Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jernej Grosar

"Ogrodje mobilne aplikacije za študente"

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2013

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00389/2013
Datum: 04.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JERNEJ GROSAR**

Naslov: **OGRODJE MOBILNE APLIKACIJE ZA ŠTUDENTE**
FRAMEWORK OF MOBILE APPLICATION FOR STUDENTS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Študentje na Fakulteti za računalništvo in informatiko potrebujejo mobilno aplikacijo, ki bi jim omogočala vpogled v ključne podatke sistem Moodle in v ključne podatke študijskega informacijskega sistema. Mobilna aplikacija bo imela veliko različnih funkcionalnosti, ki bodo razvite postopoma, kar nakazuje na potrebo po stabilnem ogrodju, ki bo različnim razvijalcem omogočalo enostavno dodajanje modulov.
Razvijte ogrodje mobilne aplikacije za študente v sistemu Android. Pri tem naj ogrodje predvideva tudi objavljanje reklamnih oglasov v aplikacij in za potrebe upravljanja z oglasi razvijte še spletno aplikacijo.

Mentor:

doc. dr. Rok Rupnik

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Jernej Grosar, z vpisno številko **63090043**, sem avtor diplomskega dela z naslovom:

Ogrodje mobilne aplikacije za študente

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 11. septembra 2013 Podpis avtorja:

Zahvalil bi se rad predvsem družini in punci, kateri so mi stali ob strani in me ves čas podpirali. Posebna zahvala gre tudi mentorju doc. dr Roku Rupniku, ki mi je bil v veliko pomoč in je bil vedno dosegljiv za vprašanja.

Kazalo

Seznam uporabljenih kratic

Povzetek

Abstract

1 Uvod **1**

2 Uporabljene tehnologije **3**

3 Mobilna aplikacija **5**

3.1 Analiza 6

3.2 Primer uporabe 6

3.3 Razvoj aplikacije 8

3.3.1 Podatkovna baza 8

3.3.2 Prijava 9

3.3.3 Oglasna sporočila 11

3.3.4 Vrstni red prikaza oglasnih sporočil 12

3.3.5 Ohranjanje stanja aplikacije 13

3.4 Testiranje 14

4 Spletna aplikacija **15**

4.1 Analiza 16

4.2 Primer uporabe 16

4.3 Razvoj aplikacije 18

4.3.1 Podatkovna baza spletne aplikacije 18

4.3.2 Preverjanje in pošiljanje podatkov 21
4.3.3 Varnost 22

5 Spletne storitve **25**
5.1 Analiza 25
5.2 Primer uporabe 25
5.3 Razvoj spletnih storitev 26
5.3.1 Pridobivanje oglasnih sporočil 26
5.3.2 Povečevanje števila klikov 27
5.3.3 Prijava 27
5.3.4 Varnost 27

6 Sklepne ugotovitve **29**

Literatura **30**

Slike **33**

Seznam uporabljenih kratic

HTML - Hyper Text Markup Language

XHTML - Extensible Hyper Text Markup Language

CSS - Cascading Style Sheets

PHP - Hypertext Preprocessor

AJAX - Asynchronous JavaScript and XML

MySQLi - MySQL Improved

JSON - JavaScript Object Notation

SQL - Structured Query Language

DOM - Document Object Model

UML - Unified Modeling Language

DBMS - Database Management System

SIM - Subscriber Identity Module

Povzetek

Ker se pri učnem procesu na Fakulteti za računalništvo in informatiko uporablja spletna aplikacija Moodle in se pri dostopu do le-te, preko spletnega brskalnika na mobilni napravi, prenese velika količina za uporabnika nerelevantnih podatkov, pri tem pa danes že skoraj vsaka aplikacija za svoje normalno delovanje uporablja prenos podatkov, je zelo pomembno, da se količina prenesenih podatkov zmanjša.

Da bi rešili ta problem, je bila razvita mobilna aplikacija, ki poleg prijave v spletno aplikacijo Moodle omogoča tudi oglaševanje. Kasnejše nadgradnje bi lahko implementirale tudi druge funkcionalnosti, ki so pomembne za študente. Ker aplikacija uporablja tudi oglaševanje, je bilo zato potrebno narediti spletno aplikacijo za dodajanje oglasnih sporočil in spletne storitve za dostop do le-teh.

Diplomsko delo je razdeljeno na tri glavne sklope. V prvem sklopu je opisana izdelava mobilne aplikacije, ki omogoča prijavo v spletno aplikacijo Moodle. Opisan je tudi način razvrščanja oglasnih sporočil. V drugem sklopu je predstavljena spletna aplikacija za dodajanje in urejanje oglasnih sporočil ter opis, kako je poskrbljeno za njeno varnost. Tretji sklop pa opisuje izdelavo in delovanje spletnih storitev.

Ključne besede:

Mobilna aplikacija, spletna storitev, spletna aplikacija, Moodle, Android

Abstract

Because in the learning process at the Faculty of Computer and Information science, Moodle web application is used and when it is accessed via web browser on mobile device, large amount of irrelevant information for user is transferred and as nowadays almost every mobile application uses data transfer for their normal operation, it is important that the amount of data transferred is reduced.

This mobile application was developed, to solve the problem with large amount of transferred data, which in addition to the Moodle web application log in, it also allows advertising. Later upgrades could include other functionalities important for students. Because this mobile application uses advertising it was necessary to create web application to allow creation and managing this advertising messages and web services to access them.

This thesis is divided into three parts. In the initial part the mobile application development is described which allows the Moodlee web application log in. This part also describes how advertising messages are sorted. The second part describes web application for adding and managing advertisement messages and how is for its safety taken care of. The third part describes how web services are made and how do they work.

Key words:

Mobile application, web service, web application, Moodle, Android

Poglavje 1

Uvod

Danes srečamo pametne mobilne naprave že na vsakem koraku in so v našem življenju že skoraj nepogrešljive. Uporabniki, ki uporabljajo te naprave, jih hočejo čim bolje izkoristiti in je zato zanje pomembno, da lahko z njimi dostopajo do vseh podatkov na svetovnem spletu kjerkoli in kadarkoli. Pri tem pa se pojavi težava, ker se pri dostopu do povprečne spletne strani ali spletne aplikacij prenese velika količina za uporabnika nerelevantnih podatkov. To na koncu meseca vodi do povečanja računa za telefon; če pa se uporabnik nahaja na območju, kjer je slabši mobilni signal, je potrebno čakati kar nekaj časa, da se ti podatki prenesejo. Zato je zelo pomembno, da se prenos podatkov optimizira in kolikor se le da zmanjša. Tu pa se lahko izkažejo mobilne aplikacije. Ker so te aplikacije specializirane za točno določene naloge, je z njimi mogoče zmanjšati prenos podatkov na minimalno raven, kar pomeni, da se prenesejo le podatki, ki jih uporabnik potrebuje.

Z upoštevanjem teh dejstev je nastala mobilna aplikacija, ki je namenjena študentom in omogoča dostop do vsebine predmetov s prenosom minimalne količine podatkov. Diplomsko delo obravnava prijavo v to aplikacijo, realizacijo dodajanja oglasnih sporočil s pomočjo spletne aplikacije, prenos le-teh s spletnega strežnika na mobilno napravo s pomočjo spletnih storitev in njihovega prikaza v mobilni aplikaciji.

