

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Dario Horvat

Tea Jarčov

Anabel Li Kečkeš

Matija Popijač

In4maticsQuiz

PROJEKTNI RAD IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA

Varaždin, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Dario Horvat, 44497/15-R

Smjer: Organizacija poslovnih sustava

Tea Jarčov 44442/15-R

Smjer: Organizacija poslovnih sustava

Anabel Li Kečkeš 44448/15-R

Smjer: Informacijsko i programsko inženjerstvo

Matija Popijač, 44441/15-R

Smjer: Informacijsko i programsko inženjerstvo

Diplomski studij

Link na repozitorij: <https://github.com/mpopijac/In4maticsQuiz>

In4maticsQuiz

PROJEKTNI RAD IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA

TEHNIČKA DOKUMENTACIJA

Mentor:

Doc. dr. sc. Zlatko Stapić

Varaždin, 2015.

Sadržaj

1.	Uvod.....	1
1.1.	Svrha.....	1
1.2.	Definicije, akronomi, skraćenice.....	1
1.3.	Reference.....	1
1.4.	Pregled.....	1
2.	Korisnički zahtjevi	2
3.	Specifikacija funkcionalnih zahtjeva	4
4.	Sveukupni opis	5
4.1.	Perspektiva proizvoda	5
4.2.	Funkcije proizvoda.....	5
4.3.	Korištene tehnologije	5
4.4.	Ograničenja	6
4.5.	Pretpostavke i zavisnosti	6
4.6.	Testni podaci.....	6
5.	Specifični zahtjevi	7
5.1.	Proizvodni zahtjevi.....	7
5.2.	Logički zahtjevi baze podataka	7
5.3.	Obilježja atributa aplikacije	7
5.1.1.	Pouzdanost	7
5.1.2.	Dostupnost.....	7
5.1.3.	Sigurnost.....	7
5.1.4.	Održavanje	8
6.	Dijagram slučajeva korištenja	9
7.	Arhitektura sustava.....	11

8.	Dijagram klasa.....	12
9.	ERA model.....	16
9.1.	Era model web servisa.....	16
9.2.	Era model baze mobilne aplikacije	17
10.	Korisničko sučelje	18
10.1.	Mockup.....	18
10.2.	Stvarni izgled sučelja	19
11.	Specifikacija web servisa	27

1. Uvod

Aplikacija će biti pomoć u učenju učenicima osnovnih škola. Kroz samu aplikaciju učenici će moći provjeravati svoje znanje iz informatike. Sama aplikacija nudit će izbor razreda za koje učenici žele odgovarati na pitanja. U aplikaciji će biti vidljivi rezultati korisnika po svojim korisničkim imenima, tj. postojat će rang lista sa najboljim korisnicima. Rang lista služila bi da se korisnici mogu uspoređivati sa drugima, tj. da ih lošiji plasman od nekog drugog potakne na učenje da postignu bolji rezultat.

1.1. Svrha

Svrha aplikacije je pomoći učitelju u dobivanju povratnih informacija od učenika, a isto tako i učiniti nastavu i učenje informatike malo drugačijim. Povratne informacije bi se dobivale u stvarnom vremenu, tj. odmah na nastavi i to preko mobilnih telefona, što nije uobičajeno provođenje provjere na nastavi. Pošto aplikacija ima i rang liste, učenici bi se međusobno mogli natjecati i tako bi željeli postati što bolji.

1.2. Definicije, akronimi, skraćenice

Korisnik- dvije vrste korisnika: učenik ili učitelj koji može biti administrator

Pitanja- pitanja koja se rješavaju u aplikaciji

Odgovori- pomoću točnih odgovora na pitanja se slažu rang liste

Rang lista- lista korisnika od najboljeg prema najgorem po najboljem rezultatu

API- Application Programming Interface

1.3. Reference

- IEEE Std 830-1998, Software Requirements Specifications
- IEEE Std 1016-2009, Software Design Descriptions

1.4. Pregled

U nastavku će biti objašnjene mogućnosti i neki primjeri korištenja.

2. Korisnički zahtjevi

Da bi bili u mogućnosti aplikaciju kreirati sukladno potrebama korisnika, najvažnije je jasno i detaljno odrediti što korisnik aplikacije želi da mu bude omogućeno pri korištenju aplikacije. Definiranje korisničkih zahtjeva u većini metodika razvoja predstavlja jedan od temeljnih koraka u razvoju programskog proizvoda pa tako i u Scrum metodologiji razvoja. Bitno je upoznati se sa svim zahtjevima korisnika kako bi se mogli kreirati zadaci pomoću kojih se realiziraju funkcionalnosti programskog rješenja. Definiranjem korisničkih zahtjeva opisujemo što će aplikacija raditi, način na koji će raditi te kako će to pomoći krajnjim korisnicima u njihovom radu.

Korisnički zahtjevi za In4maticsQuiz mobilnu aplikaciju nastali su na temelju razgovora s učiteljem informatike i želje da uvede neke promijene u nastavu. Mobilna aplikacija bi učeniku trebala omogućiti pregled pitanja za gradivo koje se obradilo na nastavi i tako mu učinilo ponavljanje zanimljivijim. Isto tako bi aplikacija služila da učitelj ima neke povratne informacije od učenika jer bi tu aplikaciju mogao iskoristiti na satu i vidjeti koliko je učenika shvatilo gradivo jer bi on, kao administrator, mogao kreirati, ažurirati i brisati već postojeća pitanja.

Tablica 1. Popis korisničkih zahtjeva

Zahtjev ID	Naziv	Opis zahtjeva
KZ 1	Prijava u sustav	Učenik se treba prijaviti u sustav kako bi se mogao nalaziti na rang listama s odabranim nadimkom i kako bi se mogao potvrditi njegov identitet prema imenu i prezimenu. Prilikom prijave učitelja prepoznaje se da je on administrator i otvaraju mu se drugačije mogućnosti od onih što su ponuđene učeniku.
KZ 2	Odabir razreda	Učeniku se nudi mogućnost prijave za koji razred želi odgovarati na pitanja.

KZ 3	Odgovaranje na pitanja	Korisniku se postavljaju pitanja na koja on može odabrati jedan odgovor, više točnih odgovora ili upisati odgovor riječima.
KZ 4	Pregled rang lista	Korisnici mogu pregledati gdje se nalaze na listi na temelju svojih odgovora. Rang liste koje se brišu nakon svakog dodavanja novih pitanja.
KZ 5	Dodavanje novih pitanja	Administratoru je omogućeno dodavanje novih pitanja u aplikaciju
KZ 6	Brisanje pitanja	Administratoru je omogućeno brisanje već postojećih pitanja u aplikaciji.
KZ 7	Ažuriranje pitanja	Administratoru je omogućeno ažuriranje već postojećih pitanja u aplikaciji.

3. Specifikacija funkcionalnih zahtjeva

Osnovna funkcionalnost ove aplikacije je rješavanje zadataka. Nakon što učenik riješi zadatak biti će smješten na rang listu ovisno o točnosti i brzini rješavanja zadataka. Na rang listi će se naći sa svojim imenom koje je upisao prilikom registracije, a kasnije prilikom prijave. Učenici će birati za koji razred žele odgovarati na pitanja i rang liste će biti posebne za svaki razred. Administrator, to jest profesor, će moći kreirati nova pitanja, ažurirati ih i brisati već postojeća pitanja.

Aplikacija In4maticsQuiz ima tri modula. Moduli se razlikuju u načinu odgovora na postavljena pitanja, to jest na različite tipove pitanja:

- 1) pitanje u kojem učenik odgovara upisom točnog pojma
- 2) pitanje u kojem učenik odgovara odabirom jedne od ponuđenih opcija
- 3) pitanje u kojem učenik označuje jedan ili više točnih odgovora

Svaki od tih modula je fragment. U aplikaciji unutar app modula mapa fragments sadrži ta tri modula (`TocnoNetocnoFragment.java`, `UnesiTocanPojamFragment.java` i `VisePonudenihOdgovoraFragment.java`). Sva tri fragmenta će se pozivati unutar aktivnosti koja se zove `ProvjeriZnanjeActivity.java`. Aktivnost `ProvjeriZnanjeActivity.java` ima layout pod nazivom `activity_provjeri_znanje` unutar kojeg je rezervirano mjesto za sva tri fragmenta. Pomoću metode `prikaziFragment` na rezervirano mjesto naše aktivnosti `ProvjeriZnanjeActivity.java` stavljamo fragmente. Metoda `prikaziFragment` prima parametar koji je tipa `Fragment`. U metodi `vrstePitanja`, koja prima parametar koji je tip `Pitanja`, pozivamo metodu `prikaziFragment`:

```
if (odgovoriTrenutno.size()==2) prikaziFragment(new TocnoNetocnoFragment());
else if
    (odgovoriTrenutno.size()>2)prikaziFragment(new VisePonudenihOdgovoraFragment());
else if
    (odgovoriTrenutno.size()==1) prikaziFragment(new UnesiTocanPojamFragment());
```

Fragmenti se slučajnim odabirom prikazuju na aktivnosti `ProvjeriZnanjeActivity.java`

4. Sveukupni opis

4.1. Perspektiva proizvoda

Aplikacija In4maticsQuiz je mobilna aplikacija koja je namijenjena korisnicima Android operacijskog sustava.

