Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Milenko Ninić

Mobilna aplikacija za oblikovanje skupin za učenje

diplomsko delo

visokošolski strokovni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2014

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: 00444 / 2013
Datum: 12.4.2013

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

Kandidat: **MILENKO NINIĆ**

Naslov: **MOBILNA APLIKACIJA ZA OBLIKOVANJE SKUPIN ZA UČENJE**
MOBILE APPLICATION FOR FORMING LEARNING TEAMS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Študentje na fakulteti se večinoma učijo v parih ali celo skupinah, ko se pripravljajo na kolokvije ali izpite. Zasnujte mobilno aplikacijo, ki študentom fakultete omogoča oblikovanje parov ali skupin za učenje. Za razvoj mobilne apliacije uporabite tehnologijo BlackBerry 10 Native SDK. Spletne storitve pa razvijte v tehnologiji .NET v oblaku Micrsoft Azure.

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 Milenko Ninić, z vpisno številko **63080375**, sem avtor diplomskega dela z naslovom:

Mobilna aplikacija za oblikovanje skupin za učenje

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 17. marec 2014 Podpis avtorja:

Zahvaljujem se mentorju, doc. dr. Roku Rupniku, za mentorstvo in pomoč pri izdelavi diplomske naloge. Posebna zahvala gre moji družini, ki me je vedno podpirala in bodrila v času mojega študija.

Kazalo

Seznam uporabljenih kratic

RIM - Research In Motion

OS – Operating system

IDE - Integrated Development Environment

SDK - Software development kit

GUI - Graphical user interface

QML - Qt Modeling Language

SaaS - Software as a Service

PaaS - Platform as a Service

IaaS - Infrastructure as a Service

MIDP - Mobile Information Device Profile

WAP - Wireless Application Protocol

BES - BlackBerry Enterprise Server

BIS - BlackBerry Internet Service

IMAP - Internet Message Access Protocol

MVC - Model View Controller

REST - Representational state transfer

CRUD - Create, Retrieve, Update, Delete

SQL - Structured Query Language

Povzetek

Diplomska naloga temelji na izdelavi mobilne aplikacije mSTUDY, ki omogoča študentom organizacijo skupinskega učenja. Namen mobilne aplikacije je, da ima vsak študent dostop do le-te ter da ima vpogled v vse objave in posledično omogočeno prijavo v želeno učno skupino. Omogočeno je ustvarjanje različnih objav, ki temeljijo na tematiki študija, ter kreiranje elektronskih vabil.

V spodnjih poglavjih so predstavljene vse tehnologije in orodja, ki smo jih uporabili pri naši rešitvi. Aplikacija je zgrajena na operacijskem sistemu Blackberry 10 z razvojnim paketom BlackBerry 10 Native SDK. V nadaljevanju so predstavljene tudi rešitve spletne storitve RESTful, zgrajene z .NET tehnologijo in migracijo le-te v oblačno platformo Windows Azure. Opisana je tudi realizacija podatkovne baze, ki se prav tako nahaja na Windows Azure platformi. Predstavljene so zahteve za aplikacije, razložili smo celotno zasnovo aplikacije in njeno arhitekturo.

Ključne besede: BlackBerry, mobilna aplikacija, spletna storitev, podatkovna baza, Qt, QML, Windows Azure, ASP.NET

Abstract

The thesis is based on the development of mobile application mSTUDY that allows students to organize a learning in the groups of students. The purpose of mobile application is that every student has access to it and to have also access to all posts and consequently able to log in to the desired learning group. It is possible to create different posts based on the topic of study and create e-mail invitations.

The section below present all of technologies and tools that we used in our solution. The application is build on operation system BlackBerry with development package BlackBerry 10 Native SDK. The following are presented solutions RESTful web service build with .NET technology and the migration of it in cloud platform Windows Azure. It is also described implementation of the database, which is also located on the Windows Azure platform. Requirements of the application are presented and we explained the overall design of the application and its architecture.

Keywords: BlackBerry, mobile application, web service, database, Qt, QML, Windows Azure, ASP.NET

Poglavje 1

Uvod

Mobilna telefonija je v zadnjem desetletju postala nepogrešljiv del našega vsakdana. Z razvojem pametnih telefonov se je mobilna tehnologija še razvila in je v nenehni razvojni fazi s stremljenjem po čim lažji uporabi. Današnji uporabniki smo vse bolj navezani na mobilne telefone in njihovo funkcionalnost. Iz dneva v dan se razvijajo različne aplikacije in programi za lažje in učinkovito delo ali študij.

Glede na vse bolj priljubljeno in razširjeno mobilno tehnologijo smo se odločili, da tudi mi prispevamo k razvoju Blackbarry aplikacij za študente. Razvili smo mobilno aplikacijo mSTUDY, ki bo olajšala študij študentom pri različnem učenju ali nudenju tutorstva.

Diplomsko delo je sestavljeno iz teoretičnega in praktičnega dela. Podrobno je predstavljen Blackberry operacijski sistem, na katerem je aplikacija zgrajena. Prav tako smo predstavili vse tehnologije razvoja ter orodja, ki so bila uporabljena ob razvoju mobilne aplikacije. Predstavljena je programska rešitev, kjer so opisani koraki raziskovanja, oblikovanja in sestave aplikacije. Predstavljena je torej zasnova aplikacije z vsemi koraki razvoja in predstavitve delovanja.

Namen diplomskega dela je olajšanje študija študentom. Mobilno aplikacijo lahko uporabljajo študentje na začetku svojega študija, saj bi s tem navezali stike tudi z drugimi študenti, ki so že dobili znanje z določenega področja.

Cilj je določeno skupino ljudi, ki ima skupne interese, združiti v učno skupino, kjer bi lahko prispevali s svojim znanjem, idejami in predlogi. Tako bi lahko spoznali več študentov z različnih področjih, prihranili bi si čas, saj bi z razlagami drug drugemu pomagali do bolj razumljivih rešitev, povečali motiviranost s spodbujanjem. Vse to lahko omogočimo z mobilno aplikacijo, ki bo olajšala in organizirala dogodke med študenti, ki si želijo znanja.

Poglavje 2

BlackBerry

BlackBerry podjetje, prej znano kot RIM (Research In Motion), je kanadsko podjetje, ki se ukvarja s telekomunikacijami in brezžično opremo. Leta 1984 ga je ustanovil Mike Laziridis. BlackBerry je najbolj znan kot razvijalec pametnih telefonov in tabličnih računalnikov blagovne znamke BlackBerry. Je vodilno podjetje na področju brezžičnih inovacij. BlackBerry je leta 1999 je naredil revolucijo na področju mobilne industrije.

Prva naprava BlackBerry 850 Pager se je pojavila leta 1999 in je znala uporabljati brezžično omrežje za spletno pošto. Ta prednost je bila zelo pomembna, saj je do konca leta 2007 podjetje pridobilo kar 32 milijonov uporabnikov. Najnovejše BlackBerry naprave so Z30, Z10, Q10 in Q5. Uporabniški vmesnik je odvisen od modela naprave. Večina naprav ima izrazito fizično tipkovnico QWERTY, medtem ko se novejše generacije sklicujejo na zaslon na dotik in virtualno tipkovnico [5].

V začetku leta 2013 je podjetje predstavilo novo tehnologijo pametnih telefonov z novim operacijskim sistemom BlackBerry 10. Nova platforma BlackBerry 10 združuje BlackBerry OS 6 in OS 7 ter QNX.

Da bi povečali razvoj in rast aplikacij na novi platformi BlackBerry 10, je na letni razvojni konferenci BlackBerry World razdelil več kot tisoč prototipnih telefonov med razvijalce, cilj tega pa je bil, da bi pospešili razvoj programske opreme za novo generacijo telefonov. Razvijalci so dobili telefone, ki so se

imenovali BlackBerry 10 Dev Alpha. Na napravi smo tudi sami razvijali našo aplikacijo z razvojnim paketom SDK 10.1 [6].

Slika 2.1: Logotip podjetja BlackBerry.

2.1 BlackBerry operacijski sistem

BlackBerry je razvil za svojo linijo pametnih naprav operacijski sistem, poimenovan BlackBerry OS. Prva verzija operacijskega sistema je nastala leta 1999.

Platforma BlackBerry je najbolj znana po zanesljivosti, večopravilni zmogljivosti, interoperabilnosti, podpori specializiranih vhodnih naprav in svoji podpori elektronski pošti preko MIDP 1.0 in v zadnjem času po MIDP 2.0. Omogoča popolno brezžično aktiviranje in sinhronizacijo s storitvami, kot so: Microsoft Exchange, Lotus, Domino ali GroupWise, elektronska pošta, koledar, opravila, beležke in stiki, kadar se uporablja z BlackBerry Enterprise Serverjem. Funkcija, ki je postala ključni del OS, je BlackBerry Messenger, ki omogoča uporabnikom neposredno komunikacijo v realnem času preko sporočil. Najnovejši BlackBerry 10 OS temelji na QNX in se bo uporabljal na najnovejših BlackBerry 10 napravah. Operacijski sistem podpira tudi WAP 1.2 [15].

2.1.1 Različice operacijskega sistema BlackBerry

Leta 1999 je bila izdana različica 1.0, kasneje so sledile novejše verzije. Vsaka se je izpopolnjevala v varnosti in tudi v zmogljivosti. Verzija 6.0 je prinesla največ sprememb, saj je bila tovrstna verzija posodobljena tudi na področju zabave in ne le na poslovnem področju [14].