Poglavje 2

Uporabljene tehnologije

Za izdelavo diplomskega dela so bile uporabljene naslednje tehnologije:

- HTML in CSS sta bili uporabljeni za vizualno oblikovanje spletne aplikacije.
- PHP je skriptni jezik, ki se izvaja na strežniku [7] in je bil uporabljen za dinamično kreiranje spletne aplikacije ter izdelavo spletnih storitev.
- JavaScrip je izvršljiv programski jezik, ki ga prevede ter izvrši spletni brskalnik. JavaScrip in jQuery (JavaScript knjižnica) sta bila uporabljena za preverjanje vnesenih podatkov v spletne obrazce ter za animacijo posameznih DOM objektov v spletni aplikaciji.
- DOM je jezikovno in okoljsko neodvisna metoda za predstavitev in delo z objekti v HTML, XHTML in XML.
- Ajax je bil uporabljen za asinhrono izmenjevanje podatkov med spletno aplikacijo in spletnim strežnikom, kar pomeni, da se lahko del podatkov na spletni strani osveži, ne da bi bilo potrebno ponovno naložiti celotno spletno stran [3].
- MySQL je sistem za upravljanje s podatkovno bazo. Uporabljen je bil za hranjenje podatkov na strežniku.

- MySQLi je gonilnik za relacijsko podatkovno bazo, ki se uporablja v PHP programskem jeziku [6]. Uporabljen je bil za pridobivanje podatkov iz podatkovne baze in njihovo shranjevanje vanjo.

- JSON je format za izmenjavo podatkov; je človeku berljiv, enostaven za strojno obdelavo in vsebuje malo dodatnih podatkov. Uporablja se pri spletnih storitvah za strukturiran prenos podatkov med spletnim strežnikom in odjemalcem (mobilno aplikacijo).

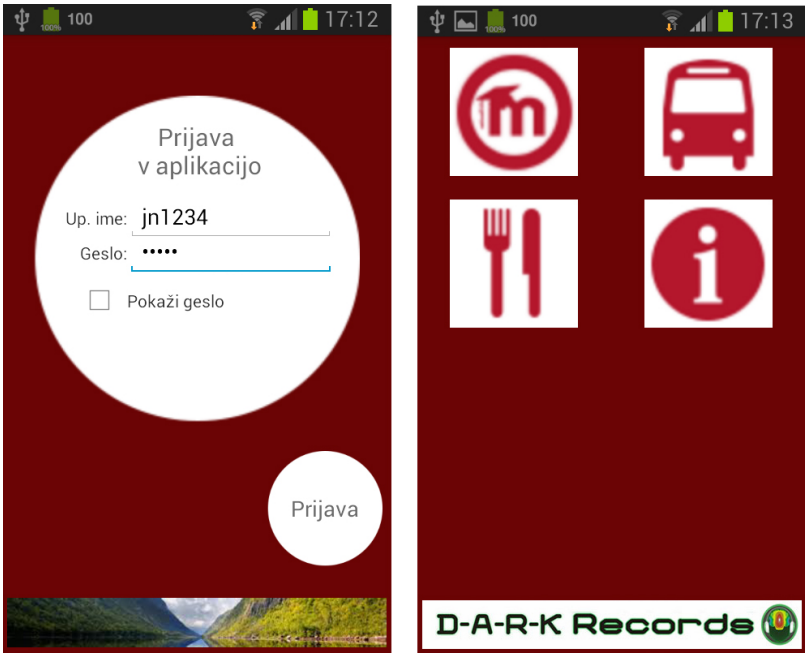
- Java je objektno orientiran programski jezik in je zasnovan tako, da je čim bolj neodvisen od platforme, na kateri se izvaja [5]. V diplomskem delu je bil uporabljen za izdelavo mobilne aplikacije.

- Android XML je način, kako se prikaže komponente Android uporabniškega vmesnika in zgradi le-tega.

- SQLite je različica SQL podatkovne baze in se uporablja na Android napravah za shranjevanje podatkov.

Poglavje 3

Mobilna aplikacija



Slika 3.1: Aplikacija na mobilni napravi. Levo prijava, desno glavni meni.

Aplikacija mFri, prikazana na sliki 3.1, je bila razvita za operacijski sistem Android. Omogoča prijavo v spletno aplikacijo Moodle, dostop do predmetov in gradiv, ki so objavljena na njem, ter ogled oglasnih sporočil, ki jih dodajo uporabniki in se prikazujejo v njej.

3.1 Analiza

Ker se pri učnem procesu na Fakulteti za računalništvo in informatiko v Ljubljani za posredovanje učnih gradiv in obveščanje študentov v glavnem uporablja spletna aplikacija Moodle, je pomembno, da je le-ta študentom dostopna vedno in povsod.

Danes se na mobilnih napravah uporablja vse več aplikacij, ki za svoje delovanje potrebujejo dostop do svetovnega spleta, mobilni operaterji pa za prenos podatkov zaračunavajo vedno več, zato je zelo pomembno, da se zmanjša količina prenesenih podatkov. Ta aplikacija je bila razvita, da bi bil dostop do spletne aplikacije Moodle enostavnejši in ne bi bilo potrebno dostopati do gradiv preko spletnega brskalnika, kar posledično zmanjša količino prenosa podatkov.

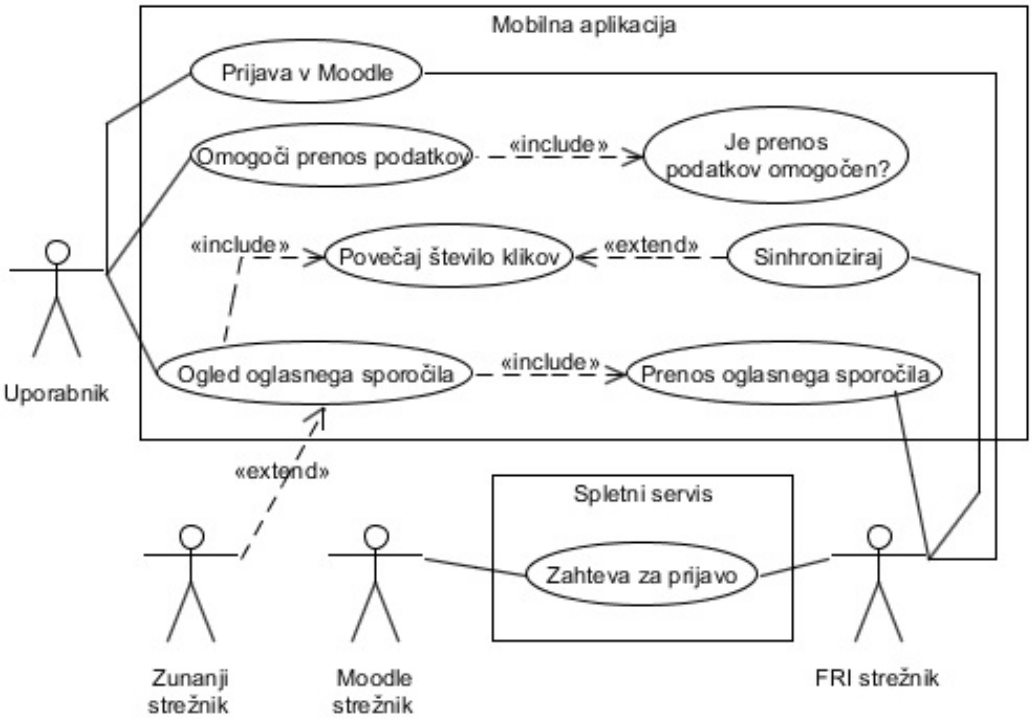
Ker študentje včasih potrebujejo pomoč pri učenju ali pa partnerja za seminarsko nalogo, podjetja pa večkrat želijo kaj sporočiti študentom, je v tej mobilni aplikaciji omogočeno tudi oglaševanje.