4.2. Funkcije proizvoda

Glavne funkcije aplikacije su:

- Prijava u aplikaciju
 - prijava sa korisničkim identitetom
- Registracija korisnika
 - Unos podataka(ime, prezime, korisničko ime, lozinka, email)
- Rješavanje kviza
 - Rješavanje kviza tako što se odabere željeni razred te se odgovara na ponuđena pitanja
- Pregled rang lista
 - Mogućnost pregleda svojih rezultata te usporedba sa drugim korisnicima
- Mogućnost ažuriranja pitanja
 - Pitanja može ažurirati korisnik koji je administrator

4.3. Korištene tehnologije

Za razvoj aplikacije korištene su različite tehnologije i alati(za razvoj, projektiranje i sl.)

Alati za modeliranje:

- Visual paradigm
- POP 2.0 - Prototyping on Paper

Sustav kontrole verzioniranja:

- Github repozitorij na kojem se nalazi sav kod i materijali vezani za projekt dostupni su na githubu.

Alati za razvoj:

- Android Studio

Web servisi i baza podataka

MySqlWorkbench CE 6.0

servis hostinger.hr (mysql, upravljanje preko phpmyadmina)

4.4. Ograničenja

Ograničenja su moguća prilikom pristupa na bazu podataka, tj. na dostupnost baze podataka, te uz minimalne razine Android API-ja.

4.5. Pretpostavke i zavisnosti

Da bi mogli imati vezu sa bazom pretpostavka je da imamo stalnu internetsku povezanost za pristup serveru gdje se nalazi baza.

4.6. Testni podaci

Kako bi mogli testirati aplikaciju testni korisnički podaci za učitelja(Administradora) su:

korisničko ime: ucitelj

lozinka je: ucitelj.

5. Specifični zahtjevi

5.1. Proizvodni zahtjevi

Proizvodni zahtjevi vezani su uz to da minimalna verzija Android API-ja na uređajima korisnika mora biti Android API 16 .

5.2. Logički zahtjevi baze podataka

Baza podataka je definirana ERA dijagramom te su podaci normalizirani.

5.3. Obilježja atributa aplikacije

U nastavku su opisane značajke aplikacije.

5.1.1. Pouzdanost

Aplikacija bi nakon isporuke trebala raditi bez greške, jer će se na njoj provoditi testiranja koja će pokazati postoje li nedostaci, koji će tom prilikom biti uklonjeni ili implementirani na način da zadovolje zahtjevima korisnika.

5.1.2. Dostupnost

Aplikacija je uvijek dostupna nakon isporuke. Mogući su rijetki slučajevi nedostupnosti zbog održavanja baze podataka.

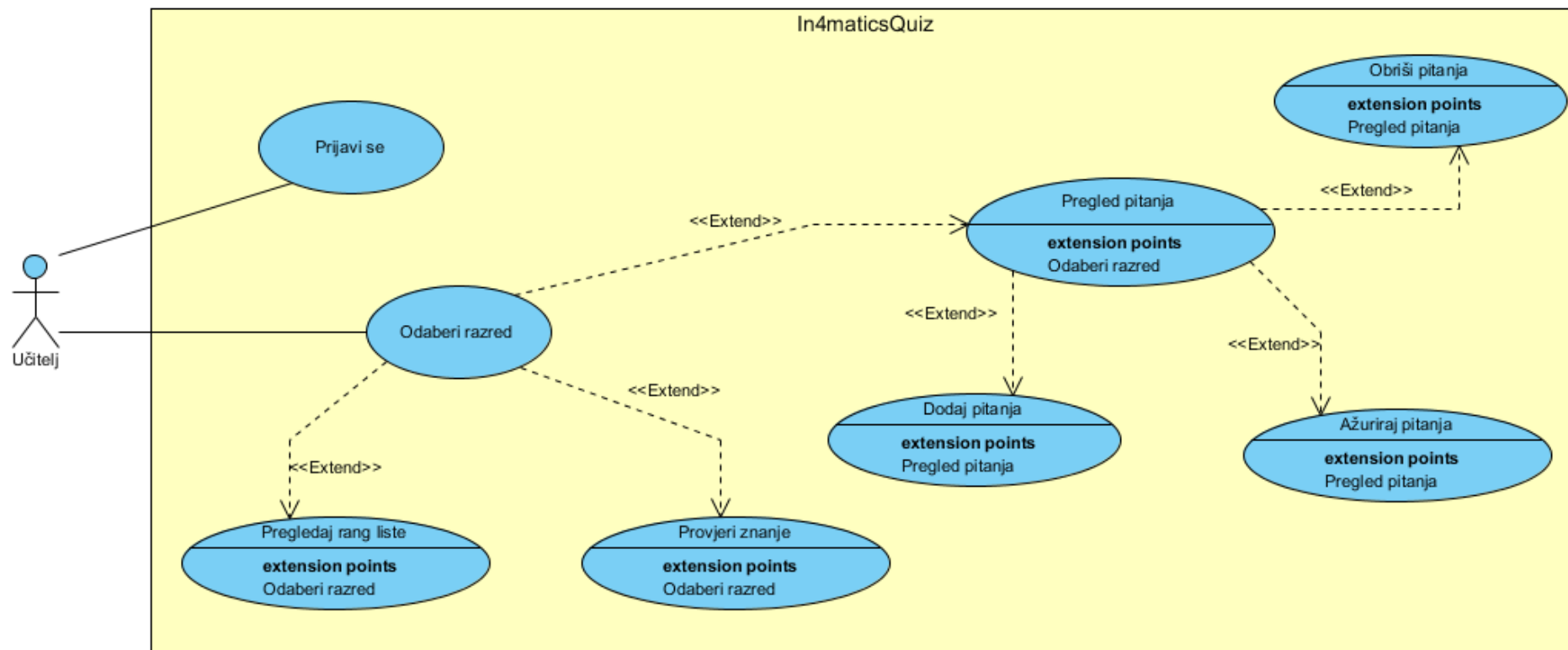
5.1.3. Sigurnost

Sigurnost može biti ugrožena ukoliko se ukrade identitet administratora te su moguće manipulacije sa pitanjima vezanim uz kviz.