Različice	Leto izdaje
1.0	januar 1999
3.6	marec 2002
5.0	avgust 2008
6.0	april 2010
7.0	avgust 2011
7.1	januar 2012
10.0	januar 2013

Tabela 2.1: Različice operacijskega sistema BlackBerry.

2.1.2 BlackBerry operacijski sistem 10

BlackBerry 10 je operacijski sistem, ki temelji na QNX operacijskem sistemu. Operacijski sistem cilja na poslovne uporabnike. Kot prvi med telefoni že v osnovi podpira koncept BYOD in omogoča ločene nastavitve poslovnega okolja in okolja za domačo rabo na isti napravi. Tehnologija se imenuje BlackBerry Balance in deluje v kombinaciji s strežniškim sistemom BlackBerry Enterprise Service 10.

Uporabniki lahko na poslovnem delu telefona nadzorujejo delovanje mobilne naprave, nabor programov, varnostne nastavitve, medtem ko lahko uporabniki pri domači rabi telefona v ločenem delu nastavijo lastne nastavitve in programe brez nevarnosti za poslovni del. Preklop med obema okoljema je mogoč zgolj z eno kretnjo, sočasno pa je možno poganjati na telefonu poslovne in domače programe [9].

Prednosti in slabosti BlackBerry 10 platforme

Prednost:

- Izrazite prednosti: elektronska pošta, sporočanje, varnost, trajanje baterije in upravljanje koledarja,

- varna platforma (vojaški-varnostni sistem enkripcije), saj je uporabljena v veliko vladnih organizacijah,
- možnost poganjanja Android aplikacij, kar omogoča emulator oziroma aplikacija Android Player,
- večopravilnost,
- stabilna in zanesljiva platforma, uporablja samo svojo strojno opremo,
- storitev, ki omogoča ločevanje poslovnih in osebnih aplikacij ter podatkov,
- omogoča enostavno upravljanje aplikacij za zaposlene, saj lahko skrbniki prek oddaljenega dostopa namestijo aplikacije na telefonih zaposlenih oz. uporabnikom priporočijo namestitev aplikacij,
- qwerty tipkovnica, ki omogoča hitro tipkanje ter izjemno integracijo z elektronsko pošto [7] [8].

Slabosti:

- Izbor aplikacij je nekaj manjši kot pri drugih mobilnih platformah [7].

2.2 Zgradba BlackBerry platforme

BlackBerry uporablja večopravilni operacijski sistem BlackBerry 10, ki je napisan v programskem jeziku C++ in izhaja iz družine mobilnih operacijskih sistemov. Operacijski sistem podpira Java MIDP 1.0 specifikacijo, ki se uporablja za mobilne naprave in WAP 1.2. protokol, ki podpira WAP brskalnike. WAP brskalnik je podoben spletnim brskalnikom, ki se uporabljajo na računalnikih in nam omogočajo dostop do interneta preko mobilnih naprav.

Operacijski sistem se popolnoma brezžično sinhronizira s koledarjem, opravili, stiki, kontakti, izmenjavo elektronske pošte in zapiski. Izvajanje teh funkcij omogoča programska oprema BES, ki je del operacijskega sistema. Alternativa BES je BIS, ki omogoča uporabnikom dostop do interneta. S tem lahko uporabnik uporabi POP3, IMAP in Outlook spletni dostop do

elektronskega poštnega računa, ne da bi se sploh prijavil skozi BES. V tej storitvi se uporablja tehnologija potiska. BIS je storitev, ki jo dejansko uporablja BlackBerry, vendar je rezervirana preko našega ponudnika mobilnih storitev.

BlackBerry operacijski sistem vsebuje API vmesnike. Uporaba API razredov omogoča tretjim razvijalcem razvoj aplikacij. BlackBerry OS omogoča tudi rabo grafičnega uporabniškega vmesnika. Uporaba grafičnega uporabniškega vmesnika omogoča interakcijo z aplikacijami in programsko opremo na mobilnem telefonu [28].

APLIKACIJE
OPERACIJSKI SISTEM
JAVA VIRTUAL MACHINE
STROJNA OPREMA

Tabela 2.2: Arhitektura BlackBerry operacijskega sistema.

2.3 Svoboda izbire tehnologije razvoja

Razvijalci za BlackBerry platformo imajo prednost pred ostalimi razvijalci na drugih platformah, saj lahko obstoječo aplikacijo enostavno izvozijo na BlackBerry 10 platformo v nekaj hitrih korakih.

Razvijalci lahko izbirajo med programskimi in skriptnimi jeziki C/C++, AIR, HTML5 ali Qt za platformo BlackBerry 10 ali BlackBerry PlayBook [2].

2.3.1 C/C++ Native SDK

BlackBerry se dobro zaveda, da je okolje C/C++ ključnega pomena za številne igre. Zaradi tega so se odločili ustvariti čim večjo podporo, predvsem za razvoj obsežnih iger [2].

Slika 2.2: Logotip orodja Native SDK.

2.3.2 Java Android Runtime

Razvijalci, ki razvijajo aplikacije na platformi Android, lahko izvozijo aplikacije napisane v Javi na platformo BlackBerry z uporabo orodja Android Runtime, ki ga je razvil BlackBerry [2].

Slika 2.3: Logotip platforme Android.

2.3.3 ActionScript Adobe AIR

Razvijalci za Flash lahko razvijajo v ActionScriptu in Adobe Flex API z uporabo Adobe AIR SDK. Zagotavlja nekatere edinstvene komponente uporabniškega vmesnika. Ena od teh je poslušanje dogodkov, kar je specifično za naprave BlackBerry. Z uporabo Adobe Flash Builder API lahko aplikacija dostopa do funkcij, ki so edinstvene za mobilne naprave, kot na primer pospešek in geolokacijske informacije.

Slika 2.4: Logotip orodja Adobe AIR.

Adobe AIR Runtime 3.5 je sedaj na voljo razvijalcem, ki uporabljajo beta BlackBerry 10 OS različice 10.2 [3].

2.3.4 HTML5 WebWorks

Novo orodje BBUI.js omogoča razvijalcem enostavno integracijo modernizirane izkušnje BlackBerry v njegove aplikacije z uporabo HTML5. Spletni razvijalci lahko ustvarjajo aplikacije HTML5 s pomočjo RIM-ovega Webworks-a in jih testirajo z emulatorjem tinyHippos Ripple.

WebWorks SDK je odprtokodna platforma za razvoj, ki temelji na Apache Cordova in omogoča ustvarjanje aplikacij z uporabo spletnih tehnologij [4].

Slika 2.5: Logotip orodja HTML5 WebWorks.

Poglavje 3

Opis uporabljenih orodij in tehnologij

Opisali bomo tehnologije in orodja, ki smo jih uporabili pri razvoju aplikacije mSTUDY.

3.1 QNX Momentics IDE

Obsežno razvojno okolje QNX Momentics IDE temelji na Eclipse integriranem razvojnem okolju z inovativnim profiliranim orodjem z največjim vpogledom obnašanja sistema. To edinstveno orodje daje razvijalcem pogled v realnem času integracij, spominskih profilov in še več: omogoča krajše čase razhroščevanja in krajši čas za dostop do trga. Posebna orodja pomagajo razvijalcem kodo prenašati iz samega jedra na multi jedra sistemov in na varno optimizirano delovanje [1].

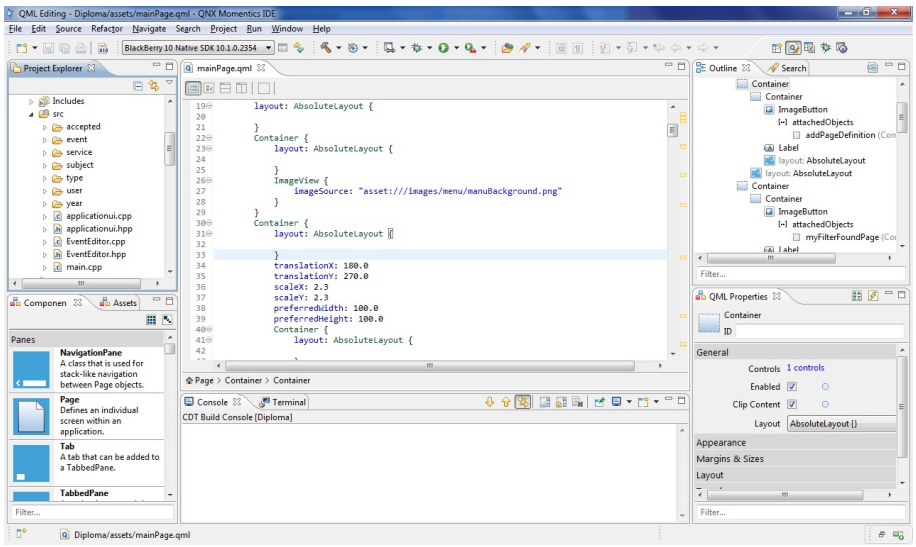
QNX Momentics IDE nudi vse razvojne in razhroščevalne funkcije, kot jih nudijo druga razvojna okolja, bazirana na Eclipse IDE, in vključuje edinstveno zmogljivost QNX, kot so večjedrno profiliranje [2].