3.2 Primer uporabe

Na sliki 3.2 je prikazan UML diagram primera uporabe mobilne aplikacije za prijavo v spletno aplikacijo Moodle in prenos ter ogled oglasnih sporočil. Akter je v tem primeru študent, ki na mobilni napravi izvaja zaporedje akcij, katere ga pripeljejo do nekega rezultata.

Ko uporabnik požene mobilno aplikacijo, se najprej preveri ali je prenos podatkov omogočen. Če ni, aplikacija uporabnika o tem obvesti in mu ponudi možnost, da ga vklopi. Ko je prenos omogočen, se v ozadju spletni storitvi pošlje zahtevek za prenos seznama oglasnih sporočil. Ko ga ta vrne, se podatki shranijo v podatkovno bazo na telefonu, obenem se preveri, katere slike za predstavitev oglasnih sporočil so že prenesene, nato pa se prične prenos tistih, ki manjkajo. Ko so vse slike prenesene, se pričnejo prikazovati oglasna sporočila. Če si uporabnik želi ogledati trenutno prikazano oglasno sporočilo, lahko klikne nanj in v spletnem brskalniku se prikaže spletna stran, ki jo je

avtor oglasnega sporočila določil ob dodajanju le-tega.



Slika 3.2: Diagram primera uporabe mobilne aplikacije.

Uporabnik lahko med prenosom in med prikazovanjem oglasnih sporočil v aplikacijo nemoteno vpiše uporabniško ime in geslo, ki pa morata biti enaka tistima, katera uporablja za prijavo v spletno aplikacijo Moodle. Ob pritisku na gumb »Prijava« se na spletno storitev pošlje zahteva za prijavo, ta pa posreduje zahtevek spletni aplikaciji Moodle. Če je prijava uspešna, Moodle vrne spletni storitvi žeton za trenutno sejo in ga posreduje mobilni aplikaciji. Ko ta prejme žeton, ga shrani v pomnilnik mobilne naprave na lokacijo, ki je dostopna samo tej aplikaciji, nato pa se pokaže glavni meni za navigacijo po aplikaciji. Če je med prijavo prišlo do napake ali pa sta uporabniško ime ali geslo napačna, spletna storitev vrne obvestilo o napaki, katero se prikaže v mobilni aplikaciji. Uporabnik ima možnost ponovnega poizkusa prijave ali pa zapustiti aplikacijo.

3.3 Razvoj aplikacije

3.3.1 Podatkovna baza

Na sliki 3.3 je prikazan fizični ER diagram SQLite podatkovne baze na mobilni napravi v kateri se hranijo preneseni podatki o oglasnih sporočilih.

banner		
PK	bannerId	int(5)
	bannerName	varchar(400)
	bannerStartDate	varchar(10)
	bannerEndDate	varchar(10)
	bannerUrl	text
	bannerPortraitImage	varchar(20)
	bannerLandscapeImage	varchar(20)
	bannerUpdateVersion	int(3)
	bannerNoOfClicks	int(5)

Slika 3.3: Fizični ER diagram podatkovne baza na mobilni napravi.

V tej podatkovni bazi se hrani entiteta activeBanners, ki vsebuje naslednje atribute:

- **bannerId:** unikaten id posameznega oglasnega sporočila in hkrati primarni ključ,
- **bannerName:** opis oglasnega sporočila,
- **bannerStartDate:** datum, ko se bo oglasno sporočilo pričelo prikazovati,
- **bannerEndDate:** datum, ko se bo oglasno sporočilo nehalo prikazovati,
- **bannerUrl:** URL naslov do spletne strani, na kateri lahko uporabnik dobi več informacij o oglasnem sporočilu,

- **bannerPortraitImage:** : ime slike, ki se bo prikazovala kot oglasno sporočilo, ko bo naprava v pokončni postavitvi,
- **bannerLandscapeImage:** ime slike, ki se bo prikazovala kot oglasno sporočilo, ko bo naprava v ležeči postavitvi,
- **bannerUpdateVersion:** trenutna verzija oglasnega sporočila. Vrednost se spremeni, kadar uporabnik spremeni podatke oglasnega sporočila v spletni aplikaciji,
- **bannerNoOfClicks:** število klikov na oglasno sporočilo v mobilni aplikaciji.

3.3.2 Prijava

Ko uporabnik požene aplikacijo, je potrebno najprej preveriti, ali je podatkovna poveza na mobilni napravi omogočena. Za preverjanje se uporablja funkcija

```
isConnectedOrConnecting()
```

iz razreda *NetworkInfo* , ki preveri, ali je podatkovna povezava preko brezžičnega ali mobilnega omrežje že vzpostavljena ali pa se trenutno še vzpostavlja.

Če nobena od njiju ni aktivna, oziroma če nista v stanju vzpostavljanja, se uporabniku na zaslonu prikaže obvestilo, da mora omogočiti prenos podatkov in mu ponudi možnost, da odpre nastavitve mobilne naprave in omogoči prenos. Ko to stori, se lahko vrne v aplikacijo.

Sedaj lahko vpiše uporabniško ime in geslo, ki ju uporablja za prijavo v spletno aplikacijo Moodle. Uporabnik ima tudi možnost izpisa vnesenega gesla z znaki (črke, številke in znaki) namesto s pikami. To stori tako, da obkljuka potrditveno polje (angl. *checkbox*). Ob izbiri le-tega se kliče funkcija

```
setTransformationMethod()
```

s parametrom *null*, ki samodejno pretvori prikaz gesla s pikami v prikaz z znaki. Ko pa uporabnik odkljuka potrditveno polje, se kliče ista funkcija,

vendar tokrat s parametrom *new PasswordTransformationMethod()* ki spremeni prikaz gesla iz prikaza z znaki nazaj v prikaz s pikami.