5.1.4. Održavanje

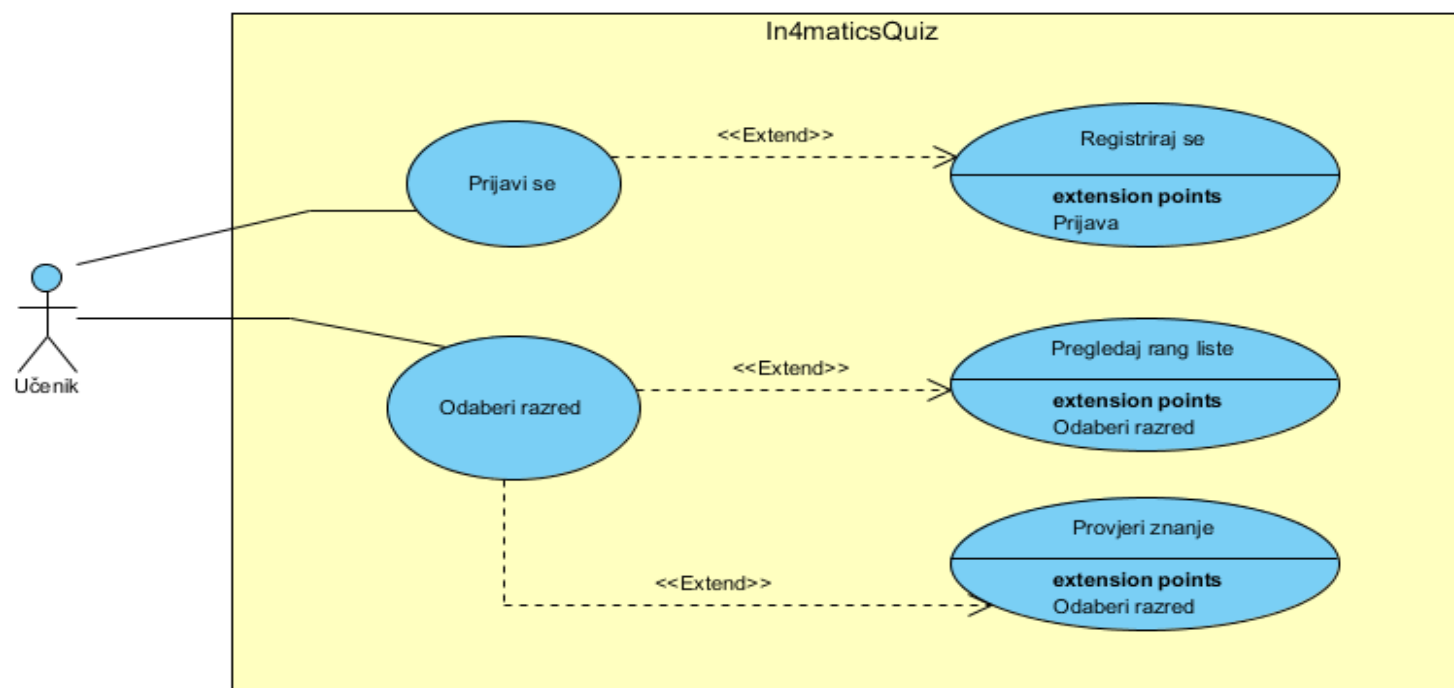
Održavanje aplikacije vrši se instalacijom novih verzija od strane korisnika.
Održavanje baze podataka vrši se automatski.

6. Dijagram slučajeja korištenja



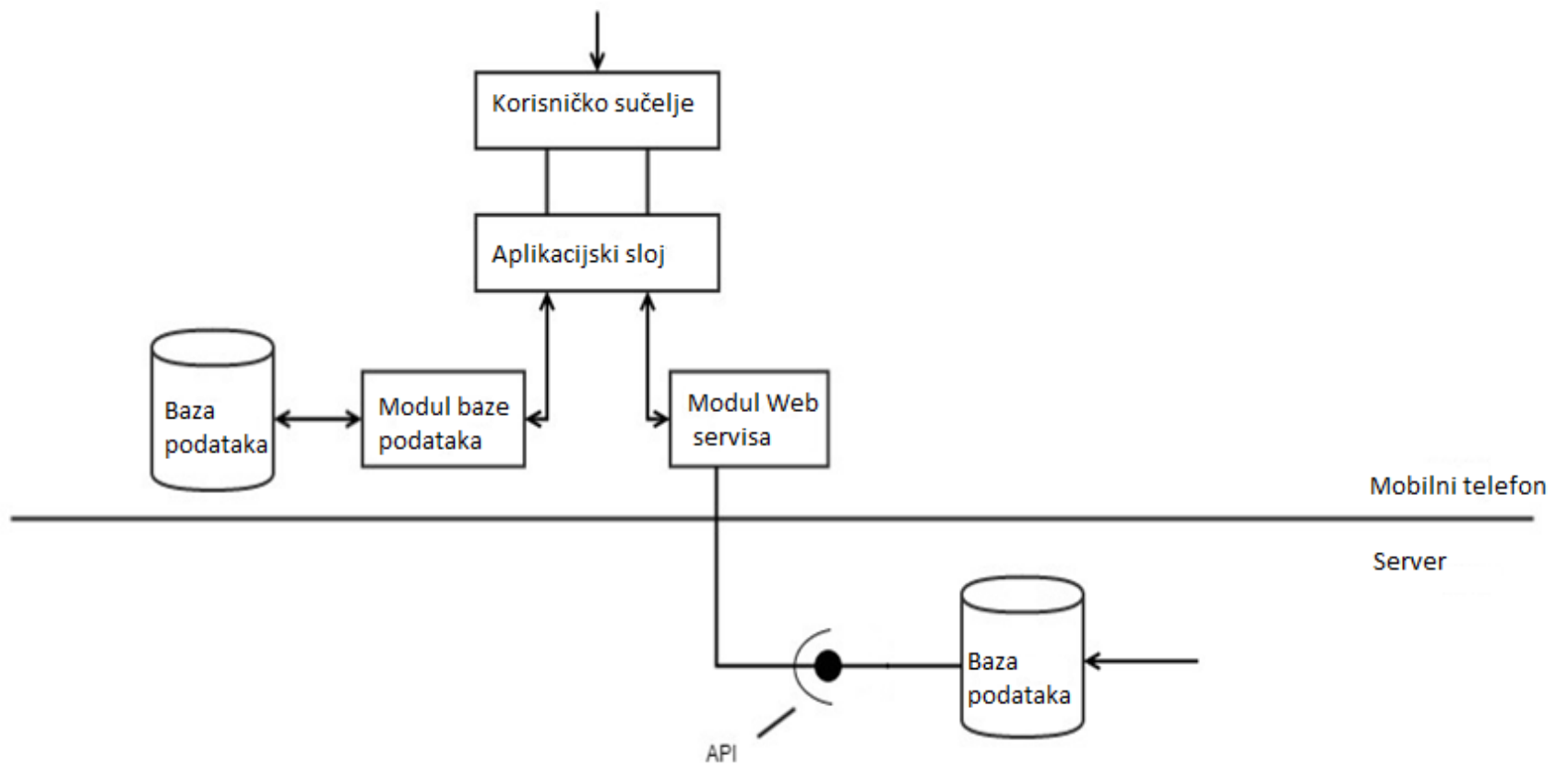
Slika 11. Dijagram slučajeja korištenja učitelja

Korisnik aplikacije može biti učenik ili učitelj. Postoje dvije razine ovlasti pa zato postoje različite mogućnosti. Dakle, korisnik se prvo mora prijaviti u sustav. Ako nije prijavljen, mora se registrirati da bi imao pristup (Slika 12). Zatim nastavlja na odabir razreda. Nakon toga ima opciju izabrati želi li rješavati zadatke i tako provjeriti znanje ili samo želi pogledati trenutne rang liste. Ako korisnik ima administratorske ovlasti (u našem slučaju to je učitelj) nakon odabira razreda može dodavati, ažurirati i brisati pitanja, ali isto tako i pregledati rang liste i riješiti test znanja (Slika 11) Učitelj se ne mora registrirati jer za njega već postoje korisnički podaci u bazi.



