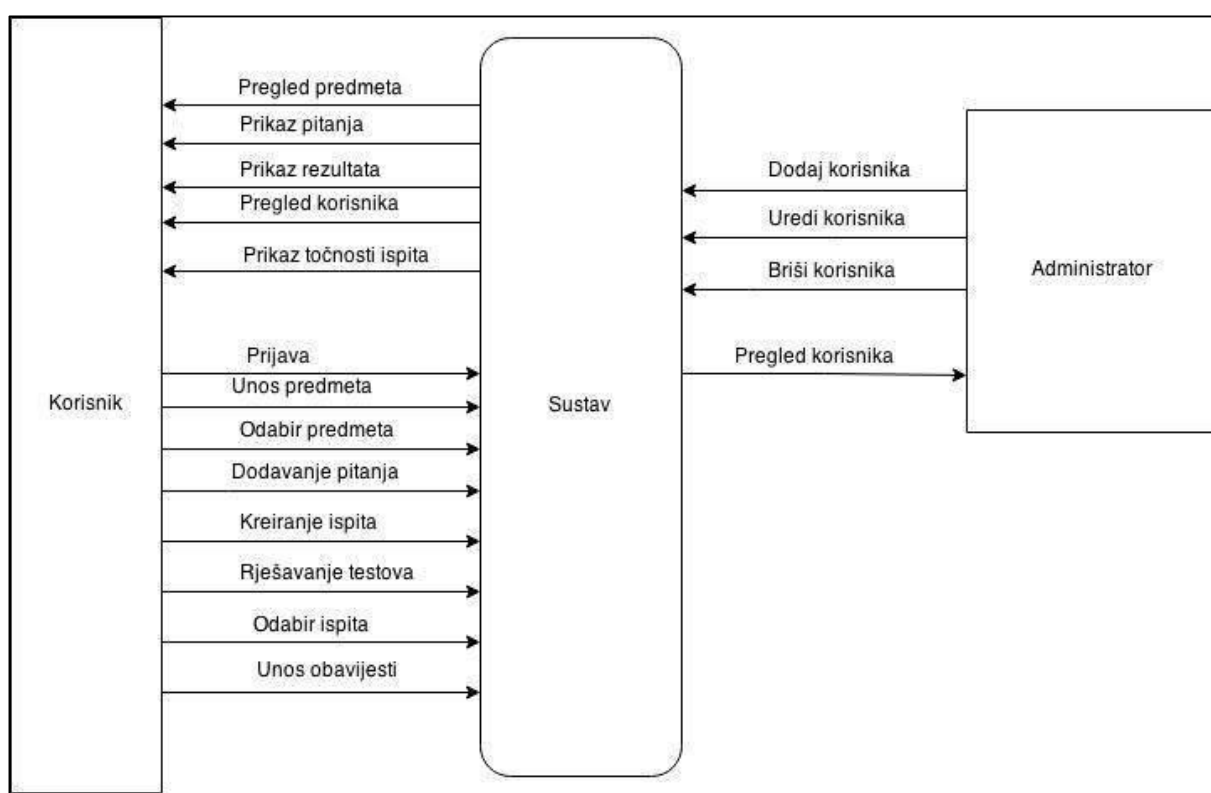
Unutar dijagrama toka podataka nulte razine prikazan je sustav i od kojih razina i pod razina se sam sustav sastoji. Sustav se sastoji od razina Korisnik i Administrator. Aktivnosti su prikazane u daljnjem postupku grananja dijagrama. Time je prikazano da Korisnik se dalje grana u dijelove sustava Učitelj i Učenik. Aktivnosti koje će moći obavljati učitelj su: unos

predmeta, unos pitanja, kreiranje ispita, pregled ispita, pregled rezultata, unos obavijesti i pregled svih učenika jednog predmeta.