Ob kliku na gumb »Prijava« se najprej preveri, ali je uporabniško ime pravilne dolžine in ali je uporabnik vnesel geslo. Če uporabniško ime ni pravilne dolžine, oziroma če ni vpisal gesla, se na ekranu prikaže opozorilo, v nasprotnem primeru pa se na spletno storitev pošlje poizvedba za prijavo. Če je prijava uspešna, spletna storitev vrne žeton za trenutno sejo, ki se kasneje uporablja za poizvedbe na spletno aplikacijo Moodle, ali pa za sporočilo o napaki in vrsto napake. Spletna storitev aplikaciji vrne JSON podatke, zato jih je potrebno, preden jih lahko aplikacija uporablja, pretvoriti s pomočjo Androidove knjižnice *json*.

Ob uspešni prijavi se žeton, ki ga je vrnila spletna storitev, shrani z uporabo funkcij razreda *SharedPreferences* v privatni pomnilniški prostor mobilne naprave, ki je viden samo tej aplikaciji ter tako druge aplikacije, nameščene na mobilni napravi, ne morejo dostopati do njega. Nato se trenutna aktivnost konča (aktivnost je ena fokusirana stvar, ki jo uporabnik lahko opravlja, navadno je to eno okno, ki se razteza čez celoten zaslon [2]) in požene nova; ta pa vsebuje glavni meni mobilne aplikacije (desni primer na sliki 3.1), preko katerega lahko uporabnik dostopa do vseh funkcionalnosti, ki mu jih ponuja mobilna aplikacija.

Če prijava ne uspe, se na zaslonu uporabniku izpiše vrsta napake, aktivnost, ki omogoča prijavo, pa se ne konča in ostane vidna, da se lahko uporabnik ponovno poizkusi prijaviti.

Ker deluje prijava tako, da pošlje poizvedbo na spletno storitev in ker lahko to traja dlje časa (zaradi počasne podatkovne povezave ali česa podobnega), je potrebno del programske kode, ki izvaja to poizvedbo, poganjati vzporedno v novi procesni niti. Android je že v osnovi zasnovan tako, da ne dovoli kakršnega koli povezovanja na splet v glavni procesni nitki, ker bi lahko to vodilo do neodzivnosti celotne aplikacije in posledično do končanja le-te.

3.3.3 Oglasna sporočila

Aplikacija omogoča tudi prikazovanje oglasnih sporočil, ki so vidna v spodnjem delu zaslona mobilne naprave. Ker se za prikazovanje le-teh uporabljajo slike, operacijski sistem Android pa je nameščen na mobilnih napravah, ki imajo bogato paleto različnih velikosti in ločljivosti zaslona, mora uporabnik ob oddaji oglasnega sporočila v spletni aplikaciji dodati tri slike različnih velikosti, da se lahko za vsako mobilno napravo ob prikazu izbere pravilna velikost slike in da se tako prepreči prevelika izguba kakovosti ob pomanjšanju ali povečanju slike.

Ob zagonu aplikacije se najprej preverijo dimenzije zaslona, nato se glede na njegovo širino določi, katera slika je najbolj primerna za to napravo. Ker aplikacija deluje tako v pokončni kot tudi v ležeči postavitvi mobilne naprave in ker imajo zasloni stranice različnih dimenzij, se za posamezno mobilno napravo uporabljata dve različno veliki sliki. Ko se te velikosti določijo, se v ozadju (v novi procesni nitki) kliče spletna storitev. Ta sprejme dva parametra, ki določita, katere velikosti slik naj se prenesejo. Konstante, ki sprejmejo ta dva parametra, so naslednje:

- **SMALL:** za prenos slike z dolžino 320 px in širino 50 px,
- **MEDIUM:** za slike z dolžino 468 px in širino 60 px,
- **LARGE:** za slike z dolžino 728 px in širino 90 px.

Spletna storitev nato vrne informacije o trenutno aktivnih oglasnih sporočilih ter imena pripadajočih slik.

Ko se te informacije prenesejo, se shranijo v podatkovno bazo telefona za kasnejšo uporabo. Preveri se tudi, ali so bile slike v preteklosti za to oglasno sporočilo že prenesene. Če katera slika ni na mobilni napravi ali pa jo je uporabnik s pomočjo spletne aplikacije medtem zamenjal, se le-ta ponovno prenese in shrani v privatni prostor aplikacijskega pomnilnika. Sedaj se lahko pričnejo prikazovati oglasna sporočila.

Slika 3.4: Delovanje animacije premika oglasnih sporočil.

Za menjavo oglasnih sporočil (slika 3.4) sta uporabljeni dve animaciji in sicer:

- prva skrije trenutno prikazano oglasno sporočilo tako, da animira premik slike navzgor s končno pozicijo izven območja, v katerem so prikazana oglasna sporočila (ko je slika izven območja prikaza oglasnega sporočila, ni vidna),

- druga pa prikaže naslednje oglasno sporočilo tako, da animira premik slike od spodaj navzgor z začetno pozicijo slike pod območjem prikaza oglasnega sporočila in končno pozicijo v območju prikaza le-tega.

Za obe animaciji se uporablja vgrajeni razred *AnimationUtils* operacijskega sistema Android.

3.3.4 Vrstni red prikaza oglasnih sporočil

Pri prikazu oglasnih sporočil je pomembno, da so ta pravično porazdeljena, kar pomeni, da se neko oglasno sporočilo, ki ima nastavljeno daljše časovno območje prikaza ali pa več ogledov (več klikov na oglasno sporočilo), prikaže kasneje. Da se to doseže, aplikacija izračuna neko vrednost V za vsako oglasno sporočilo, preden jih prične prikazovati in jih razvrsti glede na to vrednost od najmanjše do največje. Ta vrednost upošteva dolžino trajanja posameznega oglasnega sporočila in število ogledov le-teh ter se izračuna po formuli 3.1,

$$V = 1 - \frac{t}{\sum_{i=0}^{n} t_i} * \frac{a}{\sum_{j=0}^{n} a_j} \tag{3.1}$$

kjer je **t** = trajanje posameznega oglasnega sporočila, **a** = število klikov za posamezno oglasno sporočilo in **n** = števil vseh trenutno aktivnih oglasnih sporočil.

Posamezne vrednosti se shranijo v tabelo, za ureditev pa se uporabi urejevalni algoritem urejanje z vstavljanjem, ki uredi vrednosti od najmanjše do največje. Ta urejevalni algoritem je bil izbran, ker je:

- enostaven (mobilne naprave imajo slabše procesorske zmogljivosti),
- učinkovit pri skoraj urejenih podatkih (za urejanje skoraj urejenih podatkov potrebuje v najboljšem primeru O(n) primerjav),
- najhitrejši med kvadratnimi algoritmi pri urejanju majhnih količin podatkov (predpostavljamo lahko, da ne bo nikoli večjega števila (več kot 100) aktivnih oglasnih sporočil),
- za urejanje potrebuje le konstantno O(1) količino dodatnega pomnilnika, kar je pomembno, ker imajo mobilne naprave relativno malo pomnilnika,
- ne spreminja zaporedja elementov z enakimi vrednostmi [1] (ohranja zaporedje dodanih oglasnih sporočil).

Ko so te vrednosti urejene se prične prikazovanje oglasnih sporočil v takem vrstnem redu, kot so urejene te vrednosti.