Za razvoj v C in C++ nudi optimalno razvojno okolje. Razvijalci lahko delujejo bodisi v ukazni vrstici ali grafičnem IDE okolju in izkoristijo številne

funkcijske produktivnosti, kot so čarovnik, urejevalnik kode in fleksibilna podatkovna struktura.

Lastnosti razvojnega okolja:

- temelji na Eclipse IDE,
- podpira C in C++,
- večjederna podpora,
- integrirani kontrolni viri,
- razhroščevalni viri,
- sistemski graditelj,
- podatki o ciljnem sistemu,
- profil aplikacije,
- sistemski profil,
- analiza pomnilnika.



Slika 3.1: QNX Momentics razvojno okolje.

3.2 BlackBerry 10 Native SDK

BlackBerry 10 Native SDK je razvojni paket, ki nam omogoča izkoristiti vso moč BlackBerry 10 naprav. Združuje hitrost in stabilnost QNX Neutrino operacijskega sistema realnega časa z različnimi platformami in odprtokodnimi knjižnicami. SDK paket je namenjen izkušenim razvijalcem, ki si želijo izkoristiti popoln dostop do vsega, kar ponuja BlackBerry 10.

Native SDK nam ponuja nabor C/C++ aplikacij in tudi Cascades UI platformo, ki omogoča izgradnjo močne in dinamične aplikacije. Zagotavlja edinstvene knjižnice, ki omogočajo integracijo s platformo in z drugimi aplikacijami [25].

Razvojni paket je sestavljen iz dveh delov, ki pomagata pri hitrejši posodobitvi razvojnega modela in omogočata boljšo učinkovitost pri razvoju. Native SDK zagotavlja vse potrebne knjižnice, ki so nujne za izgradnjo aplikacije, medtem ko Momentics IDE za BlackBerry omogoča razvoj in testno okolje, ki vse skupaj združuje [26].

3.3 QNX

QNX je komercialni Unix-like operacijski sistem, ki je namenjen integriranemu tržnemu sistemu. Izdelek je bil prvotno razvit v zgodnjih 80-ih letih prejšnjega stoletja s strani kanadskega podjetja Quantum Software System. Kasneje je bil preimenovan v QNX Software System in naknadno pridružen BlackBerry-ju v letu 2010. QNX je bil eden od prvih komercialno uspešnih sistemov z mikrojedrom in se uporablja v različnih napravah, najzmogljivejših internetnih usmerjevalnikih, simulatorjih letenja, kontroli zračnega letenja, ladijskih navigacijskih sistemih, krmilnikih visoke hitrosti vlakov, distribucijskih sistemih, pametnih telefonih, mobilnih napravah, bolnišnicah, medicinski tehnologiji (EKG stroji, srčni monitorji), igralnih sistemih in še več.

3.4 Qt

Qt je prosto in odprto-kodno ogrodje za razvoj aplikacij, ki tečejo na več platformah, in se uporablja predvsem za razvoj grafičnih uporabniških vmesnikov. Qt uporablja standardni C++ in se pogosto uporablja za razvoj programskih aplikacij z grafičnim vmesnikom. Uporablja se ga tudi za razvoj aplikacij negrafičnih uporabniških vmesnikov s funkcijami, kot so upravljanje datotek, dostop do baze podatkov, razčlenjevanje XML in podpora omrežja. Qt je možno prek povezovalnih vmesnikov uporabljati tudi iz drugih programskih jezikov [10].

Koda 3.1: Primer Qt tehnologije.

```
#include <QApplication>
#include <QTextEdit>

int main(int argv, args)
{
        QApplication app(argv, args);
        QTextEdit textEdit;
        textEdit.show();

        return app.exec();
}
```

3.5 QML

Za izgradnjo uporabniškega vmesnika je Qt razvil specializirani označevalni jezik, imenovan QML. QML je deklarativni jezik, ki temelji na JavaScript, in je zasnovan tako, da je enostaven za uporabo. Tako kot standardni Qt tudi QML uporablja pojme, kot so objekti, spremenljivke, signali, reže, kar omogoča, da objekti komunicirajo med seboj. V QML je struktura določena na osnovi drevesne strukture objektov s posameznimi lastnostmi [12].

Za načrtovanje in razvoj v QML okolju je treba imeti nekaj predznanja o skriptnem jeziku JavaScript. Za obliko in stile v QML je poskrbljeno s strani tehnologije HTML in CSS [11].

Koda 3.2: Primer QML tehnologije.

```
import QtQuick 1.0
Rectangle{
        width:200
        height:200
        color:"blue"

        Image{
                source: "pics/logo.png"
                anchors.centerIn: parent
        }
}
```

3.6 JavaScript

Da bi izboljšali podporo programom, napisanim v programskem jeziku Java, so pri podjetju Netscape leta 1995 razvili programski jezik, ki so ga sprva poimenovali LiveScript, iz tržnih razlogov pa so ga kmalu preimenovali v JavaScript. Programski jezik JavaScript je bili razvit popolnoma neodvisno od programskega jezika Java, vendar si delita številne lastnosti in strukture .

JavaScript je objektni skriptni programski jezik, ki omogoča spletnim programerjem ustvarjanje interaktivnih spletnih strani. Izvaja se na uporabniški strani.

JavaScript spada v eno od velikih skupin programskih jezikov, ki jim na kratko rečemo tolmači. Značilnost teh programskih jezikov je, da se program izvaja sproti med njegovim razčlenjevanjem, torej za poganjanje programa ne potrebujemo prevajalnika, ki bi iz izvorne kode naredil programsko kodo, pač pa tolmač, ki izvorno kodo programa razčleni in tolmači sproti [34].

JavaScript sam po sebi ni uporaben, vanj ne moremo vnašati podatkov in nam jih tudi ne vrača. Sodelovati mora s predmeti na uporabniški strani in njihovimi lastnostni in postopki. JavaScript omogoča integracijo dokumentov HTML, spletnih komponent (Java, ActiveX...) in komponent vtičnika tako, da postanejo spletne strani dinamične.

Uporablja se na širokem področju:

- upravljanje z okni in okvirji,
- predstavitev besedila in slik,
- oblikovanje integriranih uporabniških vmesnikov,
- preverjanja vnesenih podatkov,
- pošiljanje podatkov,
- časovni nadzor,
- meniji,
- upravljanje z zgodovino in s povezavami,
- delo s piškotki [34].

Koda 3.3: Primer JavaScript tehnologije.

```
<script type="text/javascript">
        alert("Pozdravljen svet!");
</script>
```

3.7 Programski jezik C++

C++ je splošnonamenski programski jezik, kjer so podatkovni tipi statični. Kodni zapis je prost. Jezik podpira večparadigmatično proceduralno programiranje, podatkovno abstrakcijo, objektno usmerjeno in generično programiranje.

C++ je leta 1983 razvil Bjarne Stroustrup, ki je delal v Bellovih laboratorijih. Prvotno se je jezik imenoval `C with Classes` (C z razredi) in je bil razširitev programskega jezika C. Od 90. let je eden najbolj priljubljenih komercialnih programskih jezikov. Najprej so C-ju dodali razrede, nato med drugim virtualne funkcije, preobložitev, večkratno dedovanje, predloge in rokovanje z izjemami.

Leta 1998 so sprejeli tudi ISO standard za jezik C++ kot ISO/IEC 14882:1998. Trenutna različica standarda je ISO/IEC 14882:2003. Razvijajo tudi novo različico z neuradnim imenom C++0x [36].

3.8 Cascades UI platforma

BlackBerry 10 Native SDK vključuje platformo Cascade, ki jo lahko uporabljamo kot orodje z vsemi funkcijskimi lastnostmi, potrebnimi za razvoj aplikacije.

Platforma Cascade vsebuje bogat nabor komponent uporabniškega vmesnika, ki pomaga ustvariti tekočo in intuitivno izkušnjo. Skupaj z API-jem omogoča globoko integracijo z BlackBerry OS 10. Temelj Cascade-a je Qt 4.8, ki ponuja nabor nevizualnih osnov razredov in objektov.

Čeprav ima Qt popolnoma izrazito orodje za razvoj uporabniškega vmesnika, je Cascade razvil lastno platformo za uporabniški vmesnik. Platforma daje vso moč in fleksibilnost kot Qt z videzom in občutkom za interakcijo z zaslonom na dotik.