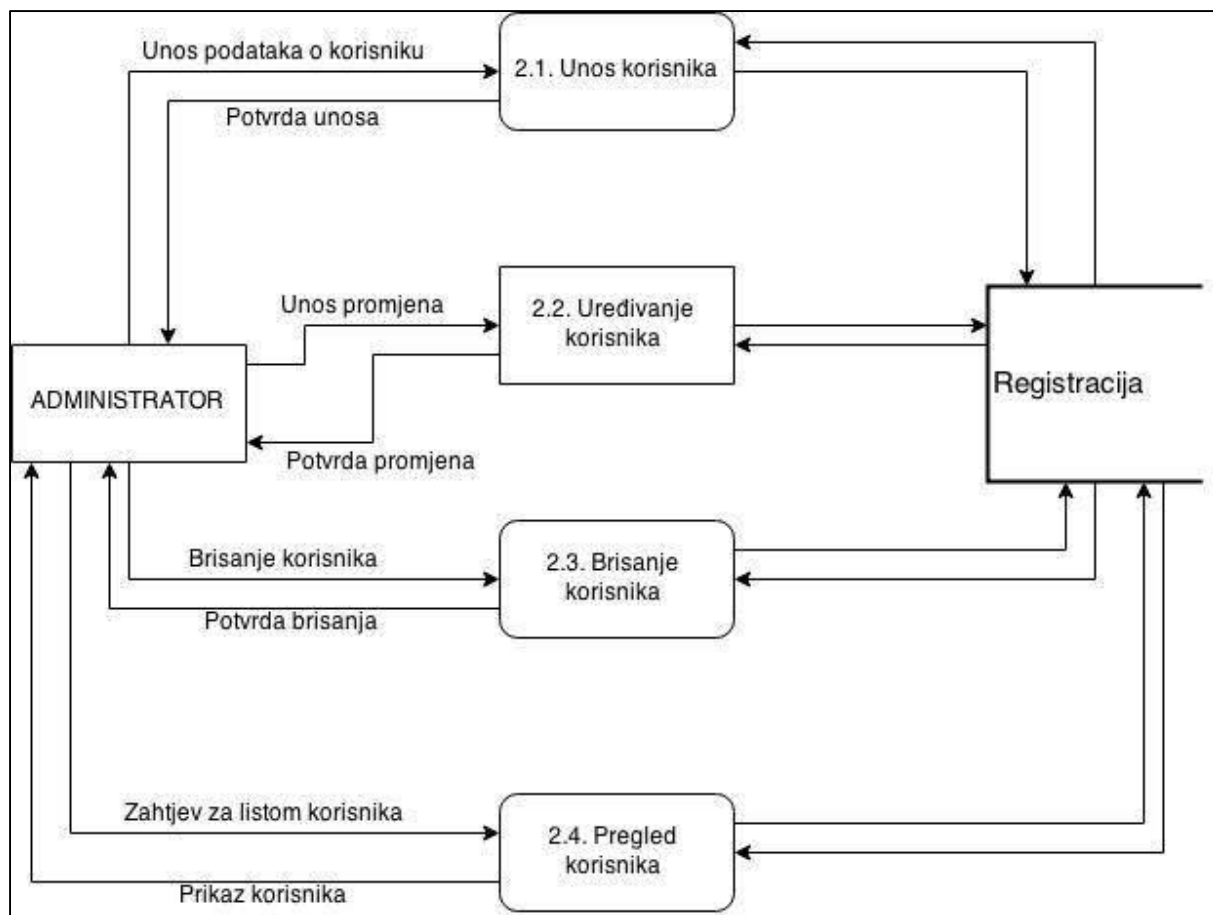
Aktivnosti koje će moći obavljati učenik su: odabir predmeta, odabir ispita, rješavanje ispita, pregled rezultata i pregled točnosti napisanog ispita.

Aktivnosti koje će moći obavljati administrator su: unos korisnika, izmjena korisnika, brisanje korisnika, pregled korisnika.