Slika 12. Dijagram slučajeva korištenja učenika

7. Arhitektura sustava

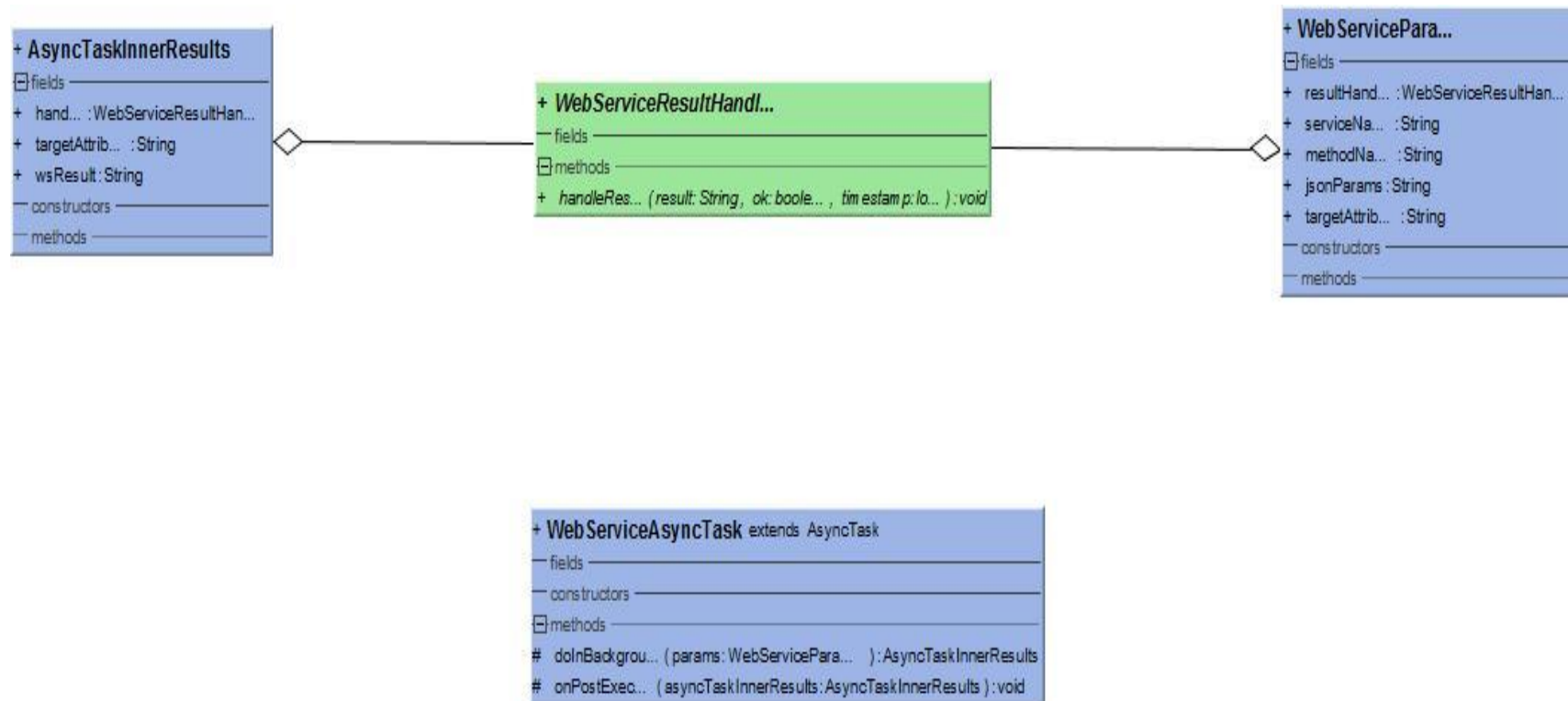


Slika 13. Arhitektura sustava

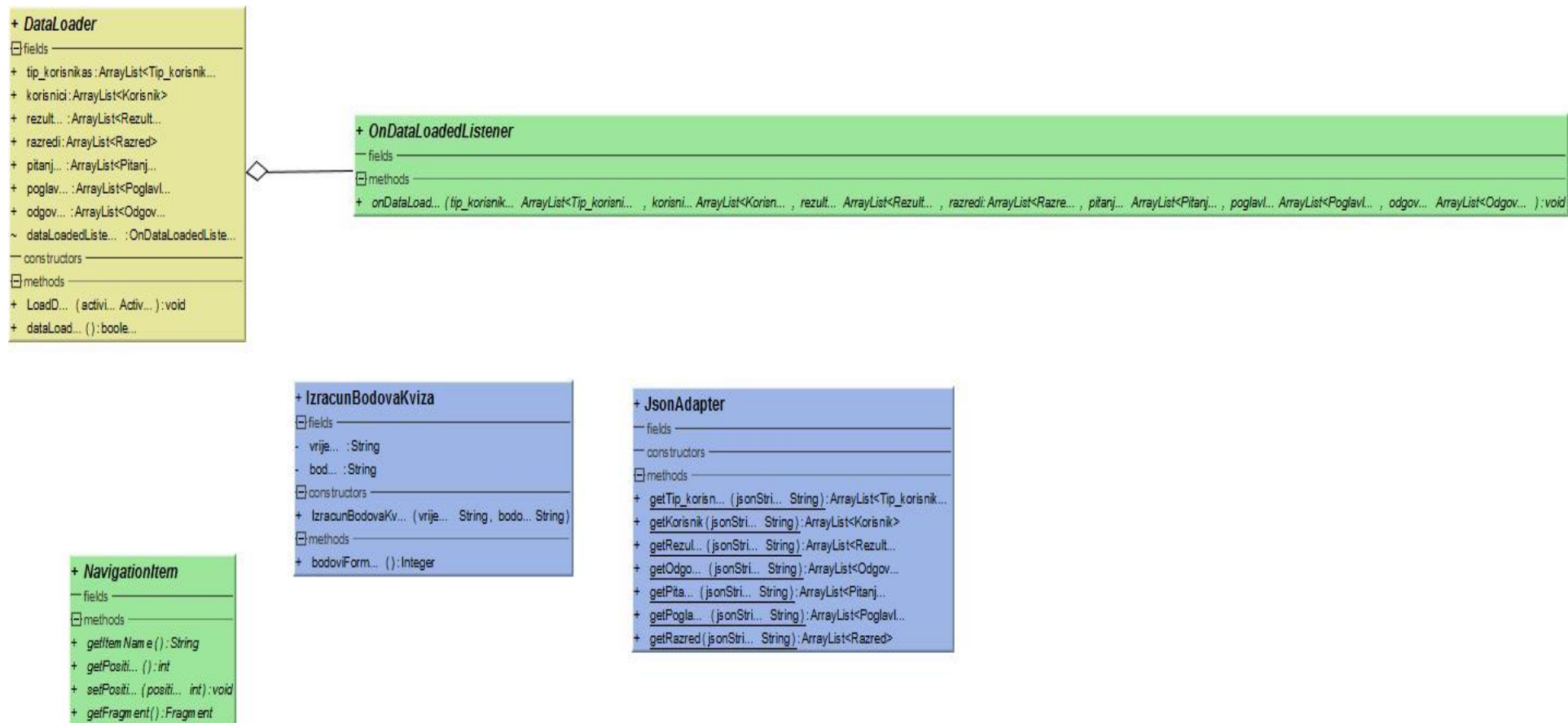
8. Dijagram klasa



Slika 14. Dijagram klasa baze podataka



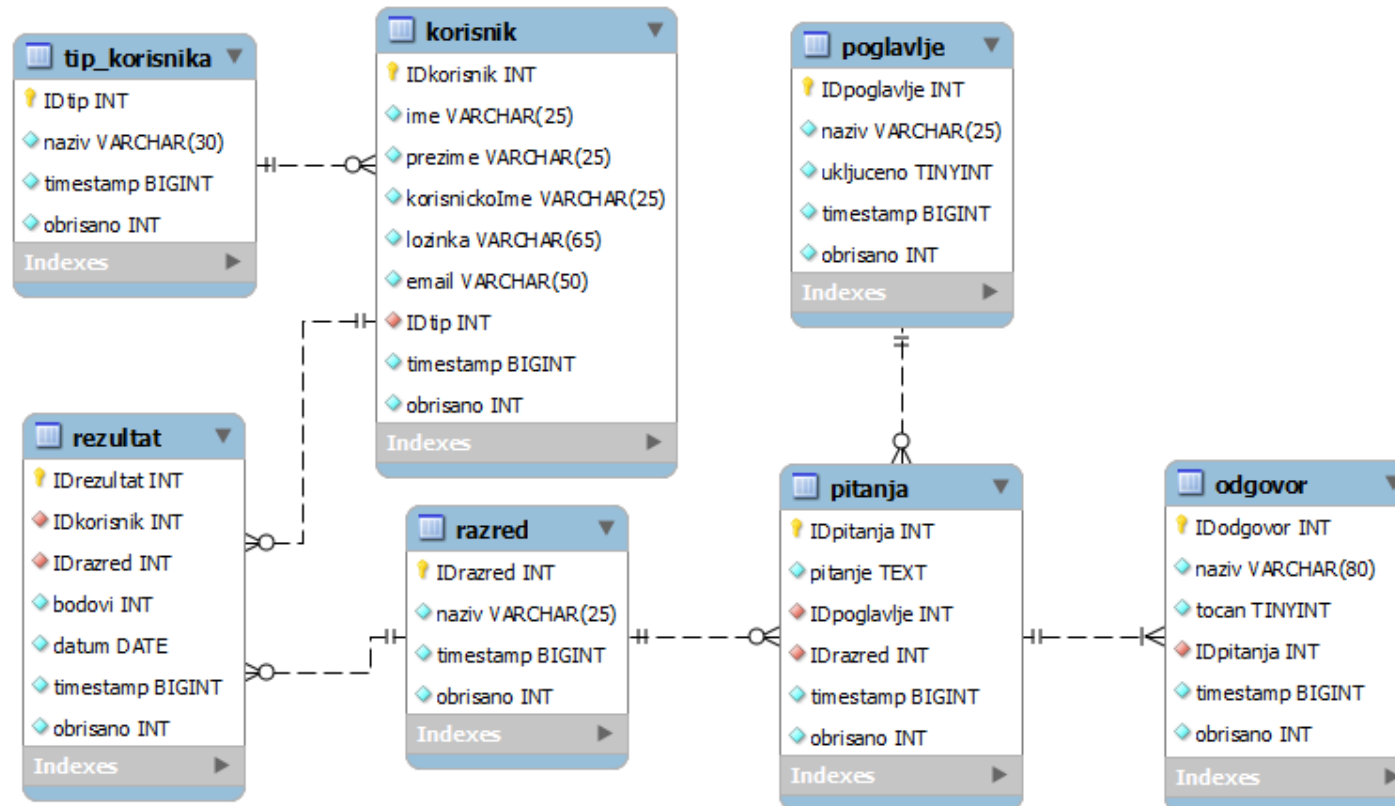
Slika 16. Dijagram klasa web servisa



Slika 17. Dijagram klase core modula

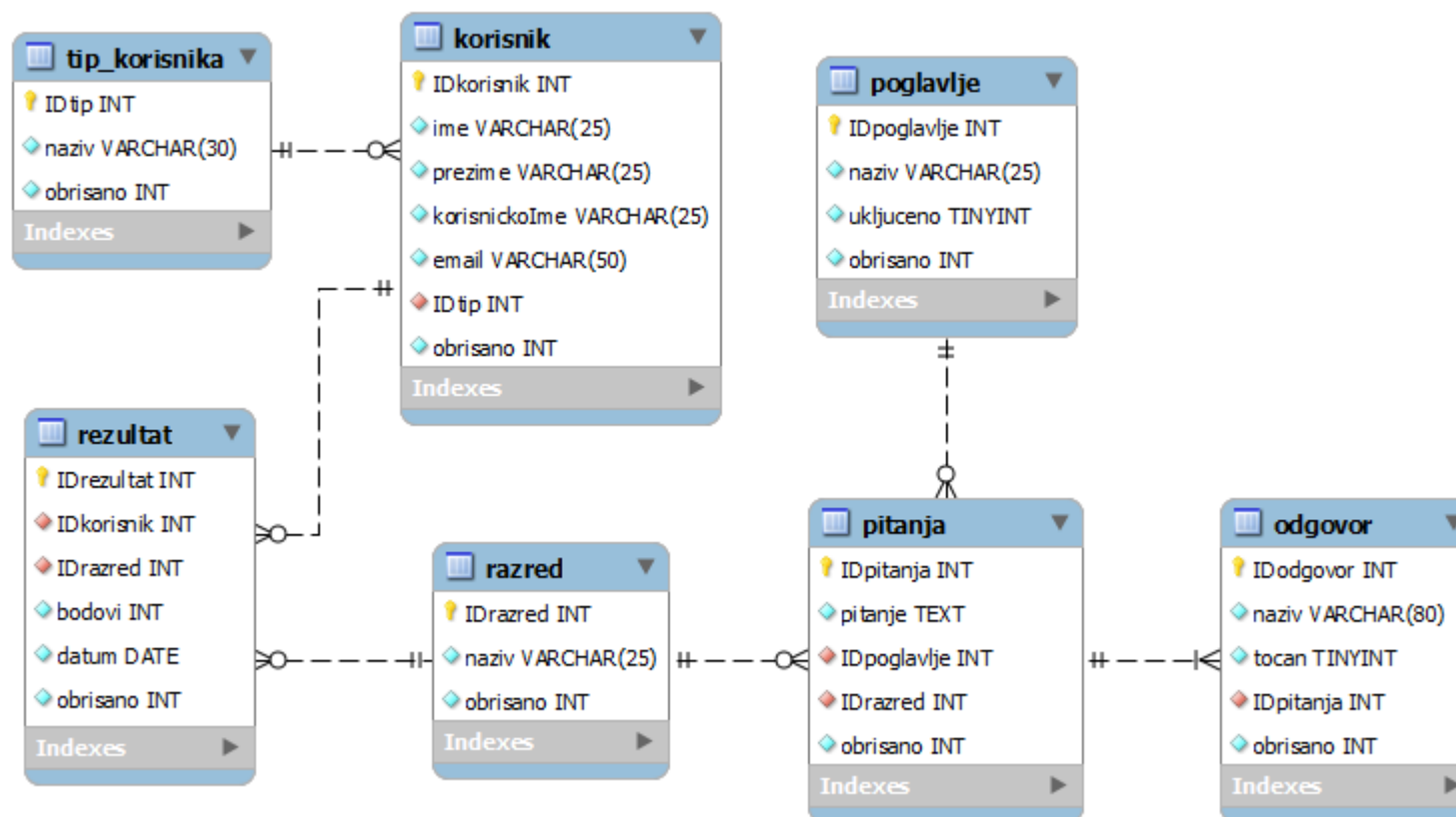
9. ERA model

9.1. Era model web servisa



Slika 18. ERA model

9.2. Era model baze mobilne aplikacije



Slika 19. ERA model baze mobilne aplikacije

10. Korisničko sučelje

10.1. Mockup

Korisničko sučelje smo izradili u aplikaciji POP 2.0 - Prototyping on Paper. To je aplikacija koja omogućava pretvoriti crteže u prototip aplikacije. Na prvom linku se nalazi aplikacija iz pogleda učenika, a na drugom linku aplikacija iz pogleda administratora. Napravili smo dva pogleda zbog ovlasti koje ima administrator. U opciji preview vidimo kako je zamišljeno da se korisnik kreće kroz aplikaciju. U nastavku su slike iz aplikacije.

- 1) <https://popapp.in/w/projects/561fa31238a672d235969674/mockups/56238a2412e879c057a628e4?from-preview>
- 2) <https://popapp.in/w/projects/5631fdeac6a612ef18abca7e/mockups/5631fe7cf95531504127b8f8>