3.3.5 Ohranjanje stanja aplikacije

Mobilni sistem Android je narejen tako, da se trenutno odprta aktivnost ob spremembi orientacije mobilne naprave (iz pokončne v ležečo ali pa obratno) ponovno zažene. Če aplikacija tega dejstva pri prenosu podatkov ne

bi upoštevala, bi se le-ti, ko bi uporabnik obrnil napravo, pričeli ponovno prenašati, kar bi posledično pomenilo večjo količino prenesenih podatkov. Ker pa je bila aplikacija narejena zato, da bi zmanjšala prenos podatkov, je bilo potrebno dodati metodo

```
onSaveInstanceState(Bundle outState),
```

katera se kliče, preden preide aktivnost v ozadje – to pomeni, da se kliče, preden se prične nova aktivnost ali pa se spremeni orientacija mobilne naprave. Tej metodi podamo vrednosti, ki jih želimo shraniti. Ko preide aktivnost ponovno v ospredje, pa se kliče

```
onRestoreInstanceState(Bundle savedInstanceState),
```

v kateri se ponovno nastavijo shranjene spremenljivke.

3.4 Testiranje

Zelo pomemben faktor pri razvoju programske opreme je tudi testiranje le-te. Pri razvoju Android mobilne aplikacije si razvijalec pri testiranju ali pa razhroščevanju le-te lahko pomaga z emulatorjem, ki simulira delovanje mobilne naprave, ali pa kar s samo mobilno napravo.

Pri razvoju te aplikacije sta bila za testiranje uporabljena tako emulator kot mobilna naprava. Emulator je bil v veliko pomoč, saj omogoča testiranje delovanja aplikacije na več različicah operacijskega sistema pri različnih ločljivostih in velikosti zaslona.

Poglavje 4

Spletna aplikacija

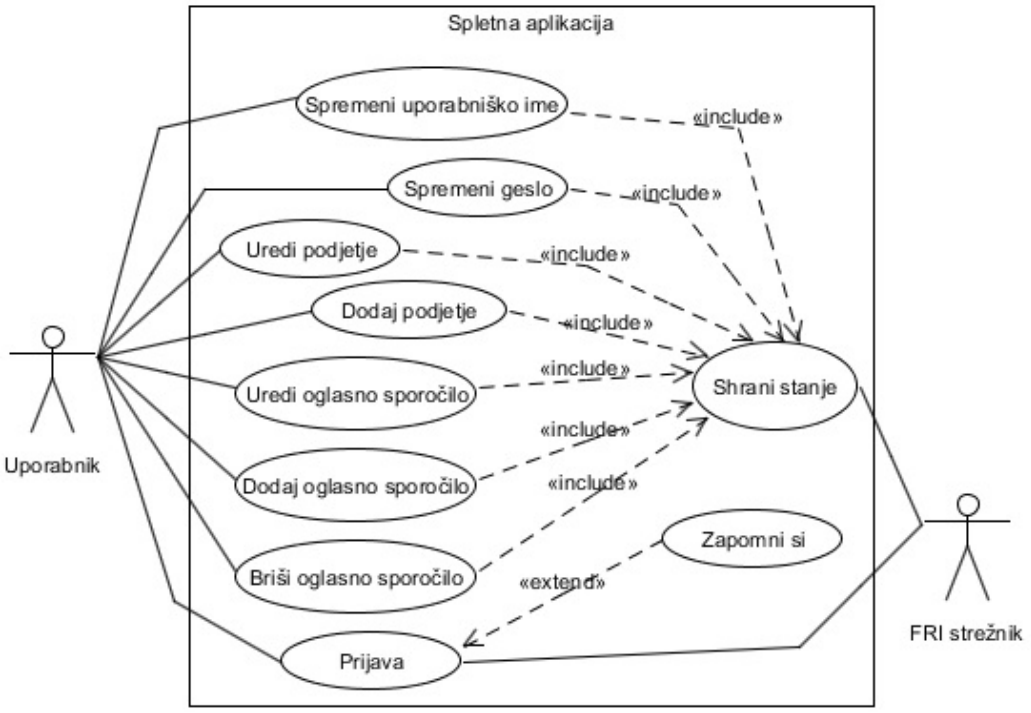
Slika 4.1: Spletna aplikacija za dodajanje oglasnih sporočil.

Spletna aplikacija je namenjena dodajanju, urejanju in brisanju oglasnih sporočil, ki se bodo prikazovala v mobilni aplikaciji. Na sliki 4.1 je prikazan primer seznama vseh oglasnih sporočil, ki se preberejo iz podatkovne baze spletnega strežnika.

4.1 Analiza

Ker mobilna aplikacija poleg dostopa do spletne aplikacije Moodle prikazuje tudi oglasna sporočila, je bilo potrebno izdelati spletno aplikacijo, ki bo v pomoč uporabnikom za enostavno dodajanje, urejanje ali brisanje oglasnih sporočil. Da bi le-ta lahko delovala, pa je potrebna tudi podatkovna baza.

4.2 Primer uporabe



Slika 4.2: Diagram primera uporabe spletne aplikacije.

Na sliki 4.2 je prikazan UML diagram primera uporabe spletne aplikacije za dodajanje oglasnih sporočil, akter pa je tudi v tem primeru študent, ki v spletnem brskalniku izvaja zaporedje akcij, katere ga pripeljejo do nekega rezultata.

Ob vstopu v spletno aplikacijo se mora uporabnik najprej prijaviti. Vpisati mora uporabniško ime in geslo, ima pa tudi možnost, da si ga aplikacija zapomni ter ga ob naslednjem obisku samodejno prijavi. Ko je uporabnik prijavljen, se mu prikaže seznam oglasnih sporočil, ki jih je že dodal, ob kliku na posamezno oglasno sporočilo pa se prikažejo podrobnosti le-tega. Med temi podrobnostmi se nahajata tudi možnosti urejanja ali brisanja.

Seznam oglasnih sporočil je za lažje iskanje razdeljen v štiri kategorije:

- vsi: tu so prikazana vsa oglasna sporočila, ki so shranjena v podatkovni bazi,
- zapadli: tu je seznam vseh oglasnih sporočil, ki se ne prikazujejo več,
- veljavni: tu se nahajajo vsa oglasna sporočila, ki so trenutno veljavna in se prikazujejo v mobilni aplikaciji,
- ne-prikazani: seznam vseh oglasnih sporočil, katerih datum pričetka prikazovanja je kasnejši od trenutnega.

V posamezni kategoriji jih lahko uporabnik po želji tudi razvrsti po datumu pričetka ali njihovem imenu.

Novo oglasno sporočilo lahko doda s klikom na gumb »Dodajte nov oglas« v meniju »Oglasi«. Prikaže se nova stran, kamor lahko vpiše naziv oglasnega sporočila, izbere podjetje, kateremu pripada to oglasno sporočilo ali pa svoje ime, če je to posameznik. Izbrati mora tudi datum, kdaj se bo sporočilo pričelo in končalo prikazovati, naslov spletne strani, kjer lahko uporabnik dobi več informacij o tem oglasnem sporočilu ter slike različnih velikosti, ki se bodo prikazovale v mobilni aplikaciji.

