

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Dušan Miška

**Razvoj mobilne aplikacije z uporabo
standardnega ogrodja**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana 2014

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00097 / 2013
Datum: 8.4.2013

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

Kandidat: **DUŠAN MIŠKA**

Naslov: **RAZVOJ MOBILNE APLIKACIJE Z UPORABO STANDARDNEGA
OGRODJA
THE DEVELOPMENT OF MOBILE APPLICATION WITH THE USE OF
STANDARD FRAMEWORK**

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Z uporabo standardnega ogrodja PhoneGap razvijte prototip mobilne aplikacije, ki študentom in pedagogom na fakulteti omogoča uporabo storitev študijskega informacijskega sistema. Aplikacija naj študentom nudi mobilnosti primerne in mobilnosti prilagojene storitve. Ker nimate možnosti dostopa do produkcijske podatkovne baze študijskega informacijskega sistema, uporabite lokalno podatkovno bazo z namišljenimi podatki na mobilnem aparatu. Delovanje aplikacije preizkusite na vsaj dveh operacijskih sistemih mobilnih aparatov.

Mentor:

doc. dr. Rok Rupnik



Dekan:

prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Dušan Miška, z vpisno številko **63080161**, sem avtor diplomskega dela z naslovom:

Razvoj mobilne aplikacije FriStudent

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 18. februar 2014

Podpis avtorja:

Za pomoč in nasvete pri izdelavi diplomske naloge se zahvaljujem mentorju doc. dr. Roku Rupniku.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Tehnologije uporabljene pri razvoju mobilne aplikacije	3
2.1	HTML	3
2.2	CSS	3
2.3	JavaScript	4
2.4	jQuery	4
2.5	jQuery Mobile	4
2.6	AJAX	5
2.7	PHP	5
2.8	MySQL	5
2.9	Ogrodje PhoneGap	6
2.10	Razvojno okolje Xcode	6
3	Mobilna aplikacija	7
3.1	Uporaba in razvoj mobilnih aplikacij	7
3.2	Vrste mobilnih aplikacij	8
3.3	Vrste dotikov zaslona in njihova uporaba	9
4	Razvoj mobilne aplikacije	11
4.1	Mobilna aplikacija FriStudent	11

KAZALO

4.2	Arhitekture mobilne aplikacije	11
4.3	Kreiranje podatkovne baze	12
4.4	Struktura podatkovne baze	13
4.5	Konceptualni model podatkovne baze	14
4.6	Diagram primera uporabe	16
5	Implementirane funkcionalnosti mobilne aplikacije	17
5.1	Prijava v sistem	17
5.2	Nadzorna plošča	19
5.3	Obvestila	20
5.4	Izpitni roki	21
5.5	Kartotečni list	22
5.6	Rezultati izpitnega roka	23
5.7	Anketa	24
5.8	Metoda Ajax	25
6	Sklepne ugotovitve	27
	Literatura	28

Kazalo slik

2.1	Podpora različnih API vmesnikov na različnih platformah . . .	6
3.1	Seznam spletnih trgovin s svojimi platformami in lastniki . . .	7
4.1	Arhitektura mobilne aplikacije FriStudent	12
4.2	Konceptualni model podatkovne baze FriStudent	15
4.3	Diagram primera uporabe za mobilno aplikacijo FriStudent . .	16
5.1	Prikaz prijave v mobilno aplikacijo FriStudent	18
5.2	Prikaz nadzorne plošče za študenta v mobilni aplikaciji Fri- Student	20
5.3	Prikaz nadzorne plošče za profesorja v mobilni aplikaciji Fri- Student	20
5.4	Prikaz seznama obvestil v mobilni aplikaciji FriStudent	21
5.5	Prikaz dodajanja obvestil v mobilni aplikaciji FriStudent . . .	21
5.6	Prikaz seznama izpitnih rokov v mobilno aplikacijo FriStudent	22
5.7	Prikaz prijave na izpitni rok v mobilni aplikaciji FriStudent . .	22
5.8	Prikaz dodajanja izpitnega roka v mobilni aplikaciji FriStudent	22
5.9	Prikaz kartotečnega lista v mobilni aplikaciji FriStudent	23
5.10	Prikaz vnosa doseženih točk in ocene v mobilni aplikaciji Fri- Student	24
5.11	Prikaz rezultatov opravljenih izpitnih rokov v mobilni aplika- ciji FriStudent	24
5.12	Prikaz izpolnjevanja ankete v mobilni aplikaciji FriStudent . .	25

KAZALO SLIK

5.13 Prikaz pregleda rezultatov ankete v mobilni aplikaciji FriStudent 25

Seznam uporabljenih kratic in simbolov

AJAX (*angl.*) Asynchronous JavaScript and XML; asinhroni JavaScript in XML.

API (*angl.*) Application Programming Interface; vmesnik za komunikacijo med različnimi programskimi deli aplikacije.

CSS (*angl.*) Cascading Style Sheets; stilna predloga spletne strani s pomočjo katere določamo obliko spletne strani.

GPS (*angl.*) Global Positioning System; sistem globalnega določanja lege.

HTML (*angl.*) Hypertext Markup Language; označevalni jezik za oblikovanje večpredstavnostnih dokumentov.

HTTP (*angl.*) Hypertext Transfer Protocol; protokol za prenos informacij na spletu. Prvotno je bil namenjen objavljanju in prejemanju HTML strani.

IDE (*angl.*) Integrated Development Environment; integrirano razvojno okolje.

iOS (*angl.*) iPhone Operating System; Applov operacijski sistem za mobilne naprave.

JS JavaScript; objektni skriptni programski jezik.

MySQL (*angl.*) My Structured Query Language; sistem za upravljanje s podatkovnimi bazami.

OS (*angl.*) Operating System; operacijski sistem.

PHP (*angl.*) Hypertext Preprocessor; je skriptni jezik na strani strežnika, namenjen za razvoj spletnih aplikacij.

KAZALO SLIK

SDK (*angl.*) Software Development Kit; komplet orodij za razvijanje programske opreme.

SQL (*angl.*) Structured Query Language; strukturirani povpraševalni jezik za delo s podatkovnimi zbirkami.

WWW (*angl.*) World Wide Web; svetovni splet.

XML (*angl.*) Extensible Markup Language; format podatkov za izmenjavo strukturiranih dokumentov v spletu.