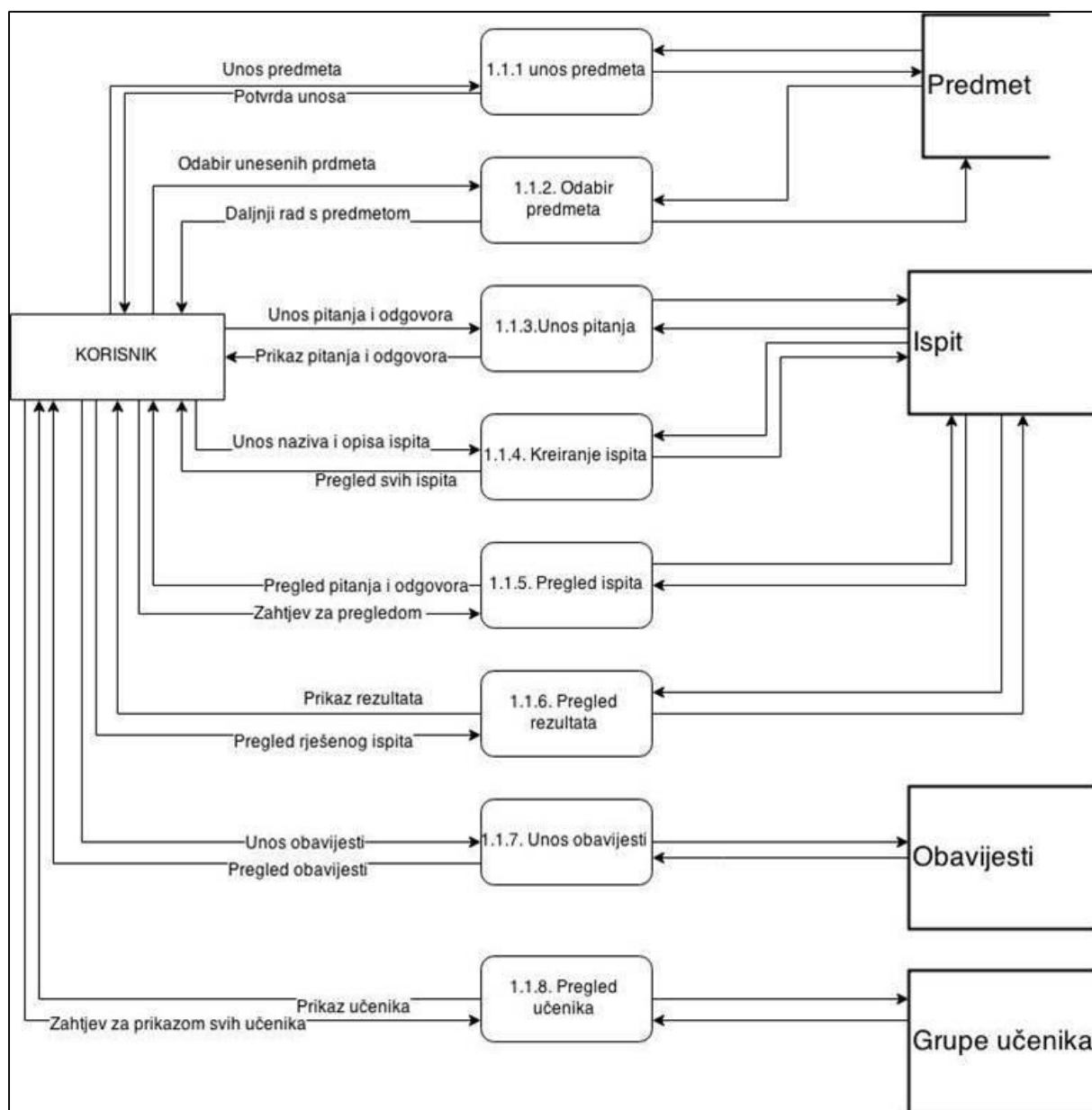
Daljnji dijagrami prikazuju na koji način i putem kojih aktivnosti će se događati procesi unutar sustava.