Če med dodajanjem oglasnega sporočila v izbirnem meniju ne zasledi imena podjetja ali posameznika, lahko le-to doda s klikom na gumb »Dodajte novo podjetje«. Prikaže se obrazec, v katerega vpiše podatke o podjetju ali posamezniku. Ko konča z vnašanjem, klikne na gumb »Shrani«, vneseni

podatki se shranijo, spustni meni seznama podjetji in posameznikov pa se osveži.

Ko uporabnik konča z vnosom podatkov, klikne na gumb »Naprej«. Prikaže se mu povzetek vsega, kar je vnesel in če so podatki pravilni, s klikom na gumb »Končaj« shrani oglasno sporočilo, drugače pa jih lahko spremeni s klikom na gumb »Popravi«.

Spletna aplikacija ima tudi možnost prikaza vseh posameznikov in podjetji, ki so že imeli dodan oglas. Te lahko razvrsti samo po imenu, ob kliku na posamezen vnos pa se, enako kot pri seznamu oglasnih sporočil, prikažejo podrobnosti o podjetju ali posamezniku.

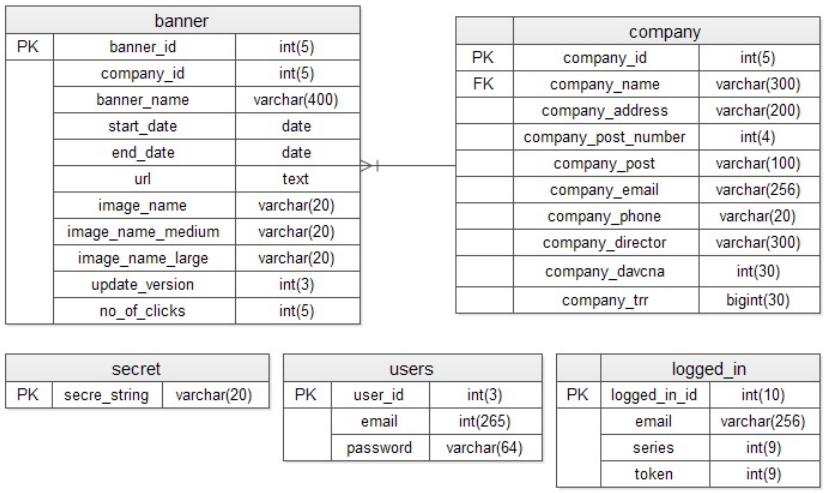
Uporabnik ima tudi možnost urejanja svojih podatkov za prijavo v spletno aplikacijo. To stori s klikom na gumb »Uredite profil« v zgornjem desnem kotu. Ko se stran naloži, lahko spremeni uporabniško ime ali geslo ter klikne »Shrani«, da se novi podatki shranijo.

4.3 Razvoj aplikacije

4.3.1 Podatkovna baza spletne aplikacije

Mobilna aplikacija za prikaz oglasnih sporočil uporablja podatke, ki se s spletnega strežnika prenesejo s pomočjo spletne storitve. Ti podatki so na strežniku shranjeni v MySQL podatkovni bazi.

Slika 4.3 prikazuje ER diagram zgradbe podatkovne baze, ki se uporablja na spletnem strežniku za hranjenje podatkov o oglasnih sporočilih in uporabnikih. V nadaljevanju je podroben opis teh entitet.



Slika 4.3: Fizični ER diagram podatkovne baze na strežniku.

secret: vsebuje 20-mestni ključ, ki se uporablja za preverjanje pristnosti poizvedbe za povečanje števila klikov, katera se kliče, ko uporabnik klikne na oglasno sporočilo na mobilni napravi.

logged_in:se shranjujejo podatki za samodejno prijavo uporabnika. Entiteta hrani:

- unikaten ključ (logged_in_id),
- uporabniško ime (email) in
- dve naključni vrednosti »series« in »token« ki se uporabljata za preverjanje pristnosti. Ti podatki se shranijo v podatkovno bazo, če uporabnik ob prijavi izbere možnost »Zapomni se me«.

users: seznam vseh uporabnikov, ki se lahko prijavijo v spletno aplikacijo. Entiteta hrani:

- unikaten ključ (user_id),
- seznam uporabniških imen (email) in

- seznam gesel (password), ki se uporabljajo za prijavo.

banner: seznam vseh oglasnih sporočil. Hrani naslednje atribute:

- unikaten ključ banner_id,
- tuj ključ company_id,
- opis oglasnega sporočila (banner_name),
- začetni in končni datum prikazovanja sporočila (start_date, end_date),
- povezavo na spletno stran z več informacijami (url),
- ime male slike, ki predstavlja oglas (image_name),
- ime srednje velike slike (image_name_medium),
- ime velike slike (image_name_large),
- trenutna verzija (update_version), ki pove ali je bilo oglasno sporočilo posodobljeno in
- število ogledov (no_of_clicks).

company: seznam vseh podjetji ali posameznikov in njihovih podatkov z naslednjimi atributi:

- unikaten primarni ključ company_id,
- ime podjetja ali posameznika (company_name),
- naslov podjetja ali posameznika (company_address),
- poštno številko (company_post_number),
- pošto (company_post),
- telefon (company_phone),

- predstavnika podjetja (company_director),
- davčno številko podjetja (company_davcna) in
- transakcijski račun podjetja (company_trr).

Entiteti banner in company sta povezani preko atributa company_id.

4.3.2 Preverjanje in pošiljanje podatkov

Aplikacija za vsa obvezna polja uporablja preverjanje vnesenih podatkov, preden se le-ti pošljejo v obdelavo. Med drugim se preverja, ali so bila izpolnjena vsa obvezna vnosna polja, ali so podatki pravilne oblike, kot na primer preverjanje strukture elektronskega naslova z uporabo regularnega izraza, ali je datum začetka predvajanja oglasnega sporočila pred datumom izteka letega in ali je izbrana slika za oglasno sporočilo pravilnega formata.

Slika 4.4: Primer preverjanja podatkov z Ajax-om.

Ker se za preverjanje podatkov uporablja skriptni jezik JavaScript, ki se izvaja na strani klienta, podatkov pri preverjanju, razen pri prijavi in spremembi podatkov za prijavo, ni potrebno pošiljati na strežnik, ampak se to izvede lokalno. Da bi aplikacija hitreje delovala in da bi bila uporabniku

prijaznejša, se pri nekaterih poizvedbah za prenos podatkov uporablja Ajax (slika 4.4), ki omogoča asinhroni prenos podatkov, kar pomeni, da ni potrebno osvežiti celotne, ampak samo del spletne strani. Ko Ajax s spletnega strežnika pridobi podatke, se le-ti prikažejo z uporabo funkcij, ki jih ponujata JavaScript in njena knjižnica jQuery.

Lahko se zgodi, da zaradi slabše internetne povezave ali večje količine podatkov prenos z Ajax-om traja dlje časa, in ker se to izvaja v ozadju, uporabnik tega ne opazi, ampak misli, da se stran ne odziva. To je lahko zanj neprijetno, ker ne ve, da se nekaj dogaja, zato je bilo potrebno na tista mesta v spletni aplikaciji, kjer se uporablja Ajax, dodati animacijo vrtečega se kroga, da ima uporabnik vizualno predstavo napredka.