Povzetek

Število pametnih telefonov se iz leta v leto večja. Posledično narašča tudi število novih mobilnih aplikacij. Zaradi številnih ponudnikov pametnih telefonov, ki delujejo na različnih platformah, se povečuje tudi število platform. Na trgu so se zato pojavila različna ogrodja oz. knjižnice za medplatformski razvoj mobilnih aplikacij. Tema diplomskega dela je razvoj mobilne aplikacije, ki bi pripomogla k enostavnejši dostopnosti do elektronskega študenta. Razvoj temelji na ogrodju PhoneGap. Ogrodje podpira vse najpomembnejše platforme. Mobilna aplikacija omogoča prijavo v sistem tako študentom kot profesorjem. Funkcionalnosti mobilne aplikacije se med študentom in profesorjem razlikujejo. Študentom omogoča pregled obvestil, ogled izpitnih rokov, prijavo na izpitne roke, pregled rezultatov opravljenih izpitnih rokov, izpolnitev anket in pregled kartotečnega lista. Profesorju pa omogoča objavo obvestil, vpis novih izpitnih rokov, vpis točk in ocene posameznega izpitnega roka za posameznega študenta in objavo anket.

Ključne besede:

mobilna aplikacija, razvoj mobilne aplikacije, PhoneGap

Abstract

Quantity of smart phones is increasing every year. Consequently also the number of new mobile applications is bigger. Because there are many providers of smart phones, which run on different platforms, there is also an increase of the number of platforms. Therefore different frameworks or libraries for cross-platform development of mobile applications were made on the market. Main topic of my thesis is the development of mobile applications that would contribute to easier access to electronic student. The development is based on the framework PhoneGap. The frame supports all major platforms. Mobile application allows students and professors to log into the system. Functionalities of mobile application are different between teachers and students. Mobile application allows students to review notices, see the examination dates, check examination dates, review the results of the exam period, fulfill surveys and review indexing data sheet. Whereas the mobile application for professor allows him to publicate notices, entry new examination dates, entry points and evaluation of exam period for each student and also post surveys.

Ključne besede:

mobile application, mobile application development, PhoneGap

Poglavje 1

Uvod

Nekoč so študentje imeli samo knjižico imenovano indeks in v njem vpisane študijske predmete z doseženimi ocenami. Z razvojem tehnologije je knjižico zamenjal Elektronski študent. S to spletno aplikacijo lahko vsak študent pregleduje svoje šolske dejavnosti. Omogoča vpis v letnik, pregled tekočih obvestil, vpogled in prijavo na izpitne roke, pregled doseženih ocen, nadzor nad doseženim povprečjem ocen, itd.

V preteklosti smo imeli majhne telefone, ki so imeli majhne zaslone in velike tipkovnice. Z njimi smo večinoma opravljali le osnovna opravila, kot so: telefoniranje, shranjevanje telefonskih števil, uporaba budilke, pisanje kratkih sporočil, itd. Sedaj imamo vedno večje telefone, ti so zmogljivejši in lažji. Z njimi lahko brskamo po internetu, pregledujemo elektronsko pošto, uporabljamo internetne aplikacije.

Ob vse večjem številu pametnih telefonov, ki so občutljivi na dotik, se je pojavilo tudi veliko število različnih operacijskih sistemov. Zaradi naraščajočega števila operacijskih sistemov smo pri razvoju novih mobilnih aplikacij nekoliko omejeni, saj izmed operacijskih sistemov protrebuje svoj lastni razvoj v svojem okolju.

Moje diplomsko delo obravnava razvoj mobilne aplikacije z osnovnimi funkcionalnostmi, ki bi študentom preko pametnega telefona omogočala dostop do elektronskega študenta. Razvoj te mobilne aplikacije temelji na ogrodju PhoneGap. Povpraševanje po raznovrstnih aplikacijah raste iz dneva v dan. V kratkem pregledu trga sem ugotovil, da trenutno fakultete še ne uporabljajo tovrstnih mobilnih aplikacij, ki bi študentom olajšale dostop do podatkov.

Poglavje 2

Tehnologije uporabljene pri razvoju mobilne aplikacije

2.1 HTML

HTML je nabor opisnih simbolov ali šifer, vstavljenih v datoteko, namenjeno prikazu na World Wide Web-u. Enostavneje, označevalni jezik za izdelavo spletnih strani. Oznake povedo spletnemu brskalniku, kako naj se stran prikaže. Kodo razdelimo na elemente, nekateri jih poimenujejo tudi oznake. Nekateri elementi pridejo v parih in prikazujejo, kje se je element začel in kje je končal.

2.2 CSS

CSS je jezik, ki predstavlja slog stilske predloge. Najpogosteje se ga uporablja na spletnih straneh, napisanih v HTML in XHTML jeziku, redkeje pa tudi za ostale jezike, kot so: XML, SVG in XUL. Namenjen je predvsem za ločevanje vsebine dokumenta in izgleda. Z njim nastavljamo postavitev elementov, tip in barvo pisave ter vrsto drugih funkcionalnosti. Za predstavitev strani moramo upoštevati pravilno definicijo in obnašanje posameznega stila na posamezni element, v kolikor želimo, da je sintaksa pravilna. V zadnjem

času se pogosto uporablja tudi najnovejša različica CSS3, ki ima številne različne podpore posameznih stilov glede na vrsto brskalnika. Z njegovo pomočjo si oblikovalci olajšajo izdelavo nekaterih stilskih dodatkov. Pri mobilnih aplikacijah ne oblikujemo samo stil predloge, ampak lahko za različne velikosti mobilnih naprav (pametni telefon ali tablični računalnik) izdelamo različne izgled aplikacije. To nam veliko pripomore pri uporabniškem vmesniku.

2.3 JavaScript

JavaScript je interpretativni programski jezik, kar pomeni, da interpretor izvede program direktno brez dodatnega pretvarjana v nižje nivojski programski jezik. Prvotno se je uporabljal zgolj za interaktivno komunikacijo med brskalnikom in uporabnikom, zadnje čase vse pogosteje tudi v razvoju iger in v namiznih aplikacijah.

2.4 jQuery

jQuery je knjižnica, ki vsebuje posamezne metode iz JavaScript jezika za poenostavitev programiranja v različnih brskalnikih. S pomočjo jQuery metod lahko z enostavnim klicem posamezne metode izvedemo zelo zapletene programske kode jezika JavaScript, kot so: ustvarjanje elementov, izvajanje manipulacije DOM, upravljanje atributov, ki opravljajo HTTP zahteve, izvajanje Ajax klicev, itd. Knjižnica tudi omogoča kreiranje različnih animacij, ki pripomorejo do lepšega izgleda spletne aplikacije.

2.5 jQuery Mobile

jQuery Mobile je optimizirana JavaScript knjižnica, ki jo je razvila skupina jQuery. Razvoj se osredotoča na ustvarjanje okvirja, ki je združljiv s široko paleto pametnih telefonov in tabličnih računalnikov. jQuery Mobile

knjižnica je združljiva z drugimi mobilnimi aplikacijami, ogrodji in platformami, kot so: PhoneGap, Worklight, ipd.