Slika 4 Dijagram aktivnosti



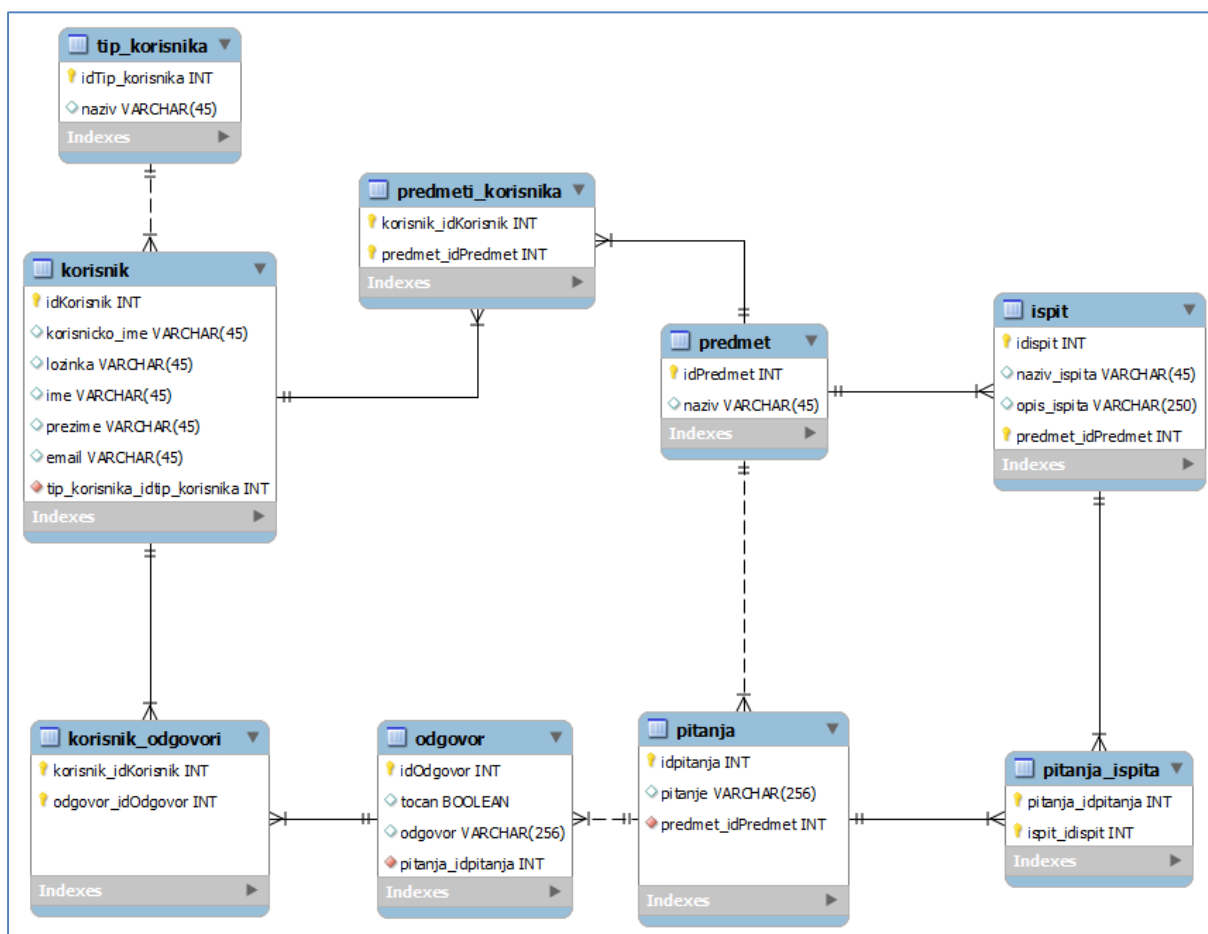
Slika 5 Dijagram aktivnosti – ADMINISTRATOR



Slika 6 Dijagram aktivnosti – KORISNIK

4.2. Početni relacijski dijagram baze podataka

Relacijski dijagram izgrađen je putem MySQL Workbench programa. Unutar njega definirani su entiteti, tablice koje će se koristiti i veze između njih. Definirani su i atributi koji će se nalaziti unutar tablica, te primarni i vanjski ključevi potrebni za održavanje integriteta baze podataka.



Slika 7 Relacijski model baze podataka

Kao što je prikazano na slici, baza podataka sastoji se od 9 tablica. Podatke potrebne za prijavu u sustav i djelovanje unutar sustava sadržavati će tablica korisnik. Nakon prijave u sustav, sljedeće podatke sadržavati će tablica predmet i složena tablica između korisnika i predmeta, predmeti_korisnik. Tablica predmet sadržavati će podatke o nazivu

predmeta te složena tablica predmeti_korisnik sadržavati će listu svih učitelja i njihovih pojedinih predmeta. Nakon što korisnik odabere predmet, moći će kreirati nove ispite ili nova pitanja potrebne za ispunjavanje ispita. Kako bi nakon kreiranja ispita, sam ispit mogao ispuniti pitanjima potrebno je bilo kreirati složenu tablicu pitanja_ispita koje se sastoji od svih pitanja pojedinih ispita. Nakon kreiranja ispita i pitanja, potrebno je i kreirati odgovore na ista pitanja. Nakon kreiranja odgovora, moguće je ispit rješavati od strane učenika i nakon ispunjavanja ispita, učenici bi bili u mogućnosti pregledati svoje odgovore i vidjeti rezultat koji su na ispitu ostvarili.

4.3. SQL upiti

Nakon kreiranja baze podataka, bilo je potrebno istu bazu podataka i testirati kako bi se provjerila ispravnost veza između tablica i provjerilo da li je sve kreirano. Kako bi se to moglo izvesti, bilo je potrebno ispisati SQL upite putem kojih smo kao povratnu informaciju dobili podatke koji su nam bili potrebni za rad sustava.