Mockup-6
#1



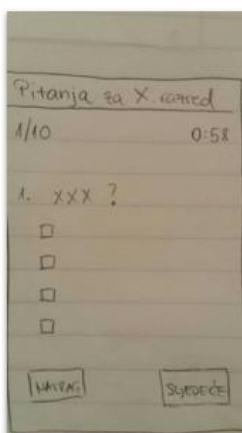
Mockup
#2



Mockup-4
#3



Mockup-5
#4



Mockup-3
#5

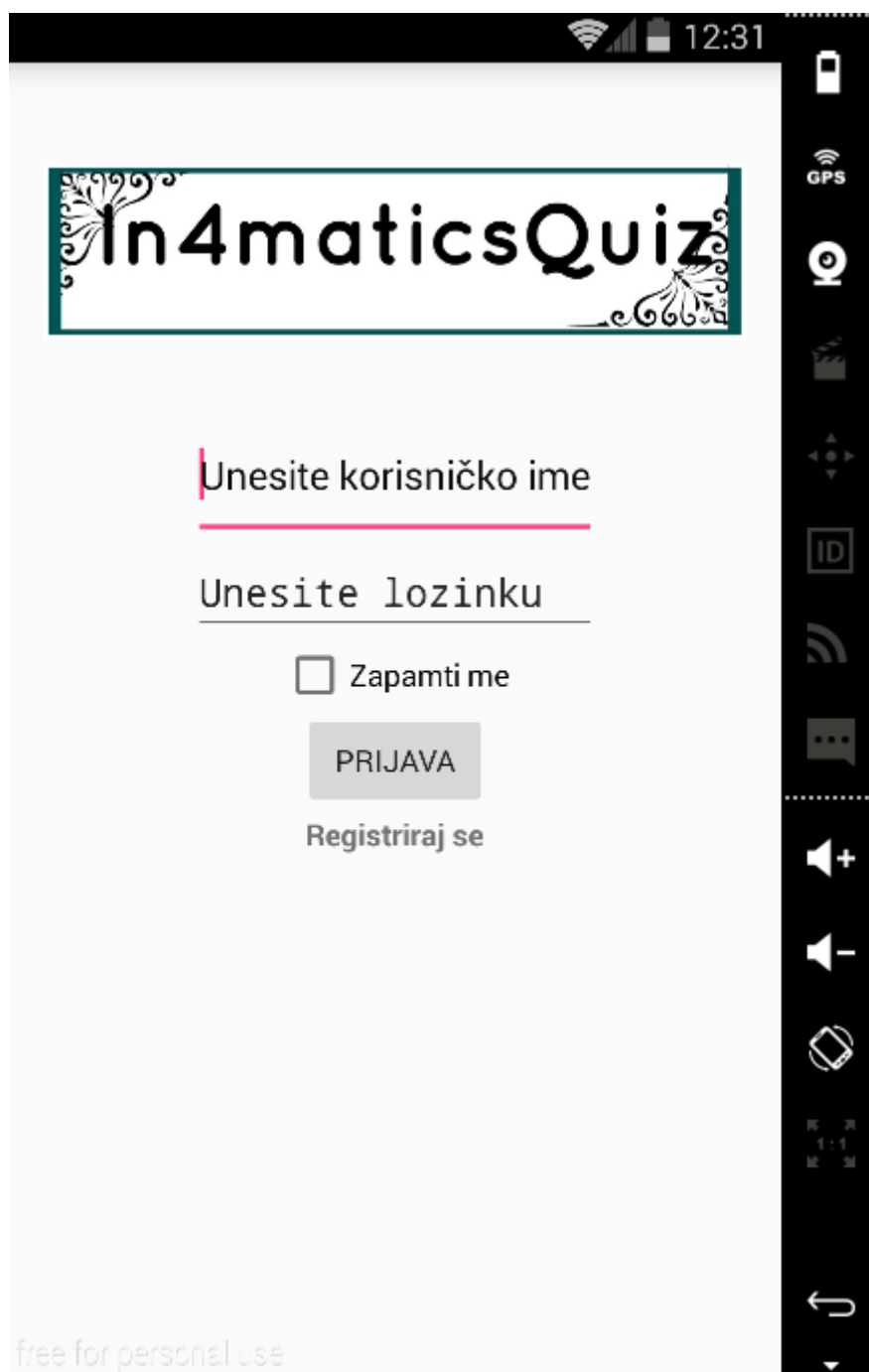


Mockup-2
#6

Slika 20. Korisničko sučelje

10.2. Stvarni izgled sučelja

Prilikom otvaranja aplikacije, otvara nam se sučelje na *Slici 21*. Ako smo već registrirani idemo direktno na prijavu, međutim ako nismo, moramo se prvo registrirati. Sučelje za registraciju je prikazno na *Slici 22*.



Slika 21. Prijava u aplikaciju

Prilikom registracije korisnik upisuje svoje ime, prezime i e-mail. Upisuje i korisničko ime i lozinku s kojima će se prijaviti u aplikaciju, a odabrano korisničko ime će biti prikazano na rang listi nakon rješavanja kviza.

Registracija

Ime
Unesite vaše ime

Prezime
Unesite vaše prezime

Korisničko ime
Unesite vaše korisničko ime

E-mail
Unesite vaš e-mail

Lozinka
Unesite vašu lozinku

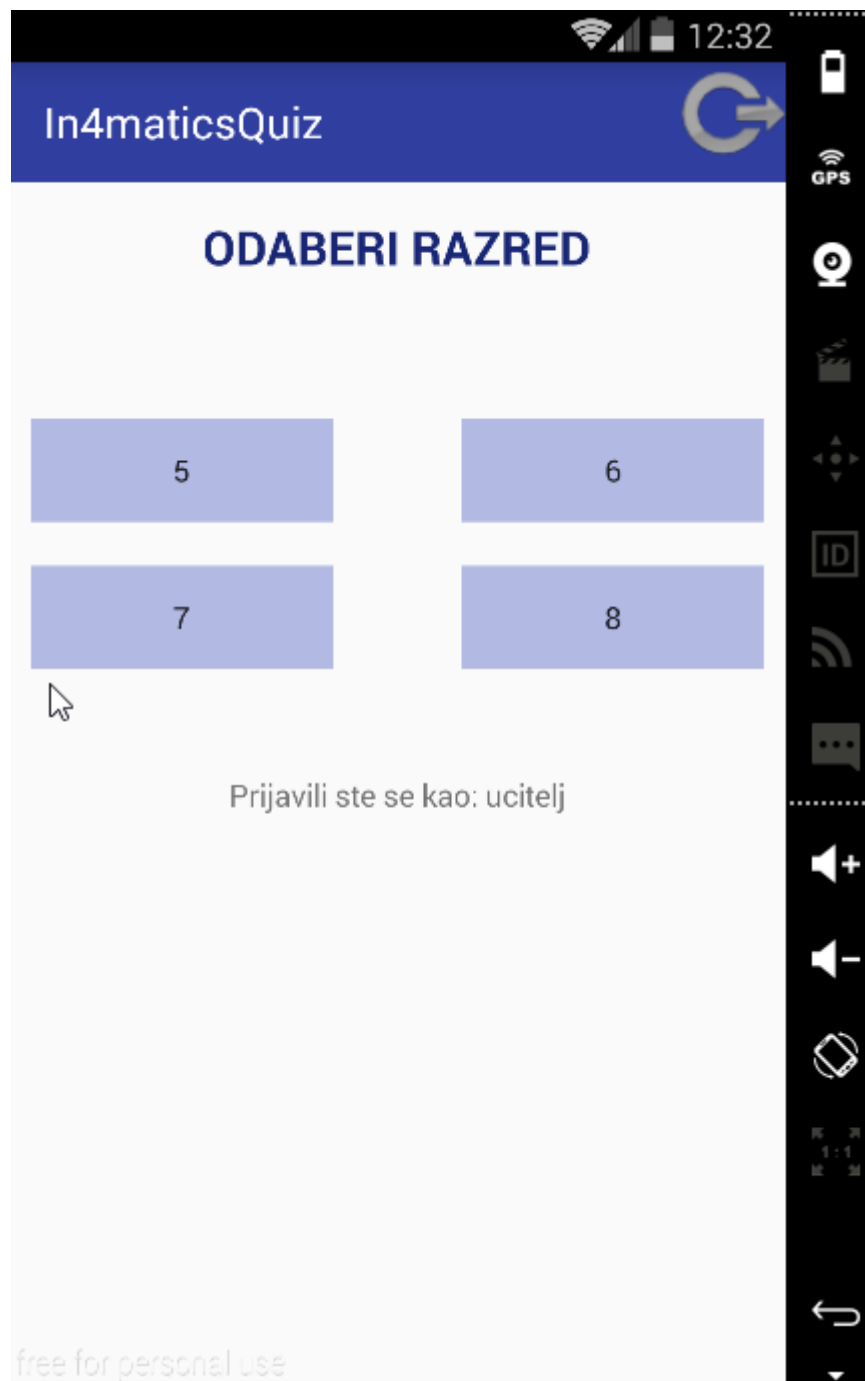
Potvrda lozinke
Potvrda lozinke

REGISTRIRAJ SE

free for personal use

Slika 22. Registracija korisnika

Nakon prijave, korisnik može odabrati za koji razred želi rješavati kviz.