2.6 AJAX

Ajax je skupina medsebojno povezanih tehnologij, ki se izvaja na strani odjemalca za ustvarjanje asinhronih spletnih aplikacij. S pomočjo Ajax-a, lahko spletna aplikacija pošilja in pridobiva podatke iz strežnika asinhrono (v ozadju), ne da pri tem posegala v izgled trenutno prikazane strani aplikacije. Podatke je mogoče pridobiti z uporabo XMLHttpRequest ali JSON. Pri tem ni potrebno, da so zahteve asinhrono. Dobljen rezultat lahko prikažemo v obliki HTML in CSS. S pomočjo JavaScript jezika dinamično posegamo v samo DOM strukturo aplikacije. Uporabniku se istočasno prikazujejo rezultati na zaslonu.

2.7 PHP

PHP je široko uporabljen odprtokodni skriptni jezik, ki je primeren za razvoj spletnih aplikacij. Razlika med jezikoma JavaScript in PHP je, da se JavaScript izvaja izključno na strani odjemalca, PHP pa na strani strežnika. Odjemalec torej dobi le rezultat PHP kode v HTML obliki. Kodo lahko pišemo direktno v HTML programsko kodo. Jezik PHP je zelo preprost, ker ponuja številne napredne funkcije za strokovno programiranje.

2.8 MySQL

MySQL je najbolj raširjen sistem za upravljanje relacijskih podatkovnih baz. Deluje na strežniku in zagotavlja večjemu šteilu uporabnikov dostop do številnih podatkov v posamezni bazi. Za dostopanje do podatkov izvajamo različne poizvedbe, ki nam vračajo rezultate željenih podatkov.

2.9 Ogradje PhoneGap

Ogradje PhoneGap je brezplačen odprtokodni sistem, ki omogoča ustvarjanje mobilnih aplikacij z uporabo spletnih tehnologij, kot so: HTML, CSS in JavaScript ter s pomočjo standardnega web API-ja. Prednost ogradja PhoneGap je uporaba obstoječega spletnega znanja, zato je tudi lažje učljiv. Vse JavaScript funkcije se pretvorijo v native funkcije, medtem ko se ostale funkcije povežejo na strežnik s pomočjo Ajax klicev.

	iPhone / Phone 3G	iPhone 3GS and newer	Android	Blackberry OS 5x	Blackberry OS 6.0+	WebOS	Windows Phone 7 + 8	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	X	X

Slika 2.1: Podpora različnih API vmesnikov na različnih platformah

2.10 Razvojno okolje Xcode

Xcode je integrirano razvojno okolje, ki vsebuje zbirko orodij za razvoj programske opreme. Razvil ga je podjetje Apple. Program je namenjen izključno razvoju programske opreme za operacijski sistem OSX in iOS. Podprt je s programskimi jeziki C++, Objective-C in Java. Program vsebuje različne simulatorje, s katerimi lahko simuliramo delovanje mobilnih aplikacij.

Poglavje 3

Mobilna aplikacija

3.1 Uporaba in razvoj mobilnih aplikacij

Mobilna aplikacija je programska aplikacija zasnovana za delovanje na pametnih telefonih, tabličnih računalnikih in drugih mobilnih napravah. V večini primerov so to že obstoječe aplikacije iz osebnih računalnikov. Za mobilne aparate so zasnovane tako, da vsebujejo le nekatere od glavnih podpornih funkcionalnosti spletne aplikacije. Mobilne aplikacije v večini primerov pridobivamo iz spletnih trgovin. Vsaka platforma ima svojo spletno trgovino.

Ime spletne trgovine	Platforma	Lastnik
App Store	iOS	Apple
BlackBerry World	BlackBerry OS, BlackBerry 10 OS	BlackBerry
Google Play	Android	Google
Nokia Store	Symbian, MeeGo, Maemo, S40	Nokia
Samsung Apps	Android Brada	Samsung, Handmark
Windows Phone Store	Windows Phone	Microsoft
Windows Store	Windows 8, Windows RT	Microsoft

Slika 3.1: Seznam spletnih trgovin s svojimi platformami in lastniki

Na začetku so se mobilne aplikacije uporabljale izključno za osnovne potrebe, kot so: prebiranje elektronske pošte, imenik, koledar, vremenske napovedi. Zaradi velikega povpraševanja in lahko dostopnega razvojnega okolja je prišlo do hitrega razvoja novih aplikacij, kot so: mobilne igre, sledenje storitvam, bančništvo. Prednost mobilnih aplikacij je tudi integracija različnih zunanjih komponent, kot so: GPS, kamera, merilnik pospeška, kompas, imenik, datoteke, omrežje, pomnilnik. Med uporabniki se je priljubljenost mobilnih aplikacij vse bolj povečevala. Zaradi velikega zanimanja za mobilne aplikacije se je pojavil problem kompatibilnosti med različnimi platformami. V primeru, da izdelujemo nativno aplikacijo, moramo za vsako platformo uporabljati svoj programski jezik. Ker bi bil pri enostavnejših programih strošek izdelave mobilne aplikacije, ki bi bila podprta na več platformah predrag, so se razvila različna ogrodja, kot na primer PhoneGap. Takim mobilnim aplikacijam pravimo Web mobile aplikacije.

3.2 Vrste mobilnih aplikacij

- **Nativna aplikacija**

Prednost nativne aplikacije je, da lahko deluje neodvisno od internetnega signala. Nativna aplikacija lahko deluje veliko hitreje z izkoriščanjem moči procesorja in tako dostopa do posebne strojne opreme, kot je GPS. V nekaterih pametnih telefonih lahko deluje samostojno in nadzoruje ostale aplikacije. Njena pomanjklivost je zahteven razvoj. Vsako aplikacijo je potrebno razviti v posebnem programskem jeziku, ki ga zahteva pripadajoča platforma. Dandanes je ključnega pomena, da je mobilna aplikacija podprta na vseh platformah. Posledično potrebujemo za razvoj nekoliko obširnejše nativne aplikacije številčnejši tim razvijalcev, kar povečuje stroške razvoja.

- **Spletna (Web) aplikacija**

Web aplikacija je napisana v jeziku, ki je znan vsem platformam. Večinoma se uporablja HTML, CSS ali JavaScript jezik. Funkcija dostopa

do strežnika z določenimi parametri. S pomočjo klica določene funkcije z določenimi parametri na strežniku, ji le-ta vrne željeni rezultat, ki se kasneje v primerni obliki prikaže v mobilni aplikaciji. Njena slabost je, da brez internetne povezave ne more delovati, saj ne more dostopati do funkcij, ki se nahajajo na strežniku. Prav tako deluje počasneje, saj je njena hitrost odvisna od hitrosti internetne povezave.

3.3 Vrste dotikov zaslona in njihova uporaba

- **tap**: sprožilec ob pritisku,
- **taphold**: sprožilec, ki se ob pritisku in držanju po določenem času sproži,
- **swipe**: sprožilec, ko povlečemo v katero koli smer vsaj 30 pikslov,
- **swipeleft**: sprožilec, ko povlečemo v levo vsaj 30 pikslov,
- **swiperight**: sprožilec, ko povlečemo v desno vsaj 30 pikslov,
- **scrollstart**: sprožilec, ko začnemo pomikati drsnik,
- **scrollstop**: sprožilec, ko prenehamo pomikati drsnik.