4.3.3 Varnost

Pri razvoju aplikacij je zelo pomembno, da se razvijalec zaveda potencialnih nevarnosti, ki bi lahko onemogočile delovanje aplikacije ali razkrile uporabniške podatke.

Da bi preprečili morebitne napade na podatkovno bazo, je bilo potrebno, poleg tega, da se je kreiral nov uporabnik za podatkovno bazo, ki ima omejene pravice izvajanja operacij na njej, poskrbeti tudi za varnost v sami spletni aplikaciji. Napadalec lahko namreč z uporabo tehnike, imenovane SQL Injection, spremeni SQL poizvedbe in tako zaobide standardna preverjanja pristnosti ter tako pridobi dostop do tistih podatkov v bazi, do katerih nima pravice dostopati (npr. do vseh uporabniških imen in gesel), ali pa, kar je še huje, lahko včasih dostopa do ukazov gostujočega operacijskega sistema [9], na katerem je nameščena podatkovna baza.

Temu se izognemo tako, da namesto običajnih SQL stavkov, v katere neposredno vstavljamo uporabniške vnose, uporabljamo pripravljene stavke (angl. *prepared statements*).

Ti delujejo tako, da spletna aplikacija najprej pripravi predlogo SQL stavka in jo pošlje sistemu za upravljanje podatkovnih baz (DBMS), nekatere vrednosti imenovane parametri pa pusti nedefinirane in namesto njih doda

znak »?«. Nato DBMS ta stavek razčleni, ga prevede, izvede optimizacijo poizvedbe in ga shrani, ne da bi ga izvedel. Ko je stavek shranjen, lahko aplikacija poda manjkajoče parametre, DBMS pa ga izvrši in vrne morebitne rezultate.

Ti stavki so odporni na SQL Injection napade zato, ker morajo biti iz parametrov, ki se kasneje prenesejo preko drugega protokola, pravilno odstranjeni vsi posebni zanki (angl. *escape characters*) [8].

Poleg tega, da pripravljeni stavki preprečujejo SQL Injection, imajo tudi drugo dobro lastnost: pri prevajanju in optimizaciji zavzamejo manj dodatnega prostora, saj se izvedejo le enkrat in ne vsakič, ko se kliče ta stavek kot pri običajni obliki. Po drugi strani pa je slaba lastnost ta, da je enkratna izvedba takšnega stavka počasnejša kot izvedba običajnega stavka, ker se mora izvesti dodatna poizvedba na strežnik [8].

Drugi način, ki je bolj enostaven, hkrati pa ne tako učinkovit pri preprečevanju napada, je uporaba PHP funkcije za odstranjevanje posebnih znakov

```
mysqli_real_escape_string(mysqli $link , string $escapestr).
```

Pri izdelavi spletne aplikacije za dodajanje oglasnih sporočil sta bili uporabljeni obe tehniki. Tehnika priprave stavkov je bila uporabljena na tistih mestih, ki so bolj ranljiva za napade; to je tam, kjer imajo neavtorizirani uporabniki dostop do aplikacije. V našem primeru je to stran za prijavo uporabnika v spletno aplikacijo, stran za samodejno prijavo ter stran za odjavo. Stran za prijavo v spletno aplikacijo vsebuje obrazec, v katerega uporabnik vpiše uporabniško ime in geslo, zato lahko tu napadalec vnese neželen ukaz, stran za samodejno prijavo in odjavo pa bere podatke iz piškotkov, katere lahko napadalec priredi, če pridobi dostop do njih.

Druga tehnika je bila uporabljena na vseh ostalih straneh, do katerih napadalec nima direktnega dostopa, saj so te strani varovane tako, da preverjajo, ali je uporabnik prijavljen; za uporabnike, ki uporabljajo to aplikacijo, pa je bilo predpostavljeno, da ne predstavljajo potencialne nevarnosti.

Poglavje 5

Spletne storitve

5.1 Analiza

Da bi mobilna aplikacija dosegla cilje opisane v poglavju 3.1, je bilo potrebno ustvariti spletno storitev, ki deluje kot vmesnik za komunikacijo med mobilno in spletno aplikacijo. Pri razvoju mobilne aplikacije je zaradi občutljivosti podatkov, ki se nahajajo na spletnem strežniku, kjer je nameščena spletna aplikacija Moodle, zelo pomembno dejstvo, da se zagotovi čim večja varnost le-teh.

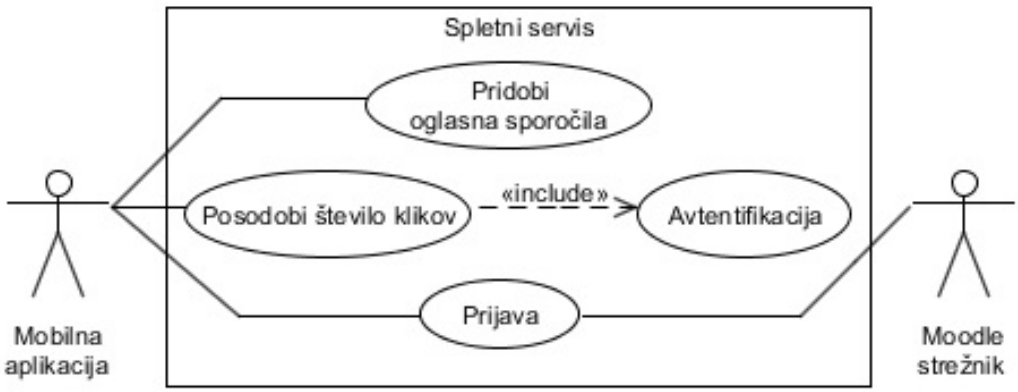
To je bilo med drugim mogoče doseči z uporabo spletnih storitev, ker je zaradi dodatne zaščite, ki jo vsebujejo le-te, za napadalca potreben večji napor, da pride do podatkov. Če pa že pridobi dostop do njih, se lahko spletna storitev preprosto izklopi in se tako zaščiti spletno aplikacijo Moodle pred napadom, ta pa še vedno nemoteno deluje dalje.

5.2 Primer uporabe

Spletne storitve se uporabljajo za prenos oglasnih sporočil iz spletnega strežnika na mobilno napravo ter komunikacijo mobilne aplikacije s spletno aplikacijo Moodle. Uml diagram primera uporabe le-tega je prikazan na sliki 5.1.

Ko se uporabnik želi prijaviti v mobilno aplikacijo, se kliče spletna storitev

za prijavo, ki pošlje zahtevo skupaj z uporabniškimi podatki spletni aplikaciji Moodle. Če slednja prejme pravilne podatke, prijavi uporabnika v sistem, spletni storitvi pa vrne žeton za trenutno sejo, kateri ga posreduje mobilni aplikaciji, v nasprotnem primeru pa spletna storitev vrne sporočilo o napaki ter vrsto napake, katero ji vrne spletna aplikacija Moodle.