- Prikaz svih predmeta određenog korisnika testirano je upitom:

```
SELECT predmet.naziv FROM predmet
LEFT JOIN predmeti_korisnika
ON predmeti_korisnika.predmet_idPredmet=predmet.idPredmet
WHERE predmeti_korisnika.korisnik_idKorisnik=idKorisnik ;
idKorisnik - trenutno prijavljeni korisnik u sustav korisnik.
```

- Prikaz svih pitanja jednog ispita testirano je upitom:

```
SELECT pitanja.pitanje FROM pitanja
LEFT JOIN pitanja_ispita
ON pitanja.idpitanja=pitanja_ispita.pitanja_idpitanja
LEFT JOIN ispit
ON ispit.idispit=pitanja_ispita.ispit_idispit
WHERE pitanja.predmet_idPredmet=1 and ispit.idispit=1;
```

idPredmet i idispit označavaju odabrani ispit ili pitanja koja se odaberu putem korisničkog sučelja unutar sustava

- Prikaz određenih odgovora od određenog korisnika iz određenog predmeta za određeni ispit:

```
SELECT odgovor.odgovor, odgovor.tocan, pitanja.pitanje FROM korisnik_odgovori
```

```
LEFT JOIN odgovor
```

```
ON korisnik_odgovori.odgovor_idOdgovor=odgovor.idOdgovor
```

```
LEFT JOIN pitanja
```

```
ON odgovor.pitanja_idpitanja=pitanja.idpitanja
```

```
LEFT JOIN pitanja_ispita
```

```
ON pitanja.idpitanja=pitanja_ispita.pitanja_idpitanja
```

```
WHERE korisnik_odgovori.korisnik_idKorisnik=1 and pitanja.predmet_idPredmet=2  
and pitanja_ispita.ispit_idispit=2;
```

idKorisnik, idPredmet, idispit predstavljaju trenutno odabrane vrijednosti unutar sustava putem korisničkog sučelja.