Poglavje 4

Razvoj mobilne aplikacije

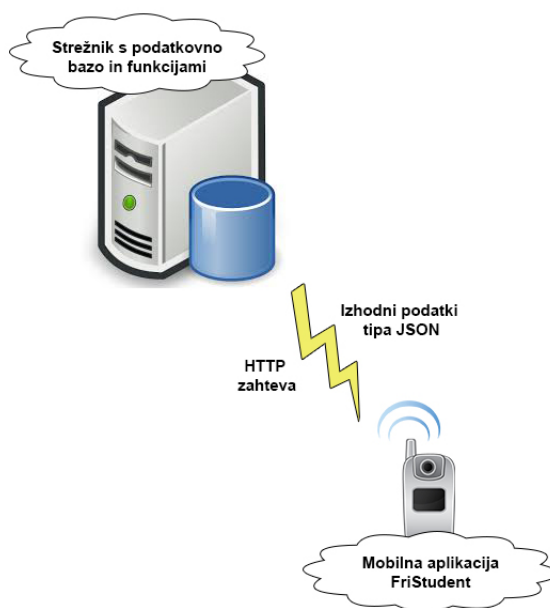
4.1 Mobilna aplikacija FriStudent

Vsaka izobraževalna ustanova, fakulteta, srednja šola in osnovna šola, potrebuje svojo spletno učilnico, v kateri se objavljajo rezultati, prijave na izpite, obvestila, pregled ocen, itd. V veliki večini imajo šole oz. fakultete to že urejeno, vendar do informacij dostopajo preko spletnega brskalnika. Ker je v zadnjem času uporaba mobilnih aplikacij izredno naraščajoča, se poraja potreba po razvoju mobilne aplikacije, ki bi podpirala vse osnovne funkcionalnosti, ki jih potrebujejo šole oz. fakultete. V svojem diplomskem delu sem razvil mobilno aplikacijo, namenjeno fakulteti za računalništvo in informatiko. Aplikacija je razvita na ogrodju PhoneGap s pomočjo orodja Xcode.

4.2 Arhitekture mobilne aplikacije

Aplikacija potrebuje za delovanje dostop do interneta. Povezuje se na strežnik na katerem se nahaja MySQL podatkovna baza z vsemi podatki o posameznih študentih in profesorjih. Ko mobilna aplikacija zahteva določeno funkcijo, ustvari Ajax klic na strežnik s potrebnimi podatki. Funkcija, ki se nahaja na strežniku se po potrebi poveže na podatkovno bazo, ki je zaščitena z uporabniškim imenom in geslom. Če je uporabniško ime in geslo pravilno,

nam podatkovna baza vrne rezultat poizvedbe. Rezultat poizvedbe nam funkcija pošlje v JSON obliki. Ta se nam nato ustrezno prikaže v mobilni aplikaciji, v kolikor med prenosom podatkov ne pride do napake. Opisana povezava med strežnikom in mobilno aplikacijo je prikazana na Sliki 4.1.



Slika 4.1: Arhitektura mobilne aplikacije FriStudent

4.3 Kreiranje podatkovne baze

Za kreiranje podatkovne baze je bilo uporabljeno orodje phpMyAdmin, ki nam s pomočjo grafičnega vmesnika omogoča kreiranje podatkovnih baz z njihovimi tabelami. S pomočjo programa se lahko testira različne poizvedbe, pregleduje podatke, vnaša nove ali briše obstoječe podatke, ter popravlja njihovo strukturo.

4.4 Struktura podatkovne baze

Tabele:

- user (uporabnik)
- user_group (uporabnik_skupina)
- subject (predmet)
- user_subject (uporabnik_predmet)
- examination (izpitni_rok)
- examination_check (prijava_izpitni_rok)
- examination_score (prijava_ocena)
- notification (obvestilo)
- notification_check (obvestilo_pregled)
- survey (anketa)
- survey_question (anketa_vprašanje)
- survey_option (anketa_možnosti)
- user_survey_answer (uporabnik_anketa_odgovor)

Glavni del podatkovne baze je tabela uporabnik (user). V tabeli so shranjeni vsi študentje in profesorji z njihovimi osnovnimi podatki. Tabela uporabnik je vezana na tabelo uporabnik_skupina (user_group), ki ločuje študente od profesorjev.

Izrednega pomena je tudi seznam predmetov (subject). Vsak predmet je povezan z uporabnikom (user). Ta povezava pove, kateri profesor predava

določeni predmet. Ker je potrebno določiti do katerih predmetov ima posamezen študent dostop in ker ima za posamezen predmet lahko dostop več študentov, je narejena vmesna tabela uporabnik_predmet (`user_subject`). V njej hranimo seznam študentov s pripadajočimi predmeti. Vsak predmet ima svoje izpitne roke (`examination`). Za vodenje prijave študenta na posamezen izpitni rok se uporablja tabela izpitni_rok_prijava (`examination_chek`). Rezultat študenta, ki se je prijavil na določen izpitni rok hranimo v tabeli prijava_ocena (`examination_score`).