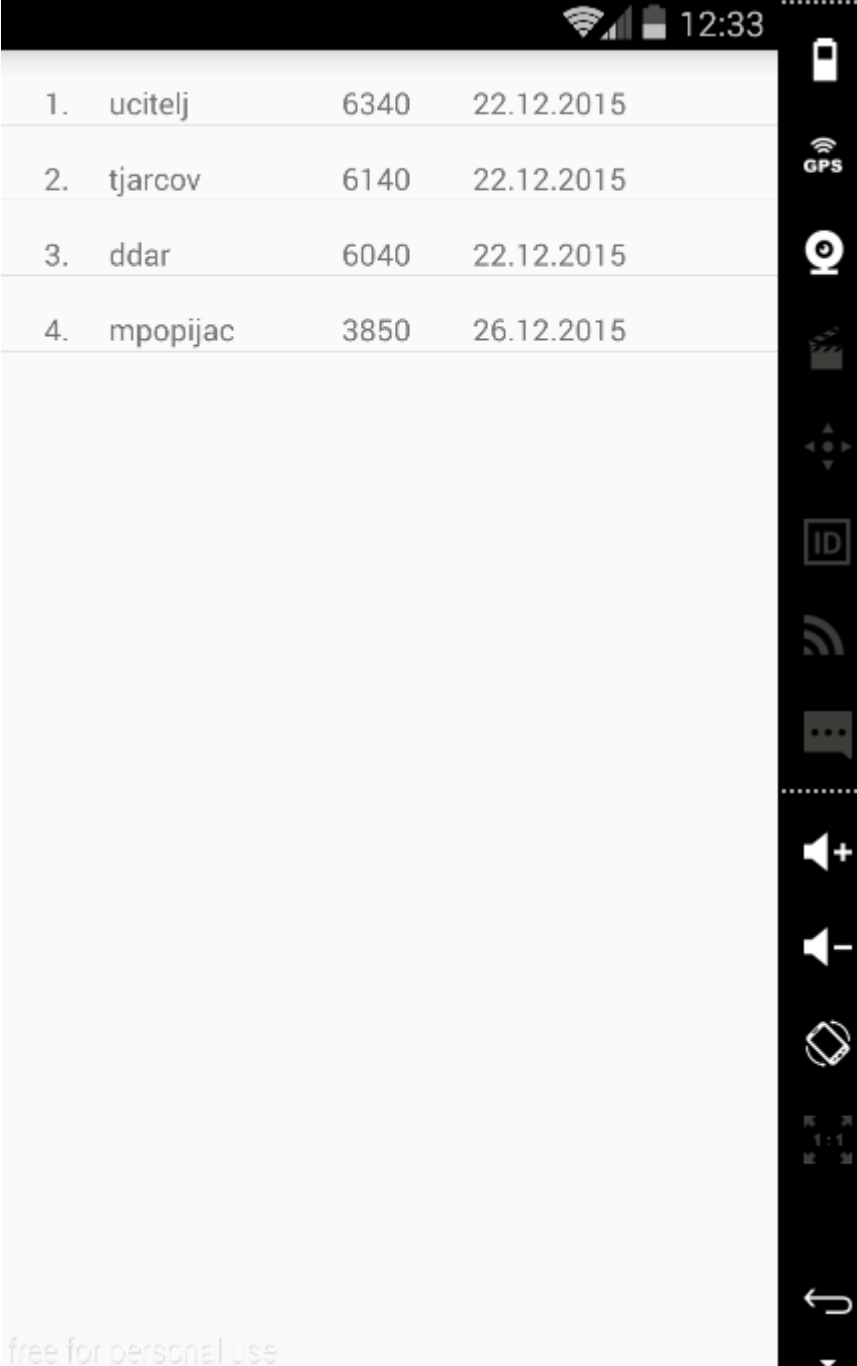
Slika 23. Odabir razreda

Nakon odabira razreda, prikazuje se sučelje gdje korisnik može odabrati želi li rješavati kviz ili želi pregledati rang liste.



Slika 24. Odabir pregleda rang listi ili provjere znanja

Ako je korisnik odabrao pregled rang listi, prikazuje mu se stanje trenutnih bodova za odabrani razred.

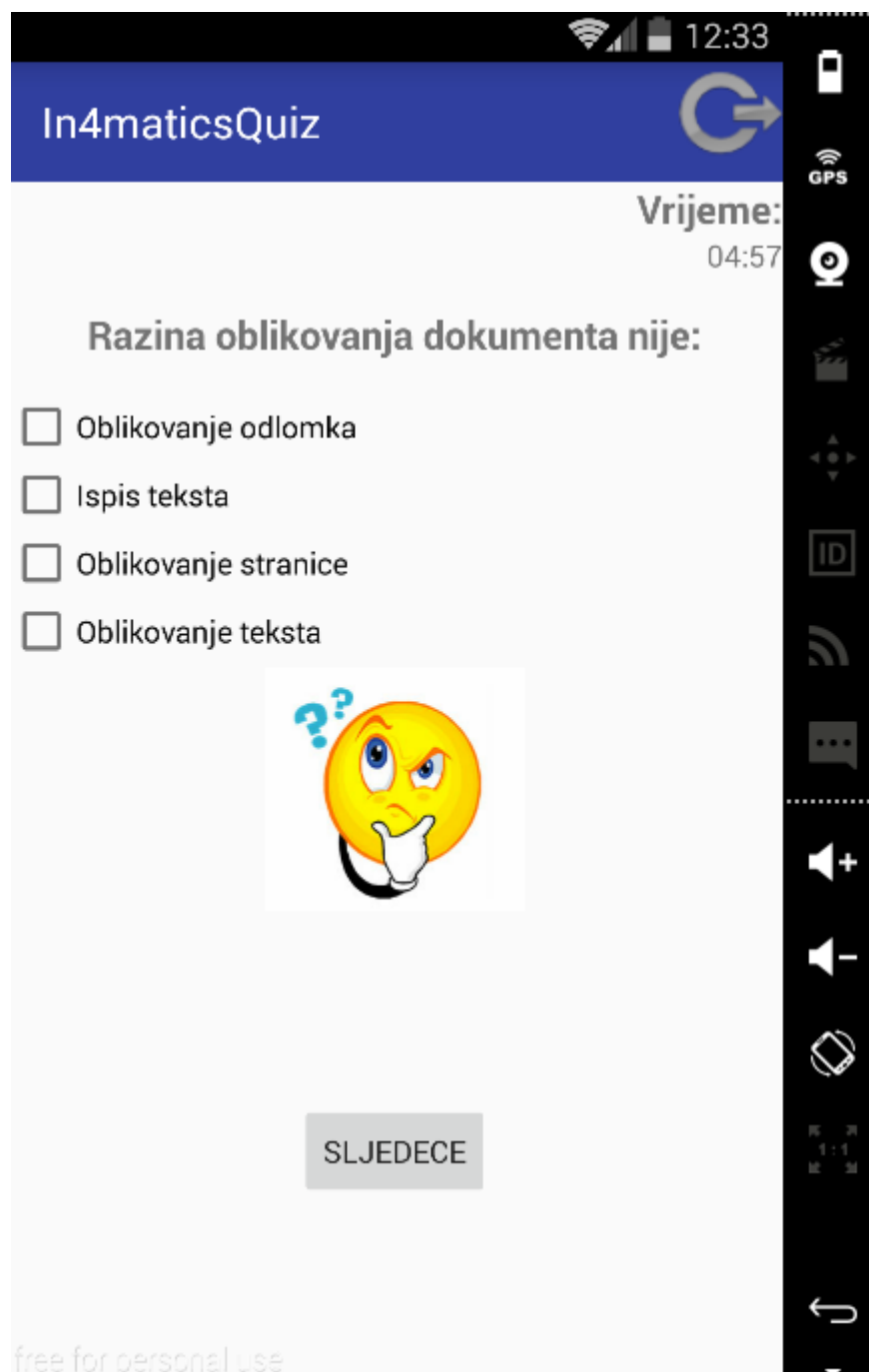


1.	ucitelj	6340	22.12.2015
2.	tjarcov	6140	22.12.2015
3.	ddar	6040	22.12.2015
4.	mpopijac	3850	26.12.2015