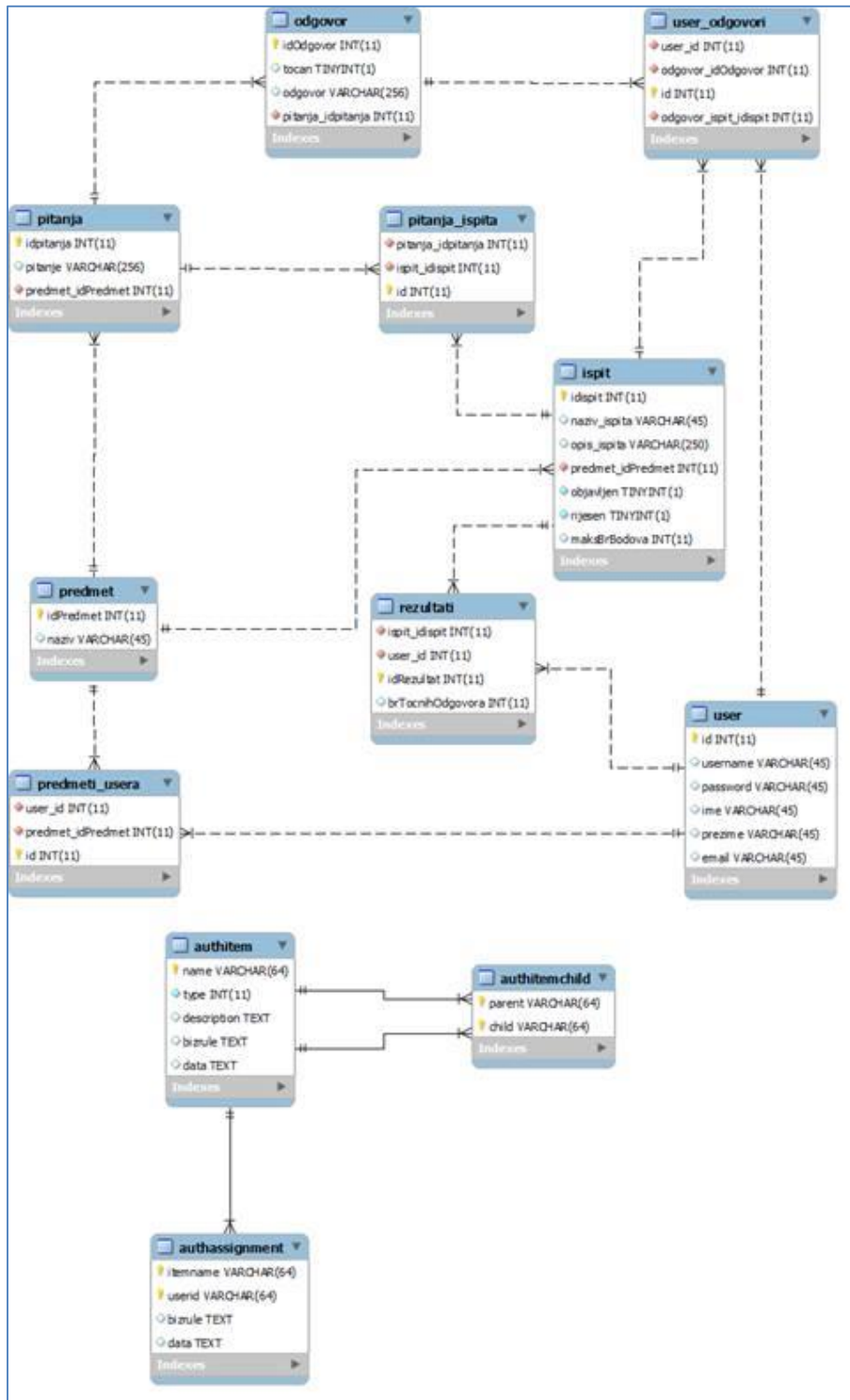
5. DIZAJN I IMPLEMENTACIJA

Prilikom početka rada na izgradnji sustava prvobitno je bilo potrebno odlučiti o samom izgledu sustava. Početne prozore i sučelje smo izradili u Mockflowu. Oni su nam pomogli prilikom izgradnje jer smo putem njih mogli odlučivati o svim funkcionalnostima koje su bile potrebne za sustav.

5.1. Aktivnosti dizajna

Nakon izgradnje relacijske baze podataka i generiranja svih osnovnih formi bilo je potrebno odlučiti na osnovu prethodnih mockupa o samom izgledu sustava i svih njegovih pojedinih dijelova. Uz izgled bilo je potrebno sve to prilagoditi i funkcionalnostima koje je potrební prozor nudio. Tokom implementacije bilo je potrebno i izmijeniti početnu relacijsku bazu zbog dodatnih funkcionalnosti koje je sustav zahtijevao. U daljnjoj projektnoj dokumentaciji biti će prikazani prozori sustava skupa s njihovim funkcionalnostima i pripadajućom izmijenjenom relacijskom bazom podataka.