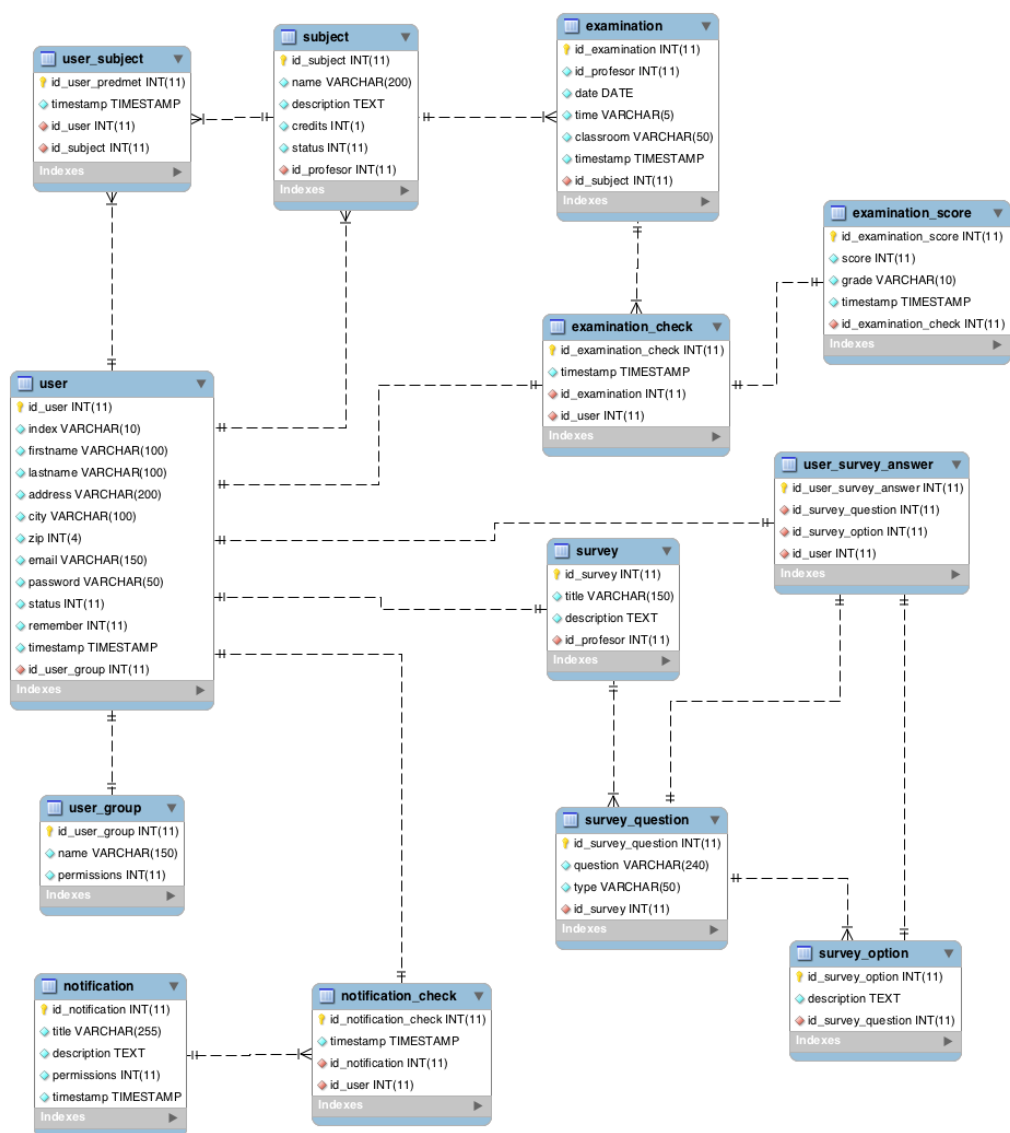
Aplikacija omogoča tudi pregledovanje obvestil (`notification`). Ko profesor objavi novo obvestilo, se ta shrani v tabelo obvestil. Za beleženje prebranih obvestil posameznega študenta je ustvarjena tabela obvestila_pregled (`notification_check`).

Profesor ima poleg objavljana obvestil tudi možnost ustvarjanja anket (`survey`). Vsaka anketa ima seznam vprašanj (`survey_question`), vsako vprašanje ima tudi možne odgovore (`survey_option`). Po objavi ankete jo lahko posamezen študent izpolnjuje. Odgovori na ankete se shranjujejo v tabelo uporabnik_survey_odgovor (`user_survey_asnwer`).

4.5 Konceptualni model podatkovne baze

Relacijski model predstavlja strukturo podatkovne baze. Uporabljamo ga za predstavitev konceptualnega nivoja. Relacijski model opisuje okolje s termini kot so entiteta, atribut, relacija, ključ.

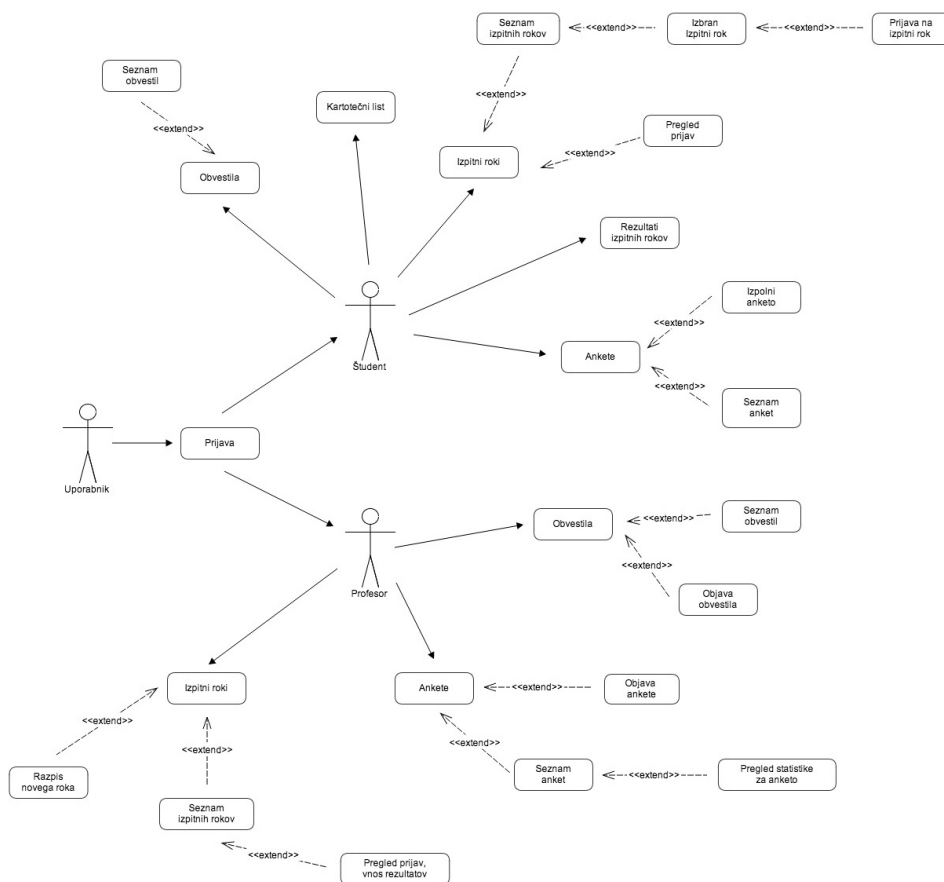
Središče strukture podatkovne baze je entita `user`. Ta povezuje predmete, prijave na izpitne roke, rezultate izpitnih rokov, obvestila in ankete. Več o sami strukturi podatkovne baze je bilo omenjeno že v prejšnjem poglavju. Konceptualni model za mobilno aplikacijo je razviden iz Slike 4.2.



Slika 4.2: Konceptualni model podatkovne baze FriStudent

4.6 Diagram primera uporabe

Možne interakcije uporabnika z zunanji sistemi lahko vizualno prikažemo z diagramom primera uporabe. Diagram orisuje na kakšen način lahko uporabnik komunicira s sistemom za izpolnitev ciljev. Diagram primera uporabe za mobilno aplikacijo je razviden iz Slike 4.3.