Slika 5.1: Diagram primera uporabe spletne storitve.

Ob zagonu mobilne aplikacije se na spletno storitev za oglasna sporočila pošlje zahtevek za seznam le-teh. Spletna storitev iz podatkovne baze pridobi vsa trenutno aktivna oglasna sporočila ter jih vrne mobilni aplikaciji. Ker je pri oglasnih sporočilih potrebno beležiti število ogledov, se ob kliku na posamezno sporočilo kliče spletna storitev, ki v spletni podatkovni bazi poveča število klikov za to sporočilo.

5.3 Razvoj spletnih storitev

5.3.1 Pridobivanje oglasnih sporočil

Ta spletna storitev omogoča pridobivanje podatkov o vseh trenutno aktivnih oglasnih sporočilih. Ker se v mobilni aplikaciji za prikazovanje oglasnih sporočil uporabljajo slike, sprejme ta preko POST metode dva parametra, ki se uporabljata za izbiro ustreznih velikosti slik iz podatkovne baze.

Zaradi relativno velike količine podatkov o oglasnih sporočilih se zato, da je pri prenosu čim manj dodatnih podatkov ter, da je obdelava podatkov lažja, uporablja JSON format. PHP programski jezik že sam po sebi omogoča zelo enostavno strukturiranje JSON sporočila. Vse podatke, ki jih želimo pretvoriti v JSON format, shranimo v podatkovno strukturo polje, to pa podamo kot parameter funkciji *json_encode(value)*, ki vrne JSON sporočilo.

5.3.2 Povečevanje števila klikov

Spletna storitev, ki to omogoča, dobi preko POST metode dva parametra, in sicer unikatno številko oglasnega sporočila – bannerId in ključ dolžine 20 znakov. Ta ključ se uporablja za zaščito spletne storitev pred nelegalno uporabo. Ob klicanju se namreč preveri ali se ključ, podan kot parameter, ujema s ključem, shranjenim v podatkovni bazi. Če se ujemata, potem se število klikov v podatkovni bazi za oglasno sporočilo, ki ima unikatno identifikacijsko številko enako vrednosti parametra bannerId, poveča za ena, sicer se ne zgodi nič.

Ta spletna storitev zaradi varnosti ne vrača ničesar, da morebitni napadalec ne more ugotoviti varnostnega ključa.

5.3.3 Prijava

Spletna storitev, ki služi prijavi, je samo vmesnik med mobilno aplikacijo in spletno aplikacijo Moodle. Ta sprejme dva parametra, in sicer uporabniško ime in geslo, katera se posredujeta spletni aplikaciji Moodle. Slednji v odgovoru vrne žeton za trenutno sejo ali vrsto napake, če se je ta pojavila. Spletna storitev to preveri in vrne ustrezno JSON sporočilo.

5.3.4 Varnost

Spletna storitev za pridobivanje podatkov o oglasnih sporočilih je zaradi dostopa do podatkovne baze varovana enako kot spletna aplikacija za dodajanje le-teh, in sicer z uporabo pripravljenih stavkov.

Podobno je varovana tudi spletna storitev za povečevanje števila klikov z razliko, da se poleg uporabe pripravljenih stavkov preverja tudi, ali je varnostni ključ pravilen.

Spletna storitev za prijavo pa ne potrebuje varovanja, ker ta samo posreduje podatke spletni aplikaciji Moodle, ki ima že sama po sebi varnostne mehanizme.

Poglavje 6

Sklepne ugotovitve

V sklopu tega diplomskega dela so bile razvite, poleg prijave v mobilno aplikacijo za uporabo spletne aplikacije Moodle, tudi spletne storitve, ki omogočajo prenos oglasnih sporočil ter spletna aplikacija za dodajanje in urejanje le-teh.

Ta mobilna aplikacija je bila razvita za mobilne naprave z operacijskim sistemom Android različice 2.3.3 ali novejšim, to pa zato, ker je skupni delež naprav s starejšim operacijskim sistemom zanemarljiv – 1. avgusta 2013 je bilo na trgu manj kot 4% takih naprav [4].

Uporabniška vmesnika, tako mobilne kot tudi spletne aplikacije, sta bila razvita tako, da sta čim bolj enostavna, hkrati pa ponujata vse glavne funkcionalnosti, ki so pomembne za delovanje aplikacije. Oba sta skladna z novo celostno podobo Fakultete za računalništvo in informatiko v Ljubljani.

V prihodnjih nadgradnjah bi se lahko za izboljšavo prijave v mobilno aplikacijo uporabil eden od dveh načinov:

- uporaba uporabnikovega osebnega certifikata, ki bi imel shranjenega na mobilni napravi,
- za uporabnika enostavnejša pa bi bila uporaba certifikata, shranjenega na SIM kartici, ki ga mobilni operaterji uporabljajo za preverjanje identitete SIM kartice, ko je ta prijavljena v mobilno omrežje [10].

Pri obeh načinih bi lahko aplikacija uporabljala certifikat za podpis zah-

teve za prijavo. Mogoča bi bila tudi implementacija drugih funkcionalnosti, ki niso povezane s spletno aplikacijo Moodle in so pomembne za študente ter bi bile tako vse uporabne funkcionalnosti združene v eni aplikaciji.

Literatura

[1] T .H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to algorithems, third edition.* Massachusetts Institute of Technology, 2009, str. 12.

[2] (2013) Activity. Dostopno na: http://developer.android.com/reference/android/app/Activity.html.

[3] (2013) Ajax (programiranje). Dostopno na: http://sl.wikipedia.org/wiki/Ajax_(programiranje).

[4] (2013) Dashboards. Dostopno na: http://developer.android.com/about/dashboards/index.html.

[5] (2013) Java (programming language). Dostopno na: http://en.wikipedia.org/wiki/Java_(programming_language).

[6] (2013) MySQLi. Dostopno na: http://en.wikipedia.org/wiki/MySQLi.

[7] (2013) PHP. Dostopno na: http://en.wikipedia.org/wiki/PHP.

[8] (2013) Prepared statement. Dostopno na: http://en.wikipedia.org/wiki/Prepared_statement.

[9] (2013) SQL Injection. Dostopno na: http://php.net/manual/en/security.database.sql-injection.php.

[10] (2013) Subscriber identity module. Dostopno na: http://en.wikipedia.org/wiki/Subscriber_identity_module.

Slike

Slika 3.1 Aplikacija na mobilni napravi. Levo prijava, desno glavni meni. 5
Slika 3.2 Diagram primera uporabe mobilne aplikacije. 7
Slika 3.3 Fizični ER diagram podatkovne baza na mobilni napravi. . . 8
Slika 3.4 Delovanje animacije premika oglasnih sporočil. 12

Slika 4.1 Spletna aplikacija za dodajanje oglasnih sporočil. 15
Slika 4.2 Diagram primera uporabe spletne aplikacije. 16
Slika 4.3 Fizični ER diagram podatkovne baze na strežniku. 19
Slika 4.4 Primer preverjanja podatkov z Ajax-om. 21

Slika 5.1 Diagram primera uporabe spletne storitve. 26