5.2. Dizajn podataka, relacijska baza podataka



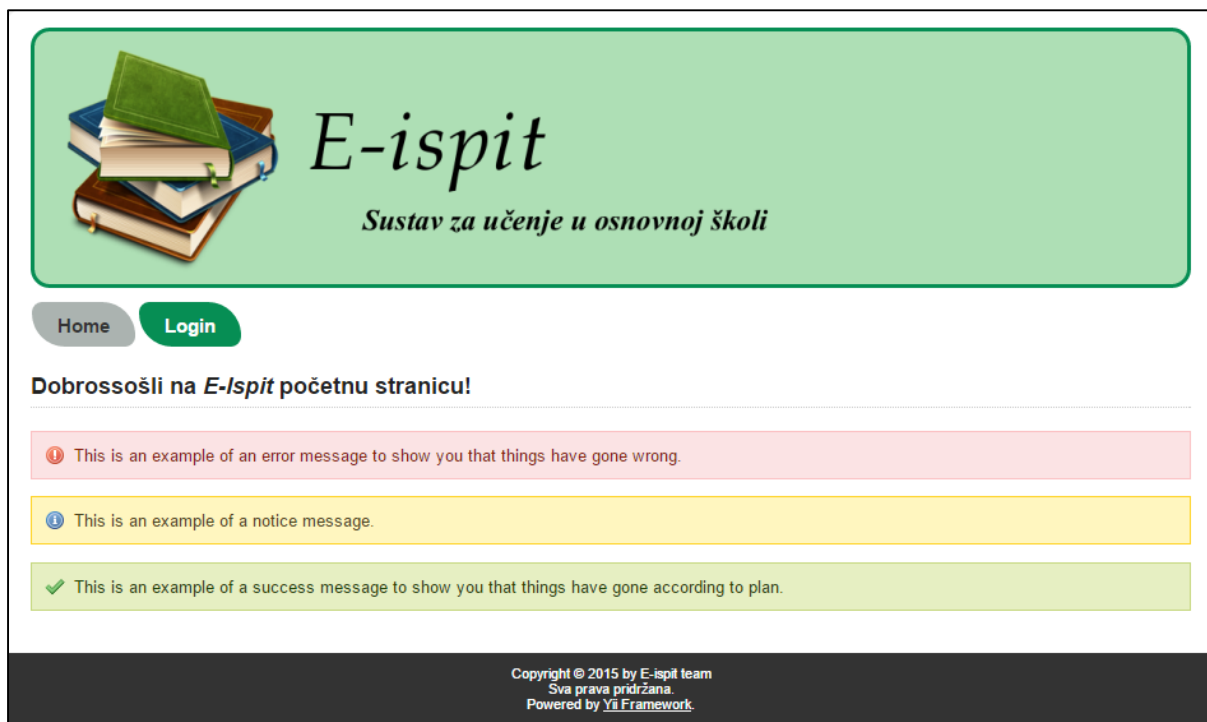
Kao što je prikazano na slici bilo je potrebno dodatno izgraditi nove relacije kako bi se pokrile sve funkcionalnosti koje je sustav zahtijevao. Prvobitna relacijska baza podataka nije sadržavala tablice pomoću kojih bi se pokrile funkcionalnosti vezane uz autentikaciju i autorizaciju. Njihovu izgradnju zahtijevao je Yii Framework kako bi mogao voditi podatke o korisničkim računima i njihovim ovlastima i rolama. Pomoću SQL koda, kojeg nam Yii Framework pruža kreirane su 3 tablice. Tablica *authitem* u kojoj se dodaju role i ovlasti. Zatim tablica *authitemchild* u kojoj se definira parent-child arhitektura rola i ovlasti. Na primjer admin je parent od profesor, profesor je parent od učenik. Posljednja tablica je *authassignment* u kojoj se dodjeljuje id iz postojeće tablice *user* koje se onda dodjeljuje rolama.

Tokom implementacije pojavila se potreba za novom tablicom koja bude skladištila sve podatke vezane uz riješeni ispit i korisniku koji je isti ispit riješio. Time je izgrađena nova tablica *rezultati*. Ona sadržava dva vanjska ključa na tablicu *user* i *ispit*. Sadrži i svoj primarni ključ i atribut unutar kojeg se skladišti broj točnih odgovora koje je korisnik tokom ispita ostvario.

Problematika tokom izgradnje pojavila nam se u tablici *user_odgovori* jer nije skladištila podatke o samom ispitu kojem je korisnik pristupio. Problem je riješen povezujući tablice *ispit* i *user_odgovori*. Vezom je tablica *user_odgovori* dobila još jedan vanjski ključ na tablicu *ispit*.

6. DIZAJN SUČELJA


Na slici je prikazan početni prozor aplikacije. Početna stranica se sastoji od izbornika Home i Login. Ovisno o tome koji se korisnik prijavi, tako mu se i prikazuje početni meni.



Slika 8. Početna stranica

Aplikacija ima tri tipa korisnika: admin, profesor i učenik.

U nastavku ćemo navesti koje sve funkcionalnosti može izvesti pojedini tip korisnika i koje funkcionalnosti pruža sama aplikacija.



E-ispit

Sustav za učenje u osnovnoj školi

[Home](#)
[Rijesi ispit](#)
[Predmet](#)
[Ispit](#)
[Pitanja ispita](#)
[Odgovori](#)
[Administracija](#)
[Logout \(admin\)](#)

[Home](#) » [Korisnici](#) » [Upravljaј korisnicima](#)

Upravljaј korisnicima

[Napredno pretraživanje](#)

1 - 3 of 3 cool records

ID	Korisničko ime	Lozinka	Ime	Prezime	E-mail	Actions
1	admin	sa1aY64JOY94w	Nikola	Rodeš	rodac@live.com	
2	profesor	saxiCMDQOTNwM	Valentina	Klepac	valentina@gmail.com	
3	ucenik	sa1BLgRRf/cmg	Jelena	Benić	jelena@gmail.com	

Odaberi

[Popis korisnika](#)

[Dodaj korisnika](#)

Copyright © 2015 by E-ispit team
Sva prava pridržana.
Powered by [Yii Framework](#)

Slika 9.