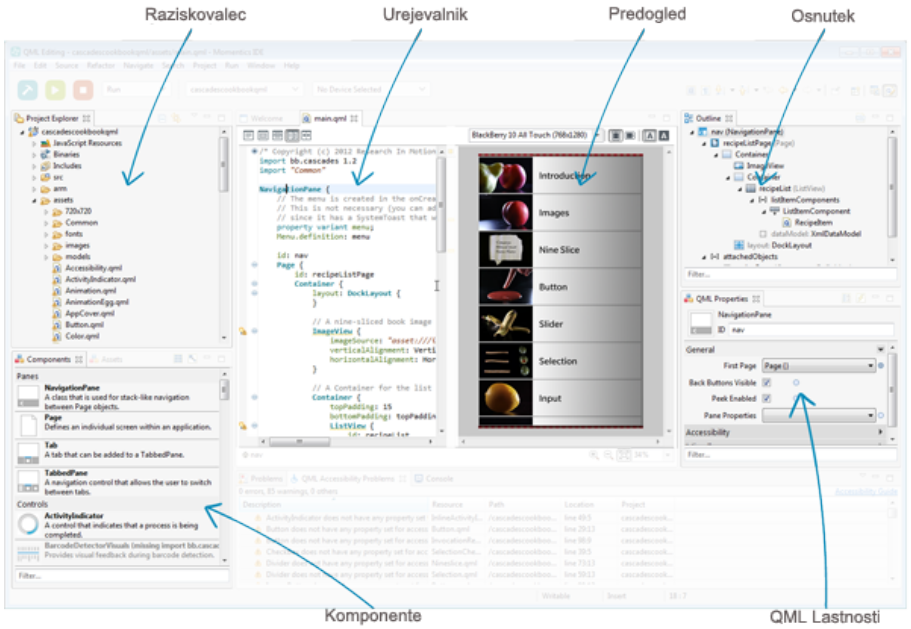
Cascade platforma je enostavna in hitra za uporabo. Ima velik nabor temeljnih kontrol, razredov za urejanje plasti ter struktur in animacij. Aplikacijska logika in UI delujeta na ločenih izvršilnih nitih, tako da med seboj komunicirajo asinhrono. Ločitev aplikacije z uporabniškim vmesnikom pomeni, da ni potrebno čakati na dolgo trajajoče procese, preden se prikaže uporabniški vmesnik. Tako nam Cascade omogoča hitro odzivnost ne glede na to, kaj se dogaja na nižjem nivoju aplikacije [12].

3.8.1 Orodje

Cascade aplikacije, kot druge native aplikacije, uporabljajo za razvoj Momentics IDE. Kot je opisano zgoraj, je Momentics IDE razvito orodje za izgradnjo C in C++ aplikacij. IDE omogoča, da prevedete, uvažate, razčlenjujete in dizajnirate svojo aplikacijo.

Momentics IDE prihaja z nizom Eclipse pogledov, ki so oblikovani posebej za pomoč pri pospešitvi procesa razvoja kaskadnih aplikacij. Dostop do teh pogledov je omogočen preko QML urejevalne perspektive v IDE okolju.

IDE ima tudi vgrajeno podporo za razčlenjevanje in profiliranje QML in JavaScript kode in tudi QML predloge, ki jih lahko uporabimo za razvoj aplikacije [12].



Slika 3.2: QML urejevalna perspektiva v Momentics IDE.

3.9 Visual Studio 2012

Visual Studio je integrirano razvojno okolje (IDE) za razvoj spletnih aplikacij in servisov ter aplikacij z grafičnim vmesnikom (GUI) ali brez njega, za namizna in mobilna okolja Windows in za vse platforme, ki poganjajo .NET Framework, Silverlight ali Mono.

Visual Studio ima vgrajeno podporo za programske jezike Visual C#, Visual

C++, Visual Basic, JavaScript in Visual F# z možnostjo podpore za ostale programske jezike (npr. Phyton, Ruby). Prav tako podpira XML/XSLT, HTML/XHTML in CSS [21].

3.9.1 .NET orodje

.NET orodje je programsko orodje za razvijanje programske opreme, ki jo je razvil Microsoft, in deluje predvsem na operacijskem sistemu Microsoft Windows [22]. Vključuje veliko knjižnic ter zagotavlja komponente jezika v več programskih jezikih. .NET je tehnologija, ki podpira izgradnjo in delovanje nove generacije aplikacij in spletnih storitev XML. Platforma je zasnovana tako, da izpolnjuje naslednje cilje:

- da zagotovi dosledno objektno orientirano programirljivo razvojno okolje,
- da zagotovi okolje kodne izvršbe, ki zmanjšuje programsko uvajanje in različice konfliktov,
- da zagotovi okolje kodne izvršbe, ki spodbuja varno izvajanje kode, vključno s kodo, ustvarjeno od tretje osebe,
- da zagotovi okolje kodne izvršbe, ki odpravlja težave glede zmogljivosti,
- zgraditi vso komunikacijo na industrijskih standardih,
- da bi ustvarili in zagotovili vso potrebno standardno komunikacijo na industrijskem področju in da je .NET platforma integrirana z drugimi tehnologijami.

Ogrodje .NET radikalno posega v samo jedro platforme Windows. Določa arhitekturo, ki strogo temelji na konceptih objektne tehnologije in komponentnega razvoja [23].

3.10 Storitve in modeli oblačnega računalništva

V informacijskih rešitvah v oblaku se moramo zavedati, da obstajajo različni infrastrukturni modeli računalništva v oblakih, ki ponujajo različne oblačne storitve. Na začetku je oblak nudil aplikacije (npr. Saleforce.com – CRM), kasneje se je izoblikovala ponudba infrastrukture v oblak (npr. Amazon AWS), sedaj pa obstajajo tudi platforme v oblaku (npr. Windows Azure). Vse tri arhitekture se med seboj razlikujejo, vendar, pa imajo eno skupno lastnost, in sicer je plačljivost storitve neposredno povezana z učinkovito in realno uporabo najetih storitev. Beseda storitev v računalništvu v oblaku je dejansko koncept, ki pomeni zmožnost ponovne uporabe neke komponente preko omrežja. Ta koncept ima definirane lastnosti, kot so visoka stabilnost, strojna neodvisnost in enostavno vključevanje in uporaba. Tako je današnje računalništvo v oblaku sestavljeno iz treh tipov storitev: SaaS, PaaS in IaaS [24].

SaaS

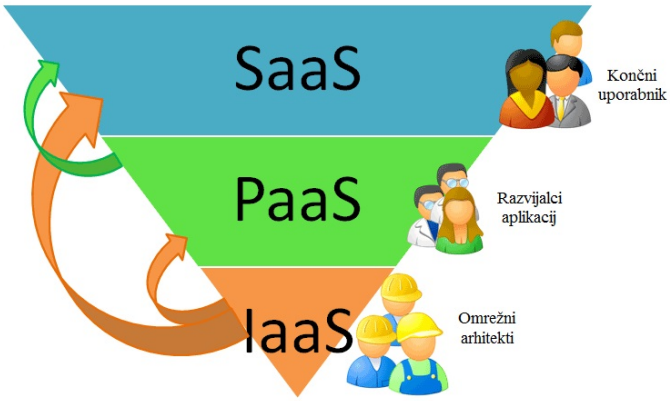
Kot storitev name je na voljo programska oprema dostopna preko spletnega brskalnika, brez nameščanja, vzdrževanja in nakupa licenc.

PaaS

Za razvoj aplikacij in njihovo izvajanje ni potrebna postavitev infrastrukture in strežniškega sistema. Storitev zagotavlja avtomatične posodobitve, visoko razpoložljivost in dodaten nabor storitev.

IaaS

Kot storitev nam omogoča postavitev virtualnega strežnika brez načrtovanja, investiranja in vzdrževanja strežniškega okolja. Poleg tega nam omogoča plačilo glede na uporabo storitve.



Slika 3.3: Prikaz treh tipov storitev računalništva v oblaku.

3.11 JSON

JSON je format zapisa za izmenjavo podatkov. Format zapisa JSON izhaja iz programskega jezika JavaScript. Zasnovan je bil kot preprosta oblika zapisa za branje in pisanje, in tudi za zajemanje, razčlenjevanje in generiranje na strojni, računalniški strani [16]. JSON temelji na dveh strukturah:

- Zbirka para ime/vrednost – v različnih programskih jezikih se to odraža kot objekt, struktura, lista.
- Urejen seznam vrednosti – v večini jezikov se to odraža kot lista, vektor, sekvenca.

Koda 3.4: Primer JSON formata iz aplikacije.

```
{"ID":3,"subject":"Matematika","idYear":1,"event_m":[]}
```

3.12 Microsoft Windows Azure

Windows Azure je postal dostopen 1. februarja 2010 kot odprta in prilagodljiva platforma računalništva v oblaku in infrastruktura, ki jo je razvil Microsoft. Omogoča razvoj, uvedbo in upravljanje aplikacij in storitev, ki se izvajajo v Microsoftovih podatkovnih centrih na različnih lokacijah po svetu. Microsoft skrbi za delovanje sistema, vzdrževanje in dograjevanje strojne opreme ter za varnost in zanesljivost podatkov. Zagotavlja PaaS in IaaS storitev in podpira različne programske jezike in tudi platforme in orodja za razvoj. Windows Azure je realizacija ideje infrastrukture in platforme kot storitve z dodatnim naborom storitev, ki se nenehno dopolnjujejo in razvijajo [17].

Nove funkcije, ki so bile izdane junija 2012:

- Spletne strani omogočajo razvijalcem uporabo tehnologije ASP.NET, Java, PHP in Node.js. Uvedba je omogočena preko FTP, Git, Mercurial ali Team Foundation Serverja.
- Virtualni stroji omogočajo selitev aplikacij in infrastrukture ne da bi spreminjali obstoječo kodo. Izvajanje je možno na Windows Serverja ali Linux virtualni mašini.
- Oblačna storitev - okolje Microsoft platforma kot storitev (PaaS, ki) se uporablja za ustvarjanje razširljivih aplikacij in storitev. Podpira večslojne scenarije in avtomatsko uvajanje.
- Upravljanje podatkov – SQL podatkovna baza, prej znana kot SQL Azure Database, sodeluje pri ustvarjanju, obsegu in razširitvi aplikacije v oblaku z uporabo Microsoft SQL Server tehnologije. Integrira se z Active Directory, Microsoft System Center in Hadoop.
- Medijske storitve – PaaS omogoča uporabo kodiranja, vsebino, zaščito, pretakanje in/ali analizo.