Slika 4.3: Diagram primera uporabe za mobilno aplikacijo FriStudent

Poglavje 5

Implementirane funkcionalnosti mobilne aplikacije

5.1 Prijava v sistem

Prijava je temeljni del aplikacije, ki ločuje prijavljenje študentov in profesorjev. Uporabnik se s svojim uporabniškim imenom in geslom prijavi v sistem. Prikaz prijave v sistem je razviden iz Slike 5.1. Glede na pravice, ki so zapisane v tabeli `user_group`, sistem ločim ali gre za študenta ali profesorja. Prijava ima tudi možnost shranjevanja podatkov v poljih z možnostjo “Zapomni si me”. V kolikor je omenjena možnost izbrana, se nam ob ponovnem zagonu aplikacije avtomatično izpolnijo predhodno vpisani podatki. Torej nam ni potrebno ponovno vpisovati uporabniškega imena in gesla. Shranjevanje deluje s pomočjo pomnilnika mobilne naprave. Uporabniško ime in geslo se shranita v pomnilnik telefona.

Shranjevanje v pomnilnik mobilne naprave

Vsaka mobilna naprava ima tudi svoj pomnilnik. S pomočjo JavaScript metode `localStorage` lahko enostavno dostopamo do pomnilnika in sicer po naslednjih postopkih.

V kolikor želimo shranjevati, brati ali brisati vrednost spremenljivke:

```
window.localStorage.setItem(\key", \value");  
var value = window.localStorage.getItem(\key");  
window.localStorage.removeItem(\key");
```

V kolikor želimo shranjevati, brati ali brisati vrednosti objekta:

```
window.localStorage.setObject(\key", object);  
Var value = window.localStorage.getObject(\key");  
window.localStorage.removeObject(\key");
```

Brisanje celotnega pomnilnika:

```
window.localStorage.clear();
```



Slika 5.1: Prikaz prijave v mobilno aplikacijo FriStudent

5.2 Nadzorna plošča

V kolikor je validacija pravilna nas aplikacija preusmeri v nadzorno ploščo. Nadzorna plošča se med študenti in profesorji nekoliko razlikuje. Razliko med nadzornima ploščama prikazujeta Sliki 5.2 in 5.3. Namen nadzorne plošče je združevanje vseh funkcionalnosti, ki so potrebne za študenta ali profesorja. Desno zgoraj se nahaja gumb za urejanje našega profila, kjer je mogoče spreminjati elektronski naslov in geslo. Zaradi varnosti je pri zamenjavi gesla potrebno vpisati tudi staro geslo.

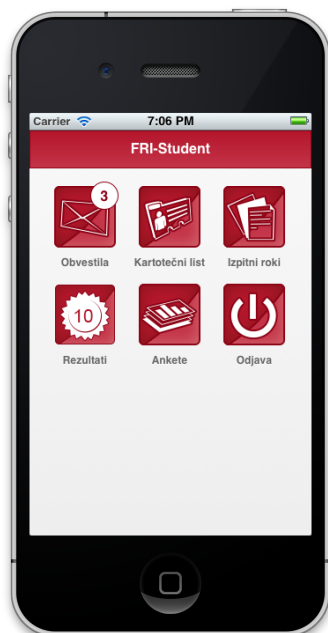
Funkcionalnosti za študente:

- obvestila,
- kartotečni list,
- izpitni roki,
- rezultati,
- ankete.

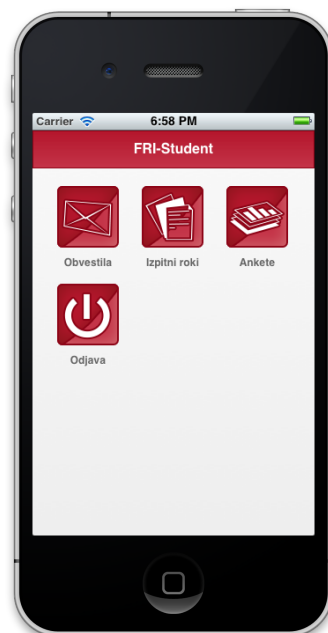
Funkcionalnosti za profesorje:

- obvestila,
- izpitni roki,
- ankete.

Gumb "Obvestila" ima dodatno funkcionalnost, ki študentu omogoča prikaz števila neprebranih obvestil. Ta se spremeni, v kolikor študent prebere neprebrano obvestilo.



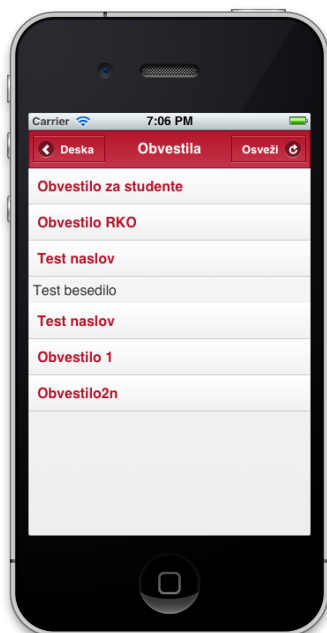
Slika 5.2: Prikaz nadzorne plošče za študenta v mobilni aplikaciji FriStudent



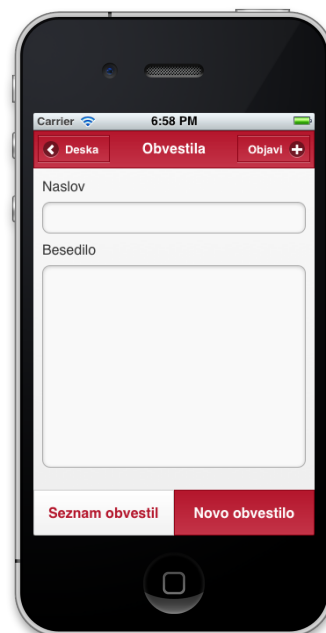
Slika 5.3: Prikaz nadzorne plošče za profesorja v mobilni aplikaciji FriStudent

5.3 Obvestila

Študnetu se na nadzorni plošči ob pritisku na gumb "Obvestila" prikaže seznam objavljenih obvestil, ki ga prikazuje Slika 5.4. S klikom na naziv, se mu ta odpre. Prikaže se mu celoten opis obvestila. Na seznamu obvestil so obvestila, ki še niso bila prebrana označena z klicajem. Profesor ima možnost nove objave obvestila. To stori z vpisom naziva in besedila obvestila, kot prikazuje Slika 5.5. Po objavi novega obvestila se ta pojavi na seznamu študentovih obvestil. Ker se obvestila ne osvežujejo v realnem času se desno zgoraj nahaja gumb "Osveži". Ob pritisku na gumb se seznam obvestil posodobi, v kolikor je prišlo do novih obvestil.