Administracija

Administrator ima pravo upravljati korisnicima, što znači da može vidjeti popis svih korisnika i može dodati novog korisnika. Administrator ima ovlasti da dodaje novog korisnika te odlučuje o tipu korisnika kojeg dodaje. Klikom na popis korisnika otvara nam se novi izbornik gdje možemo pregledati sve učenike i profesore zasebno. Osim toga administrator kreira pojedini predmet korisniku.

Predmet

Ovdje se vide svi predmeti koji su do sada kreirani i kreiraju se novi predmet . Kada kreiramo novi predmet za njega možemo kreirati ispit i kreirati pitanja za ispit. Kreiranja pitanja za ispit se sastoji od labele u koju upisujemo pitanja i ona se spremaju. Kasnije se na temelju pitanja kreiraju odgovori. Ovdje je samo potrebno kreirati pitanja za ispit.

Predmet

Displaying 1-2 of 2 results.

Informatika

Kreiraj ispit Kreiraj pitanja

Matematika

Kreiraj ispit Kreiraj pitanja

Slika 10. Kreiranje ispita i kreiranje predmeta

Ispit

Nakon što se kreira ispit potrebno je kreirati ispit za određeni predmet . Potrebno je unijeti naziv ispita (npr. Informatika 1) te opis ispita (npr . prvi test iz informatike). Ukoliko se to radi o predmetu informatika potrebno ga je odabrati, na slici je prikazan padajuću izbornik gdje se nalaze svi predmeti koji su do sada kreirani.

Kreiraj ispit

*Fields with * are required.*

Naziv Ispita

Opis Ispita

Predmet Id Predmet *

Informatika ▼

Objavljen *

Rijesen *

Create

Slika 11. Forma za kreiranje ispita

Pitanja ispita

Kreiranje ispita se sastoji od dva padajuća izbornika gdje moramo izabrati ispit i pitanja koja pripadaju tom ispitu.

Odgovori

Kreiranje odgovora se radi tako da se odabere pitanje i upisuje se odgovor, kad je odgovor točan označimo Tocan. Kad želimo upisati sljedeći odgovor, opet odaberemo isto pitanje i upisujemo odgovor.



The screenshot shows a web form titled "Kreiraj odgovor". Below the title is a note: "Fields with * are required." The form contains the following elements: a checkbox labeled "Tocan (ako je odgovor tocan stavi kvacicu, ako nije neka bude prazno)", a text input field labeled "Odgovor", a dropdown menu labeled "Pitanje:" with the selected option "Matična ploča nalazi se u:", and a "Create" button at the bottom.

Slika 12. Forma za kreiranje odgovora

Riješi ispit

Prije nego što učenik počinje rješavati ispit, administrator mu to mora omogućiti tako da u administraciji kreira predmet usera. I nakon toga učenik može pristupiti ispitu

7. Faza testiranja

Aplikaciju smo postepeno testirali kako smo ju izgrađivali. Prilikom testiranja naišli smo na neke bugove. Na samom početku izrade aplikacije problem nam je predstavljala sama dokumentacija yii frameworka jer je dokumentacija zahtijevala od korisnika da poznaje sve što je u dokumentaciji ispred traženog pojma.

Najveći problem nam je bio povezivanje tablica u svrhu unosa i ispisa iz tablice putem sql upita JOIN. Taj problem smo riješili proučavanjem sintakse koja je bila vezana uz to.

Još jedan bug je bio da kod složene tablice nije radio razlomljeni search (riječ ili fraza mora biti doslovno ista kao i u bazi) na primjer kad upišemo prva tri slova od riječi koju želimo pretražiti izbacivalo bi da ta riječ ne postoji.

Yii ne može generirati modele i crud operacije nad tablicama sa složenim ključevima.

8. Evolucija

Ovaj sustav za provjeru znanja prvobitno je bio zamišljen za osnovne škole. Taj sustav planiramo unaprijediti i poboljšati. Jedno od poboljšanja bi bilo da se navigacija u samom sustavu smanji, da se kreiranje ispita, odgovora i pitanja stavi sve na jedno mjesto. Osim toga imamo u planu uvesti esejska pitanja. Kada bi profesor ispravljao ta esejska pitanja sa strane pokraj svakog pitanja bi se pojavio točan odgovor kako bi to profesoru olakšalo posao. Postupak kreiranja samog ispita se namjerava poboljšati, na način da se izvrši u nekoliko koraka.