Windows Azure podpira tako odprtokodne tehnologije kot celotno .NET ogrodje, Windows in SQL strežnik, zato lahko platformo hitro in učinkovito

integrirate z vašim obstoječim IT sistemom in tako ponudite končnim uporabnikom prednosti oblaka, kot je učinkovita povezljivost in možnost fleksibilne uporabe neomejene računalniške moči. Windows Azure lahko povežete v vaše VPN omrežje ali pa ga uporabite za zanesljivo in varno povezovanje z drugimi sistemi.

Windows Azure platforma nudi API zgrajen na REST, HTTP in XML, kar omogoča razvijalcem integracijo s storitvami. Microsoft omogoča tudi na klientovi strani knjižnice razvojni paket, ki vključuje celo funkcionalno integracijo s storitvami. Poleg tega se integrira z Microsoft Visual Studio, Git in Eclipse [18].

Razvojni paket Windows Azure SDK vključuje razvojne zmožnosti za pisanje kode za izboljšane storitve. Prav tako je posodobljena podpora za Javo, PHP, .NET in Python.

Storitev Virtual Network omogoča uvedbo in upravljanje virtualnih zasebnih omrežij na platformi Windows Azure in varno razširitev omrežij na lastni lokaciji v oblak. Storitev nudi nadzor nad omrežno topologijo, vključno s konfiguracijo IP-naslovov, usmerjevalnih tabel in varnostnih politik. Za zagotavljanje varnih povezav med poslovnimi VPN-prehodi in platformo Windows Azure uporablja standardni protokol IPSEC [19].

Windows Azure Compute storitev ponuja visoko razpoložljive instance virtualnih strežnikov, ki jih uporabljamo po potrebi kot storitev in zato ne plačujemo visokega enkratnega zneska za strojno opremo in licence, ampak le ceno porabljene računalniške moči glede na čas uporabe.

Vsaka Azure Compute instanca je v oblaku popolnoma izolirana od drugih instanc. Platforma Windows Azure avtomatično poskrbi za virtualizacijo, namestitev operacijskega sistema in strežnika, nastavitve omrežne povezave in uravnoteženje obremenitev. Tako smo razbremenjeni dela upravljanja in vzdrževanja infrastrukture in se lahko popolnoma posvetimo aplikativnemu nivoju rešitve. Windows Azure nam po pogodbi SLA zagotavlja, da je virtualna instanca dosegljiva vsaj 99.95% časa izvajanja [20].

Poglavje 4

Programska rešitev mobilne aplikacije

V poglavju opišemo zahteve in zasnovo aplikacije. Opisana in prikaza je tudi arhitektura aplikacije ter kratka poslovna logika delovanja aplikacije, kar se tiče odjemalčeve in strežniške strani. Prikazane so tudi funkcije uporabniškega vmesnika s poudarkom na funkcionalnosti posameznih strani.

4.1 Zahteve za aplikacijo

Mobilna aplikacija mSTUDY je namenjena vsem študentom, ki bi radi poleg formalnega znanja pridobili še dodatno znanje s pomočjo drugih študentov s podobnimi izkušnjami. S tem lahko olajšamo študij študentom, saj lahko kadarkoli objavijo objavo glede skupnega učenja, ne da bi poznali druge sodelujoče. Na objavo se lahko odzovejo vsi, ki jih tovrstni predmet zanima in bi radi sodelovali pri svojem razvoju in razvoju drugih študentov. Odlično komunikacijsko orodje je lahko tudi za študente prvih letnikov, ki šele začenjajo svojo študijsko pot in potrebujejo veliko dobrih nasvetov. Skupino ljudi lahko združimo v učno skupino, kjer prispevajo s svojim znanjem, nasveti in predlogi. Za nekaj takega je potrebno le določiti lokacijo, uro srečanja ter tematiko. Objavo vidijo zainteresirani študentje, ki se prijavijo

v učno skupino, to pa se ustrezno shrani ter dokumentira.

4.2 Zasnova aplikacije

Programsko rešitev mobilne aplikacije smo razvili v razvojnem okolju QNX Momentics z razvojnim paketom BlackBerry Native 10 SDK. Razvojni paket nam je omogočil dostop do odprtokodnih knjižnic in integracijo z BlackBerry platformo. Za potrebe razvoja smo uporabili dostop do večjih integriranih knjižnic, nam najbolj pomemben dostop do funkcij koledarja in izbire oz. iskanja lokacije.

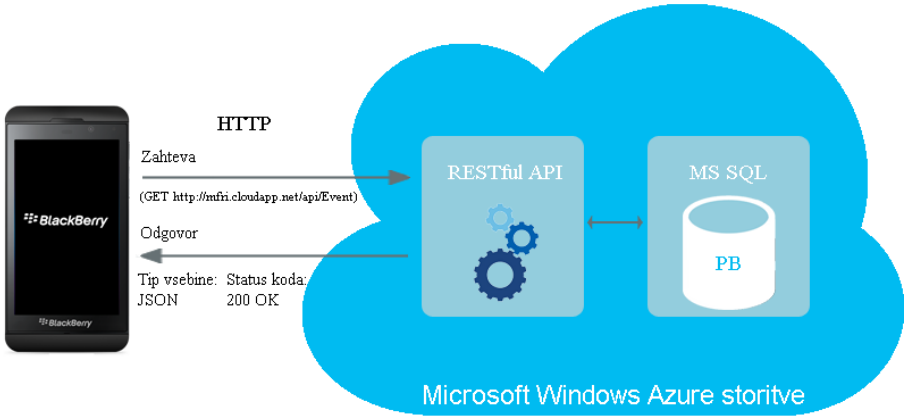
S pomočjo orodja Visual Studio 2012 smo aplikaciji omogočili integracijo s podatkovno bazo. S tehnologijo .NET in Windows Azure SDK za Visual Studio smo zgradil RESTful spletno storitev. Spletna storitev je programski vmesnik-API, ki nam omogoča dostop do podatkovne baze.

Za upravljanje s podatki smo uporabili Microsoft SQL podatkovno bazo. Podatkovna baza je sestavljena iz večjih tabel, ki so med seboj relacijsko povezane. Baza in spletna storitev se nahajata na Microsoft Windows Azure platformi.

4.3 Arhitektura

Pred samim razvojem rešitve smo morali točno zastaviti sistem arhitekture. Arhitektura je sestavljena iz treh delov, in sicer mobilne aplikacije, vmesnikom RESTful API, ki skrbi za dostop in integracijo s podatkovno bazo, tretji del pa je podatkovna baza.

S protokolom HTTP skrbimo za komunikacijo med odjemalcem in RESTful spletno storitvijo, medtem ko se podatki prenašajo v formatu objektov JSON. Odjemalec, ki teče na mobilni aplikaciji, vsako zahtevo pošlje do vmesnika, ta pa naprej do podatkovne baze. Zahteva se primerno odzove z ustreznimi podatki do odjemalca v JSON formatu.



Slika 4.1: Prikaz arhitekture aplikacije.

4.4 Poslovna logika

Poslovna logika opisuje delovanje mobilne aplikacije s poudarkom na komunikaciji med odjemalcem in spletno storitvijo RESTful. Opisane so tudi tehnologije, uporabljene za izgradnjo RESTful API-ja, in migracije le-te na platformo Microsoft Windows Azure.

4.4.1 Odjemalec

Za dostop do spletnih storitev preko mobilne aplikacije smo uporabili razred QNetworkAccessManager. QNetworkAccessManager je razred, ki omogoča aplikaciji pošiljanje zahtev preko omrežja do RESTful spletne storitve. Network Access API, ki omogoča omrežni dostop, je zgrajen okrog enega QNetworkAccessManager objekta. Objekt ima skupno konfiguracijo in nastavitve za pošiljanje zahtev. Vsebuje proxy in predpomnilniško konfiguracijo, pa tudi signale, ki nam omogočajo nadzor operacije procesa podatkov skozi omrežja [29].

Koda 4.1: Primer pošiljanja zahteve GET.

```
QUrl url("http://mfri.cloudapp.net/api/Event");
QNetworkRequest request(url);
QNetworkAccessManager manager;
QNetworkReply *reply = manager.get(request)
QObject::connect(reply, SIGNAL(finished), myClass, SLOT(replyFinished()));
```

Ko se QNetworkAccessManager objekt kreira v aplikaciji se lahko prične pošiljanje zahtev skozi omrežje. Na voljo so skupni funkcijski standardi, ki sprejmejo zahtevo in podatke. Odgovor se vrne v obliki objekta QNetworkReply. Vrnjeni objekt se uporablja za sprejemanje odgovora in posledično pridobivanje podatkov. V našem primeru smo dobili JSON format podatkov.

QNetworkAccessManager ima asinhroni API, kar nam omogoča, da se sproži reža ob končanem odgovoru storitve v katerem so podatki oz. QNetworkReply objekt, ki vsebuje podatke, kot tudi http specifikacije. Režo smo uporabili za prikaz procesiranja podatkov ob tem, ko uporabnik da zahtevo po dostopu do storitev. Najbolj očiten primer je ob prijavi v aplikacijo, kjer je prikazan proces pridobivanja podatkov oz. vrstica napredka [30].