Slika 5.4: Prikaz seznama obvestil v mobilni aplikaciji FriStudent



Slika 5.5: Prikaz dodajanja obvestil v mobilni aplikaciji FriStudent

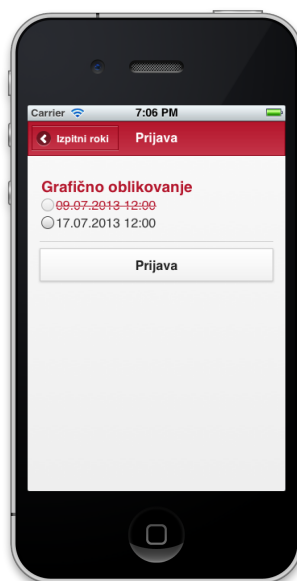
5.4 Izpitni roki

Funkcionalnost izpitni roki študentu omogoča prikaz seznama predmetov na katere se lahko prijavi. Prikaz seznama predmetov je razviden iz Slike 5.6. Po izboru predmeta se mu pojavi seznam izpitnih rokov, ki ga prikazuje Slika 5.7. Če sta datum in ura prečrtana in obarvana z rdečo barvo pomeni, da je na ta izpitni rok študent že prijavljen. Če izpitnega roka ne moramo označiti pomeni, da je čas za prijavo na izpitni rok že potekel. Študent se lahko na izpitni rok prijavi najkasneje dan pred izvedbo. V kolikor študent označi željeni izpitni rok, se nanj lahko prijavi. Po uspešni prijavi lahko preide na seznam prijav, kjer so izpitni roki navedeni s krajom in časom izvedbe. Na seznamu prijav ima študent možnost odjave od izpitnega roka. To stori z drsom v desno preko željenega izpitnega roka.

Profesor ima možnost razpisa novega izpitnega roka. To opravi tako, da preide na podmeni razpis roka. Izbrati mora predmet, za katerega želi razpisati nov izpitni rok, datum, uro in predavalnico. Ta postopek prikazuje Slika 5.8. Pri tem lahko izbira samo med predmeti, ki jih sam predava. Po objavi novega izpitnega roka, se ta študentom pojavi na seznamu izpitnih rokov.



Slika 5.6: Prikaz seznama izpitnih rokov v mobilno aplikacijo FriStudent



Slika 5.7: Prikaz prijave na izpitni rok v mobilni aplikaciji FriStudent

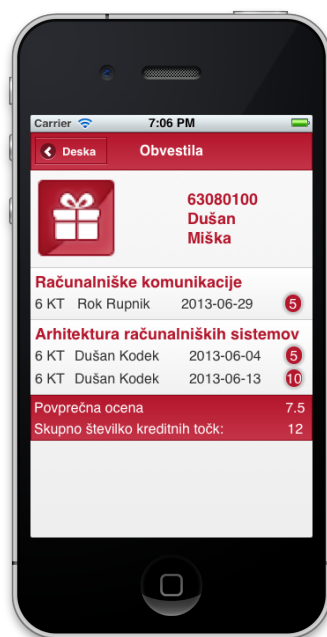


Slika 5.8: Prikaz dodajanja izpitnega roka v mobilni aplikaciji FriStudent

5.5 Kartotečni list

Kartotečni list se prikaže samo študentom. V njem so zajeti vsi podatki o opravljenih izpitih in podatki študenta (ime, priimek in indeks). Pri vsakem opravljenem izpitu se prikaže oceno, datum opravljanja, ime profesorja, ki predmet poučuje ter število kreditnih točk. Poleg tega sta navedena tudi

povprečje ocen vseh opravljenih izpitov in seštevke kreditnih točk. Prikaz kartotečnega lista je razviden iz Slike 5.9.



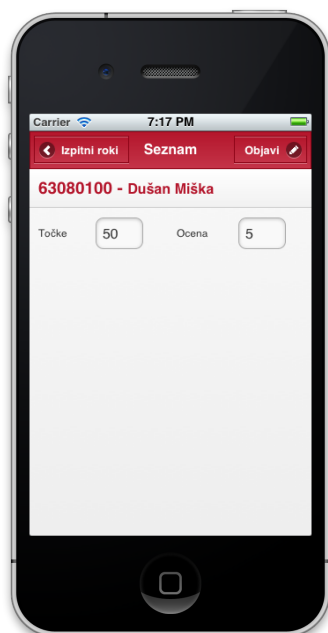
Slika 5.9: Prikaz kartotečnega lista v mobilni aplikaciji FriStudent

5.6 Rezultati izpitnega roka

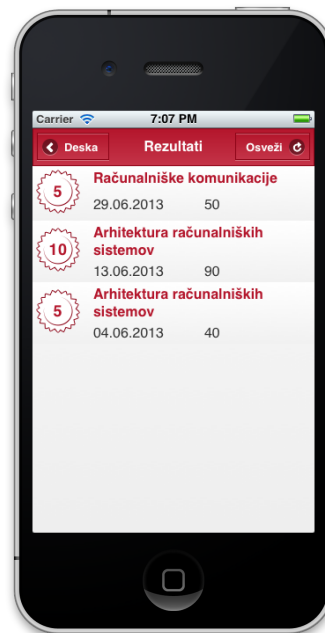
Profesor na seznamu študentov prijavljenih na izpitni rok vnese število doseženih točk in oceno za posameznega študenta. Prikaz vnašanja je razviden iz Slike 5.10. Po opravljenem vnosu profesor s klikom na gumb "Objavi", ki se nahaja desno zgoraj, objavi podatke.

Študentu se po profesorjevi objavi dosežene točke in ocena opravljenega izpitnega roka pojavita v funkcionalnosti "Rezultati". Funkcionalnost rezultati vsebuje seznam vseh opravljenih izpitnih rokov, ki ga prikazuje Slika 5.11. Pri tem se omenjeni rezultati študenti prikažejo tako na seznamu rezultatov,

kot tudi v kartotečnem listu.



Slika 5.10: Prikaz vnosa doseženih točk in ocene v mobilni aplikaciji FriStudent

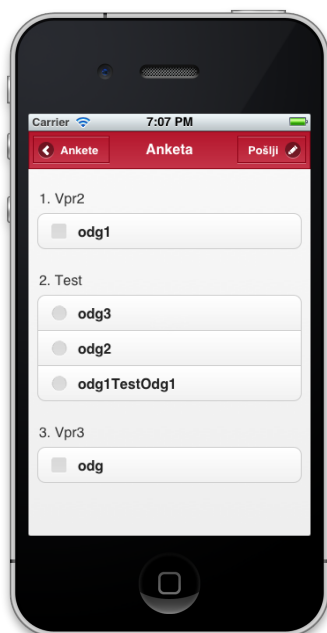


Slika 5.11: Prikaz rezultatov opravljenih izpitnih rokov v mobilni aplikaciji FriStudent

5.7 Anketa

V funkcionalnosti "Anketa" je podana možnost izpolnjevanja oz. objavljanja novih anket. Profesor mora najprej ustvariti novo anketo. Najprej mora vpisati naziv ankete, nato pa poljubno dodaja možna vprašanja. Vsakemu vprašanju lahko doda poljubno število odgovorov. Odgovori so lahko tipa "več možnih odgovorov" (checkbox) ali pa "le en možen odgovor" (radio-button). Vsakemu odgovoru doda tudi oznako. Po uspešno ustvarjeni in objavljeni anketi, se anketa objavi v študentov seznam. Študent iz seznama anket izbere željeno anketo in jo izpolni. Po objavljenih študentovih odgovorih lahko profesor za vsako izmed njegovih objavljenih anket pregleda

rezultate.