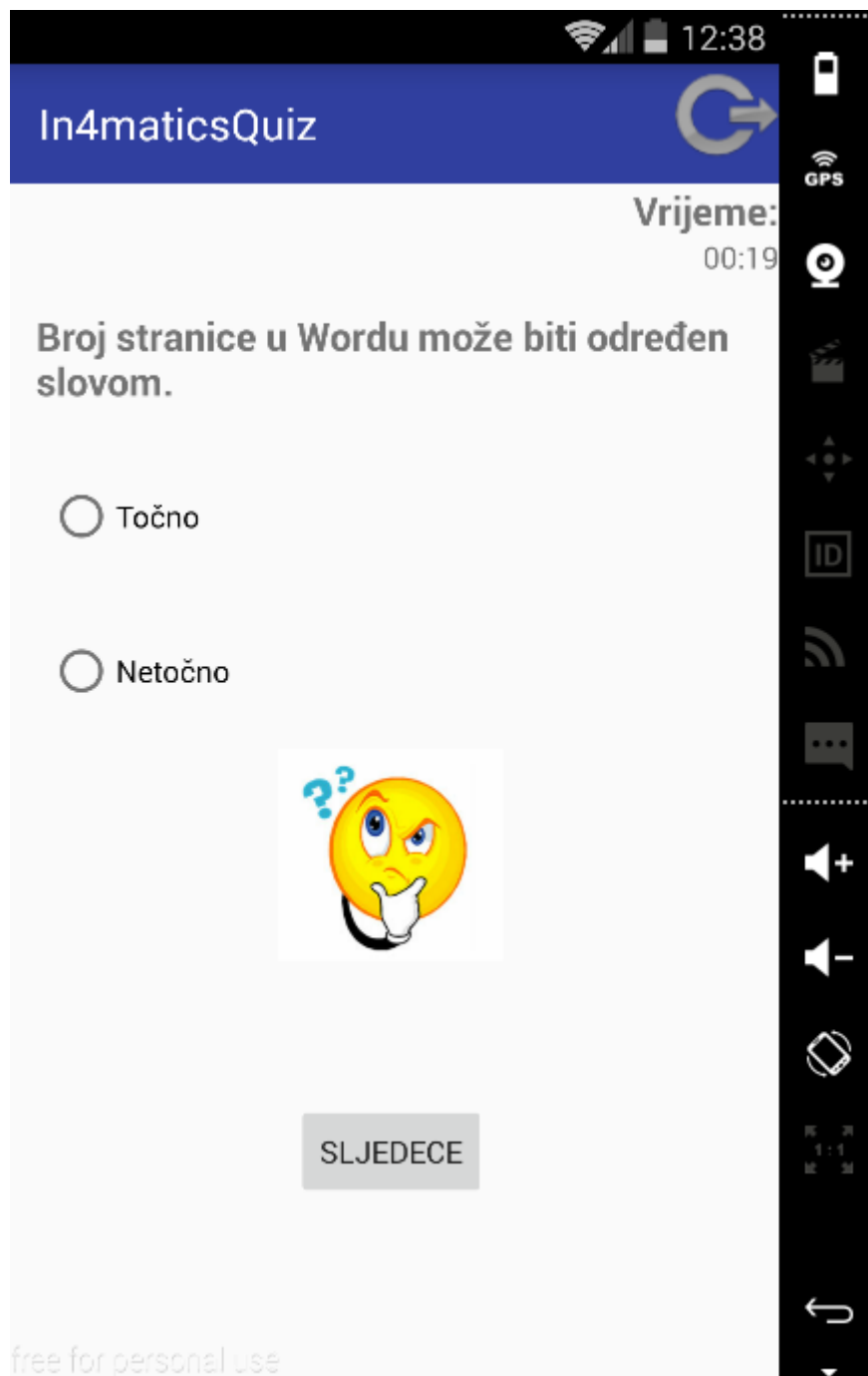
free for personal use

Slika 25. Pregled rang liste

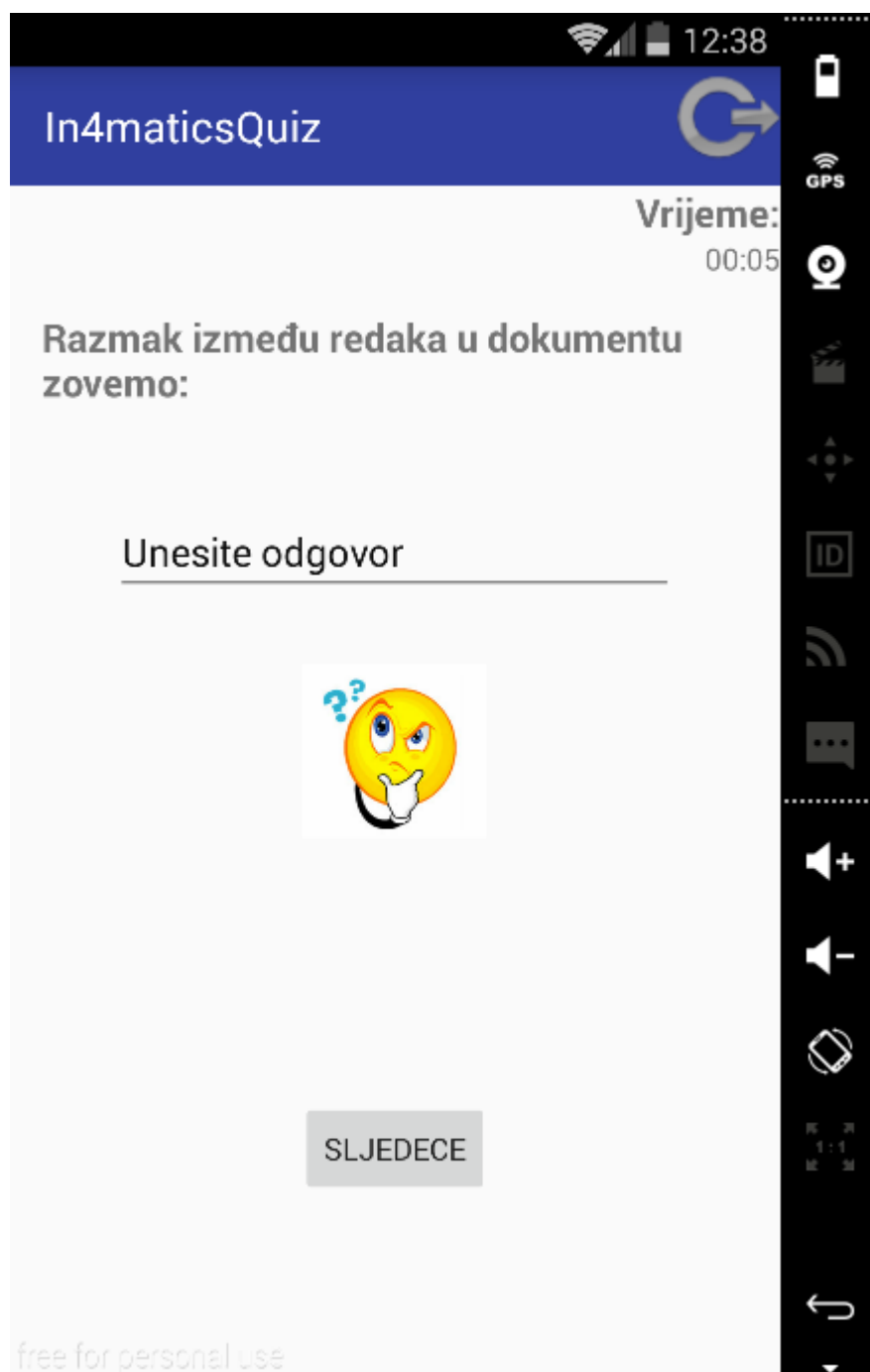
Nakon odabira *Provjeri znanje* u sučelju na *Slici 24.* korisniku se otvara 10 pitanja iz odabranog razreda. Tada počinje teći vrijeme i korisnik u zadano vrijeme mora riješiti kviz. U kvizu postoji tri vrste pitanja. Prva vrsta, koja je prikazana na *Slici 25.*, je pitanje u kojem je ponuđeno više odgovora u kojem jedan ili više odgovora može biti točnih. Druga vrsta je pitanje prikazano na *Slici 26.* Tu korisnik bira između *Točno* i *Netočno*. Treća vrsta pitanja je pitanje prikazano na *Slici 27.* Kod njega korisnik mora sam upisati točan odgovor.



Slika 25. Primjer pitanja s više ponuđenih odgovora



Slika 26. Primjer pitanja s ponuđenim točno i netočno odgovorom



Slika 26. Primjer pitanja s unosom odgovora

11. Specifikacija web servisa

Korisničko sučelje želi dobiti podatke za prikaz te traži od app modula podatke. App modul zove *WebServiceDataLoader.java* (u našem projektu nalazi se u app modulu u mapi loaders) koji pokreće *WebServiceAsyncTask.java*. *WebServiceDataLoader.java* neizravno implementira i sluša *WebServiceAsyncTask.java*.

U ws modulu nalaze se klase *AsyncTaskInnerResults.java*, *WebServiseAsyncTask.java*, *WebServiceParams.java* i jedan interface pod nazivom *WebServiceResultHandler.java*.

WebServiceAsyncTask.java kontaktira web servis kako bi dobio podatke od njega. Nakon što dobije podatke od njega on koristi klase *AsyncTaskInnerResults.java* i *WebServiceParams.java* kako bi proslijedio te podatke *WebServiceResultHandler.java*.

Na kraju app modul dostavlja podatke db modulu iz kojeg dohvaćamo podatke po potrebi.