Koda 4.2: : Primer sprejemanja odgovora.

```
QNetworkReply *reply = qobject_cast<QNetworkReply*>(sender());
if (reply) {
        if (reply->error()) == QNetworkReply::NoError) {
            QString myData = reply->property("data").toString();
        }
        reply->deleteLater();
    }
}
```

4.4.2 Spletna storitev RESTful API

Windows Azure omogoča ustvarjanje in upravljanje spletnih storitev računalništva v oblakih. Da omogočimo dostop do storitev Windows Azure, je treba najprej ustvariti Windows račun. Windows ponuja dostop do funkcij en mesec brezplačno. Za registracijo potrebujete Windows račun in kreditno kartico. Za naše potrebe smo na Windows Azure portalu ustvarili spletno storitev,

do katere bo dostopala naša mobilna aplikacija in podatkovna baza. Le v nekaj korakih ustvarimo storitev v oblaku, v katerem navedemo URL naslov za dostop do storitve in regijo. S tem je storitev pripravljena za migracijo API-ja.

Slika 4.2: Primer Windows Azure oblačne storitve.

Z orodjem Visual Studio 2012 in paketom Windows Azure SDK smo zgradili storitev ASP.NET Web API. Web API je zgrajen s tehnologijo ASP.NET MVC. To orodje gradi na standardni ASP.NET tehnologiji v smislu uporabe skupnih knjižnic. ASP.NET MVC je postalo zelo priljubljeno razvijalsko orodje, ki uporablja Microsoftove tehnologije. Velika prednost orodja je, da se lahko kateri koli del razširi in prilagodi potrebam.

Po kreiranju Web API-ja nam preostane kreiranje modela, ki nam bo omogočil dostop do podatkovne baze. Podatkovno bazo SQL smo prej kreirali na Windows Azure portalu skupaj s spletno storitvijo.

Model za dostop do podatkovne baze kreirano z uporabo tehnologije ADO.NET Entity Data Model. S pomočjo čarovnika po korakih vnesemo podatke SQL strežnika in izberemo podatkovno bazo. Po končanih korakih v čarovniku nam preostane še povezava s podatkovno bazo in našim Web API-jem. To storimo s krmilniki (Controller). Za vsako tabelo iz podatkovne baze kreiramo krmilnik, preko katerega dostopamo z mobilno aplikacijo do podatkovne baze preko Web API-ja.

ASP.NET MVC

Orodje ASP.NET MVC je alternativni model za razvoj spletnih rešitev. Zasnovano je bilo, ker so želeli poiskati dober princip razvoja programske

opreme. Temelji na ASP.NET tehnologiji, ki omogoča razvijalcem izgradnjo spletnih aplikacij. Ta tehnologija je sestavljena iz treh nivojev: Model, Krmilnik, Pogled.

Model

Model prestavlja poseben vidik aplikacije. Vsebuje aplikacijsko logiko, ki ni del krmilnika in pogleda. V našem primeru predstavlja poslovno logiko dostopa do baze podatkov. Model je objekt, ki predstavlja podatke. ASP.NET Web API lahko samodejno serializira objekt oz. model v JSON, XML ali katero koli drugo obliko. Serializiran objekt podatkov zapiše v telo odgovora HTTP.

Krmilnik

V spletnem API-ju je krmilnik objekt, ki obravnava http zahteve. Krmilnik nadzoruje tok izvajanja aplikacije tako, da komunicira med modelom in pogledom. Naloge krmilnika so:

- lociranje oz. klicanje primerne metode in potrditev le-te da se lahko kliče,
- pridobivanje vrednosti parametrov ob klicanju metode,
- skrbi za napake, ki se lahko pojavijo med izvajanjem metode.

Vsaka metoda, ki je javna, se imenuje akcijska metoda. Vsaka metoda je povezana z URL naslovom preko ASP.NET usmerjevalnega sistema. V primeru, da pošljemo zahtevo na nek URL, ki je vezan na akcijsko metodo v krmilniku, krmilnik izvede logiko, opredeljeno znotraj klicane metode.

Pogled

Pogled sprejme potrebne podatke od krmilnika in jih tudi vrača. Vsebuje HTML oznake ter logiko potrebno za ustvarjanje pogleda. Naloga pogleda

je prikaz zajetih podatkov in določanje oblike prikaza ter interakcijo z uporabnikom. Pogled za naše delo smo uporabljali zgolj za lokalno testiranje podatkov med podatkovno bazo in Web API-jem [32].

Web API

Web API je orodje, ki omogoča preprosto izgradnjo http spletnih storitev z zelo kratko kodo. Doseže lahko širok spekter odjemalcev. Web API je idealna platforma za izgradnjo RESTful aplikacij v .NET tehnologiji. Bazira na ASP.NET MVC orodju, kot knjižnica je zgrajena na novo. Za izgradnjo modernih aplikacij je ključnega pomena. Web API je osrednji del naše arhitekture in je ključni del med aplikacijo in podatkovno bazo.

Strežniška stran web API-ja je programski vmesnik, ki se na določeni sistem odziva v JSON ali XML formatu. Izpostavljen je preko spleta, ki temelji na http spletnem strežniku.

Web API lahko poimenujemo kot sinonim za spletne storitve. Web 2.0 spletne aplikacije so se odmaknile od SAOP spletnih storitev k bolj kohezivni zbirki spletnih virov RESTful. RESTful spletni API-ji so dostopni preko standardnih http metod z različnimi http klienti, vključujoč brskalnike in mobilne naprave [35].

REST in RESTful

REST (Representational State Transfer) je arhitekturno-programska aplikacija, ki je zasnovana glede na to, kako so podatki predstavljeni po dostopnosti in kako so modificirani na spletu. V REST arhitekturi so podatki in funkcionalnost odvisni od virov. V našem primeru od podatkovne baze. Dostopnost do teh virov je omogočena preko naslova URI, ki dostopa preko povezave do spleta. Viri delujejo na podlagi preprostih in dobro definiranih operacij. REST arhitektura je namenjena izmenjavi komunikacijskega protokola, največkrat http-ja. V REST arhitekturi klienti in strežniki izmenjujejo vire

z uporabo standardnih vmesnikov in protokolov. Takšna načela spodbujajo REST aplikacijo, da bo preprosta, lahka in z visoko zmogljivostjo.

RESTful spletna storitev je spletna aplikacija, ki temelji na REST arhitekturi.Naloge spletne storitve:

- razkrivanje virov (podatke in funkcionalnost) skozi spletni URI,
- uporabljajo glavne štiri HTTP metode: ustvarjati, pridobivati, posodabljati in brisati vire (CRUD).

RESTful spletna storitev umešča glavne HTTP metode v tako imenovane CRUD akcije: kreiranje, pridobivanje, posodabljanje in brisanje. Spodnja tabela prikazuje http metode na CRUD ukrepih [33].

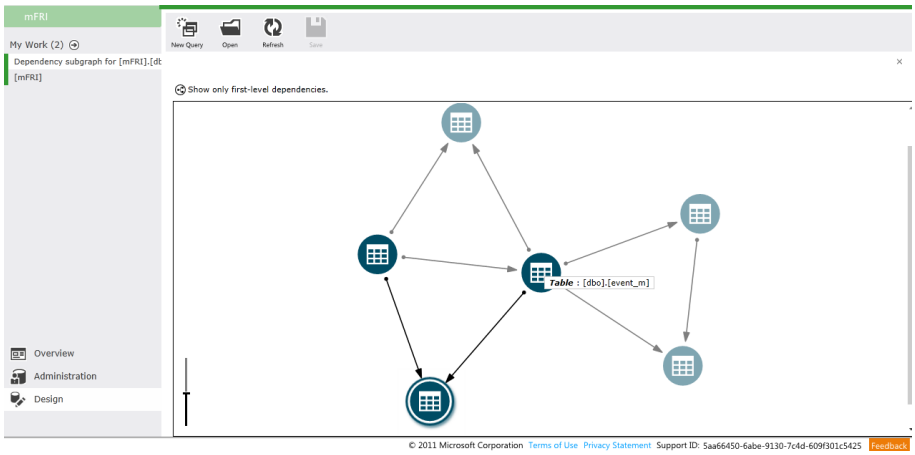
HTTP metoda	CRUD akcija
GET	Pridobivanje virov
POST	Kreiranje virov
PUT	Posodabljanje virov
DELETE	Brisanje virov

Tabela 4.1: HTTP metode s pripadajočimi CRUD akcijami.

4.4.3 Podatkovna baza

Microsoft nam je omogočil celotno upravljanje s podatkovno bazo preko spletnega vmesnika Windows Azure. Za naše potrebe smo uporabili Windows Azure SQL podatkovno bazo oz. formalno kot SQL Azure. Za kreiranje podatkovne baze smo preko spletnega vmesnika le v nekaj korakih ustvarili delujočo podatkovno bazo. Spletni vmesnik omogoča celotno načrtovanje podatkovne baze, vključno s kreiranjem, spreminjanjem, brisanjem in tudi dodajanjem odvisnosti med tabelami z dodajanjem primarnega in tujega ključa.