Slika 5.12: Prikaz izpolnjevanja ankete v mobilni aplikaciji FriStudent



Slika 5.13: Prikaz pregleda rezultatov ankete v mobilni aplikaciji FriStudent

5.8 Metoda Ajax

Za vsak klic posamezne metode, ki se nahaja na strežniku, je potrebno prilagojeno prikazovanje podatkov. Za to sem razvil prilagojeno ajax metodo, ki je splošna za vse ajax zahteve. Metoda sprejme tri vrste parametrov:

- `functionName` nam pove kako naj se pridobljeni rezultati poizvedbe prikažejo na zaslon. V parameter vnesemo ime funkcije. Izbrano ime metode se nato izvede s podanimi parametri.
- `location` izberemo glede na zahtevan klic. Ker imamo klice ločene na študente in profesorje, z lokacijo določimo ali naj se izvede funkcija za

študenta ali za profesorja. Poleg tega navedemo tudi ime datoteke v kateri se nahaja željena funkcija.

- data je seznam parametrov, ki so potrebni za pridobitev željenih podatkov.

```
function ajax(functionName, location, data) {
    ajaxLoader('show');

    var request = $.ajax({
        url: "http://kewlzone.si/test/controller/"+location,
        type: "POST",
        data: {data : data},
        dataType: "json"
    });

    request.done(function(msg) {
        ajaxLoader('hide');

        runFunction(functionName, [msg]);
    });

    request.fail(function(jqXHR, textStatus) {
        ajaxLoader('hide');
        alert(textStatus+ " : Morda vam je potekla seja.");
    });
}
```

Poglavje 6

Sklepne ugotovitve

Namen diplomske naloge je razvoj mobilne aplikacije, ki bo podpirala osnovne funkcionalnosti, ki jih potrebuje vsak študent. Razvoj mobilne aplikacije je temeljil na ogrodju PhoneGap. Ogrodje se je izkazalo za dobro izbiro, saj je prenos mobilne aplikacije na druge platforme enostavnejši. Edina slabost je nekoliko počasnejša odzivnost aplikacije. Razvoj in simulacija delovanja aplikacije sta potekala v orodju Xcode. Simulacija je potekala na dveh operacijskih sistemih, in sicer iOS 5 in iOS6.1.

Mobilna aplikacija ne izvaja zahtevnejših operacij, ampak le osnovne poizvedbe, in sicer branje, dodajanje in urejanje podatkov iz podatkovne baze. Fakultete že imajo spletno aplikacijo elektronski študent, ki omogoča vodenje in pregled študentovih ocen, prijavo na izpite, pregled obvestil, itd. Nadgradnja obstoječih funkcionalnosti spletne aplikacije bi omogočala enostavnejši razvoj mobilne aplikacije. Potrebna je le implementacija izgleda in prikaza podatkov. V primerjavi s samostojnim razvojem vseh funkcionalnosti, je nadgradnja enostavnejša, cenejša in hitrejša. Sistem bi sam zaznaval ali gre za dostop preko spletne ali mobilne aplikacije. S takšno nadgradnjo se izognemo tudi problemom, ki bi jih povzročila dodatna implementacija podatkovne baze in podvajanj že obstoječih podatkov, ki se uporabljajo na samem spletu.

Prednost mobilne aplikacije je predvsem večja dostopnost. Pametni telefoni so vedno bolj zmogljivejši in lažji od prenosnih računalnikov. Izdelava takšne mobilne aplikacije pripomore k temu, da bi študentje imeli možnost spremljanja in uporabljanja elektronskega študenta tudi brez osebnega računalnika. Potrebna sta le pametni telefon in internetna povezava.

Literatura

- [1] (2013) Mobilna aplikacija. Dostopno na:
http://en.wikipedia.org/wiki/Mobile_app

- [2] (2013) Razvoj mobilne aplikacije. Dostopno na:
http://en.wikipedia.org/wiki/Mobile_application_development

- [3] (2013) Razlika med nativo in spletno mobilno aplikacijo. Dostopno na:
<http://www.insivia.com/what-is-the-difference-between-a-native-mobile-app-vs-web-app/>

- [4] (2013) HTML. Dostopno na:
<https://en.wikipedia.org/wiki/HTML>

- [5] (2013) HTML. Dostopno na:
<http://www.webopedia.com/TERM/H/HTML.html>

- [6] (2013) Definicija HTML. Dostopno na:
<http://searchsoa.techtarget.com/definition/HTML>

- [7] (2013) PhoneGap. Dostopno na:
<http://en.wikipedia.org/wiki/PhoneGap>

- [8] (2013) Kako začeti z razvojem okolja PhoneGap. Dostopno na:
<http://phonegap.com/blog/build/getting-started-with-phonegap-and-phonegap-build/>

-
- [9] (2013) Delovanje PhoneGap. Dostopno na:
<http://stackoverflow.com/questions/9443548/how-phonegap-really-works>
- [10] (2013) PHP. Dostopno na:
<https://en.wikipedia.org/wiki/PHP>
- [11] (2013) JavaScript. Dostopno na:
<https://en.wikipedia.org/wiki/JavaScript>
- [12] (2013) Interpretativni programski jezik. Dostopno na:
http://en.wikipedia.org/wiki/Interpreted_language
- [13] (2013) CSS. Dostopno na:
http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [14] (2013) Kaj je CSS. Dostopno na:
<http://webdesign.about.com/od/beginningcss/a/aa021607.htm>
- [15] (2013) jQuery mobile. Dostopno na:
<http://jquerymobile.com/demos/1.2.0/docs/api/events.html>
- [16] (2013) Razlika med jQuery in jQuery mobile. Dostopno na:
<http://stackoverflow.com/questions/10391856/difference-between-jquery-and-jquery-mobile>
- [17] (2013) Kaj je AJAX. Dostopno na:
http://www.tutorialspoint.com/ajax/what_is_ajax.htm
- [18] (2013) AJAX programiranje. Dostopno na:
[http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [19] (2013) MySQL. Dostopno na:
<https://en.wikipedia.org/wiki/MySQL>
- [20] (2013) Xcode. Dostopno na:
<http://en.wikipedia.org/wiki/Xcode>

- [21] (2013) Kaj je Xcode. Dostopno na:
<http://answers.yahoo.com/question/index?qid=20120515142808AAFigP3>
- [22] (2013) PhoneGap podpora APIjev. Dostopno na:
<http://phonegap.com/about/feature/>
- [23] (2013) Uporaba mobilnega pomnilnika v okolju PhoneGap. Dostopno na:
http://docs.phonegap.com/en/1.2.0/phonegap_storage_storage.md.html#localStorage