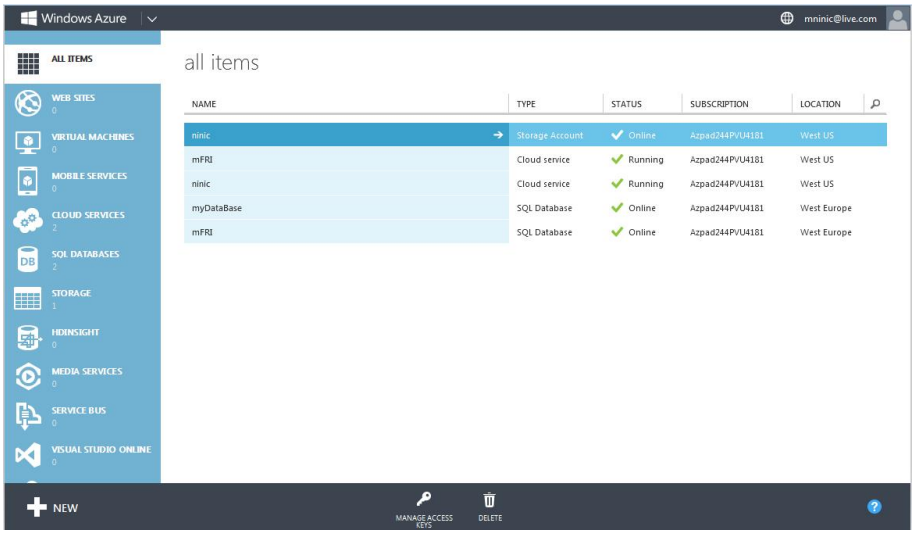
Za naše potrebe smo podatkovno bazo v celoti zgradili preko spletnega vmesnika. Na Sliki 4.3 so preko spletnega vmesnika prikazane odvisnosti med tabelami v podatkovni bazi.



Slika 4.3: Grafični prikaz tabel v odvisnosti.

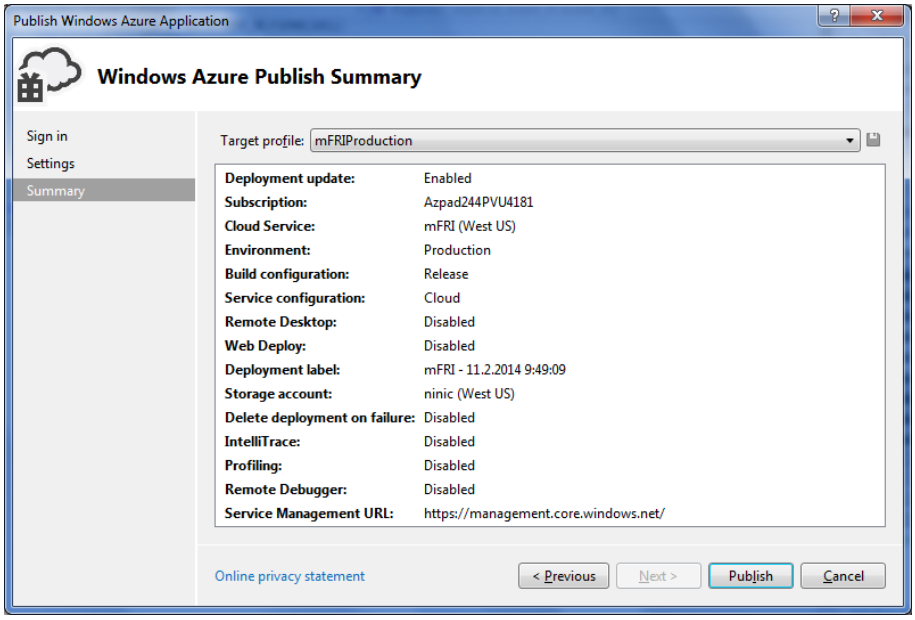
4.5 Migracija RESTful API-ja v Windows Azure

Ko zaključimo z razvojem, načrtovanjem in testiranjem REST API-ja na lokalnem omrežju, je čas, da storitev prestavimo na oblačno platformo Windows Azure. Za dostop do storitev, kot omenjeno zgoraj, potrebujemo Microsoft račun. Na Sliki 4.4 je prikazan spletni vmesnik Windows Azure.



Slika 4.4: Spletni vmesnik Windows Azure.

Programsko orodje Visual Studio 2012 nam omogoča enostaven in hiter prenos aplikacij in storitev na Windows Azure platformo. S klikom miške na zavihek `Build` v razvojnem okolju izberemo opcijo `Publish to Windows Azure`. Odpre se nam okno na sliki 4.5, kjer se najprej prijavimo z istim računom kot v Windows Azure portalu. S klikom na `Publish` se začne prenos na oblačno platformo Azure. Po končanem prenosu testiramo delovanje prenosa podatkov s pošiljanjem zahtev na URL `http://mfri.cloudapp.net/`.



Slika 4.5: Prenos na Windows Azure platformo.

Testiranje omogočimo z vtičnikom `RESTClient` za brskalnik `Mozilla Firefox`. Z `RESTClient`-om omogočimo pošiljanje RESTful zahtev do vmesnika in prikaz odgovora.

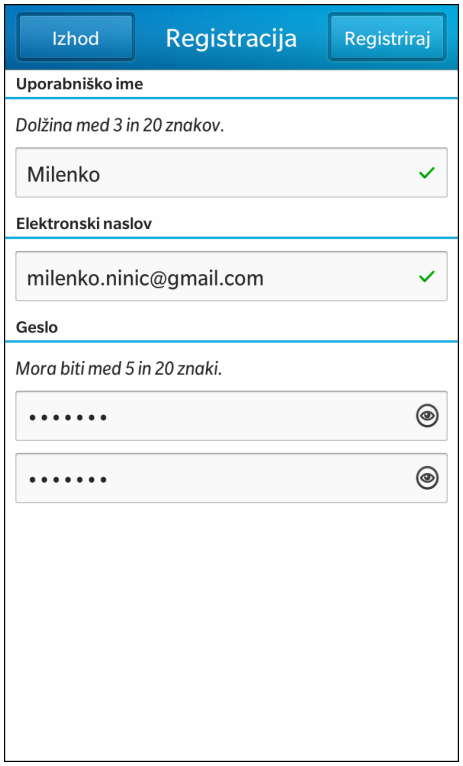
4.6 Uporabniški vmesnik

Pri načrtovanju uporabniškega vmesnika je koristno, da vnaprej izrišemo želeni model uporabniškega vmesnika. Tako smo se odločili tudi pri našem načrtovanju. Prvotno verzijo uporabniškega vmesnika smo izrisali grafično in tako dobili boljšo predstavo o videzu in funkcionalnosti aplikacije. S pomočjo skiciranja uporabniškega vmesnika smo lažje določali potencialne težave, ki se lahko pojavijo.

4.6.1 Registracija in prijava v aplikacijo

Da se uporabnik prijavi v aplikacijo in jo začne uporabljati, je potrebna registracija. Za registracijo uporabnik na začetni strani klikne na gumb

`Registriraj`. Uporabnik se registrira z vnosom zahtevanih podatkov. Vsa polja, ki so prikazana, so obvezna. Uporabnik vnese svoje ime oz. vzdevek, elektronski naslov in geslo. Vsi podatki se med izpolnjevanjem polj validirajo tako, da če študent vnese napačen vpisni format, ga aplikacija opozori in prikaže obvestilo o napaki. V primeru prekinitve registracije ima uporabnik izbiro gumba `Izhod` za prekinitev. Po končanem vnosu podatkov za registracijo v aplikacijo uporabnik klikne na gumb `Registriraj`. S klikom na gumb se pošlje http zahteva, s katero omogočimo vnos uporabniških podatkov v podatkovno bazo, in s tem je zaključena registracija. Po registraciji se uporabniku prikaže okno, v katerem je obvestilo o uspehu oz. napaki prijave. Z uspešno prijavo se avtomatsko okno za registracijo umakne in prikaže se začetna stran aplikacije z vnosnimi polji `Uporabniško ime` in `Geslo`.



Slika 4.6: Prikaz okna za registracijo uporabnika.

Po končani registraciji se študentu omogoči prijava v aplikacijo. Z vnosom

uporabniškega imena in gesla ter klikom na gumb `Prijava` se pošlje http zahteva z vpisanimi podatki, kjer se na vmesniku in posledično na podatkovni bazi preverijo podatki. V času preverjanja podatkov se uporabniku prikaže ikona napredka. V primeru napačnega vnosa se študentu prikaže obvestilo o napačnem vnosu uporabniškega imena oz. gesla. V primeru izpada omrežja se, prav tako prikaže obvestilo. Z uspešno prijavo se uporabniku prikaže glavni meni aplikacije, ki je prikazana na Sliki 4.8.

Slika 4.7: Prikaz okna za prijavo v aplikacijo.

4.6.2 Meni aplikacije

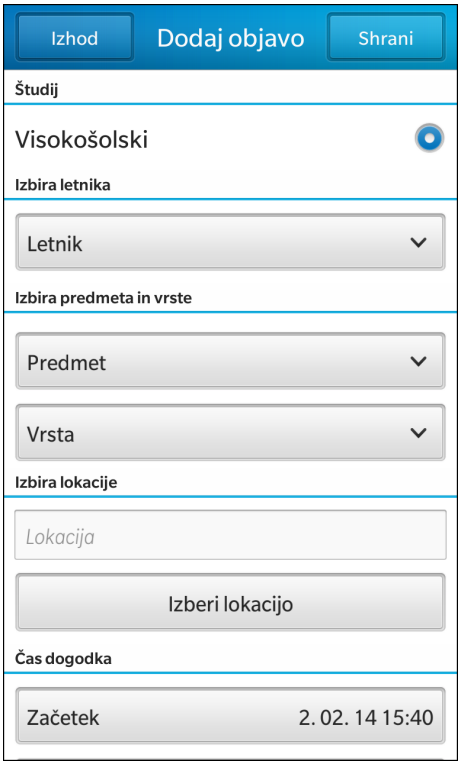
Ob prijavi ima uporabnik vse možnosti upravljanja z aplikacijo. Prva funkcija aplikacije je dodajanje objav. Funkcija omogoča uporabniku izbiro vseh atributov objave. Po končani objavi se uporabniku podatki dogodka sinhronizirajo s koledarjem na operacijskem sistemu. Temeljit opis funkcije `Dodaj`

bo opisan v nadaljevanju. Po končani objavi ima uporabnik možnost pregleda svojih objav in upravljanja z njimi s klikom na gumb `Moje`. Podroben opis funkcije `Moje` bo opisan v nadaljevanju. Sledi funkcija `Vse`, ki nam omogoča prikaz vseh objav uporabnikov aplikacije. Uporabnik ima možnost nastavitve izbire iskanja svojih priljubljenih objav. To pomeni, da uporabnik nastavi, katere objave ga zanimajo. Na primer: študent prvega letnika si bo nastavil prikaz vseh objav prvega letnika itd. Sledi opcija `Najdi`, ki nam omogoča iskanje objav po vseh atributih objave. Na primer: po organizatorju dogodka, predmetu, tipu ... Zadnja opcijska izbira v meniju omogoča odjavo iz aplikacije. Omenjene funkcije bodo bolj podrobno opisne v spodnjih poglavjih.

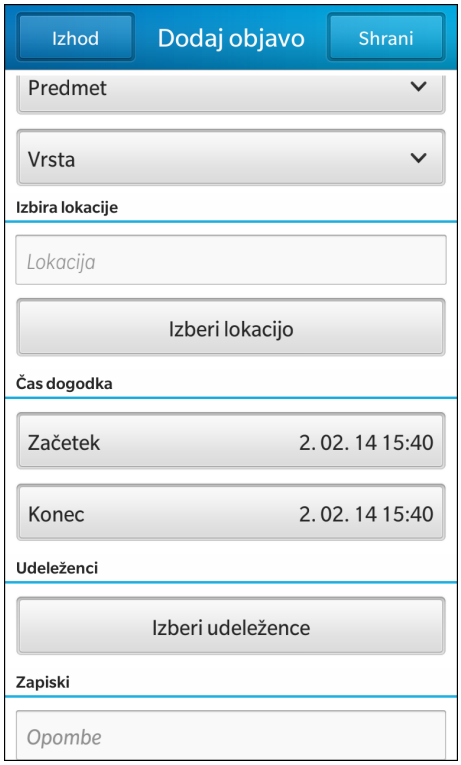
Slika 4.8: Prikaz menija aplikacije.

4.6.3 Dodajanje objav

Za dodajanje objav študent klikne na opcijo `Dodaj` v meniju aplikacije. S klikom se prikaže stran za dodajanje objav za izbrani predmet, tip in letnik študija. Za uspešno dodajanje objave je potrebno izpolniti ustrezna zahtevana polja.



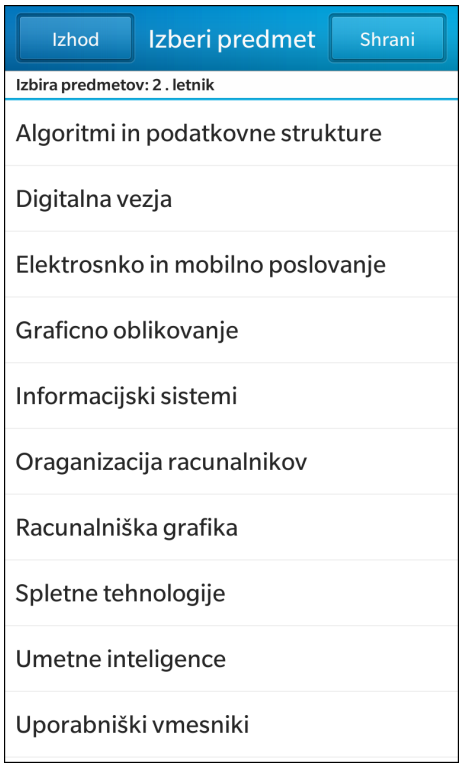
Slika 4.9: Kreiranje objave.



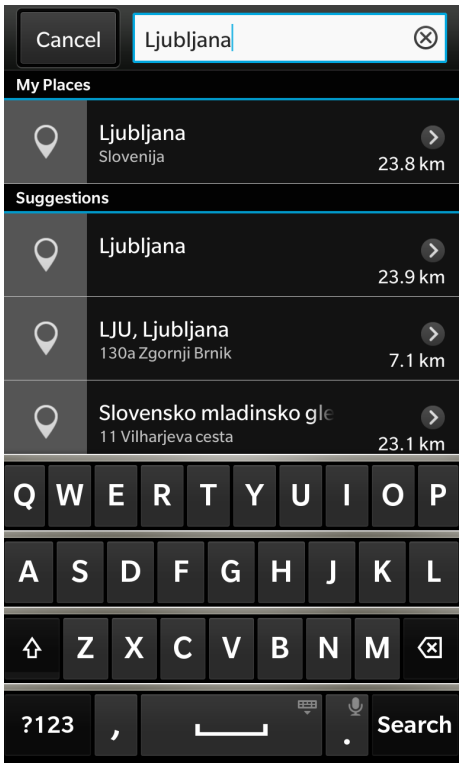
Slika 4.10: Kreiranje objave.

Prva izbira je letnik, kjer študent izbere za kateri letnik bo ustvaril objavo. Na podlagi izbranega letnika študija, študent izbere za kateri predmet bo ustvaril objavo. Primer izbire predmeta na podlagi izbranega letnika je prikazano na Sliki 4.11. Po izbiri predmeta študent izbere za katero vrsto predmeta bo ustvaril objavo. Imamo več vrst objav: izpit, kolokvij in seminar. Sledi izbira lokacije. Študentu aplikacija omogoča izbiro lokacije, kje naj se bi se dogodek izvedel. Študent s pomočjo API-ja izbere lokacijo, in le-ta se prikaže v polju `Lokacija`, ki jo lahko še po želji spreminja. Prikaz

izbire lokacije je na Sliki 4.12.

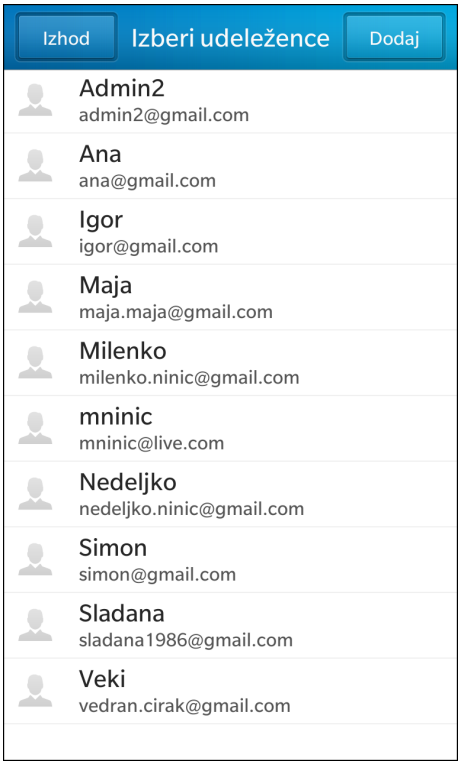


Slika 4.11: Primer izbire predmeta.

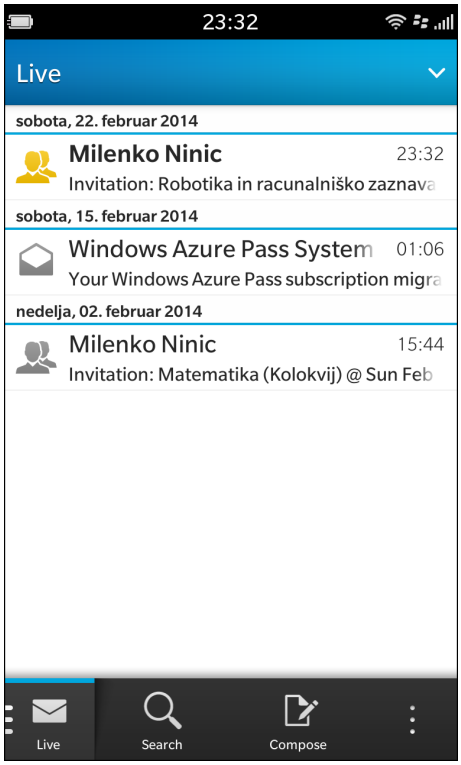


Slika 4.12: API za izbiro lokacije.

Sledi določanje časa dogodka in koliko časa bo trajal dogodek. Pri vsaki objavi ima študent izbiro dodajanja udeležencev. Tako je študentu omogočeno ustvarjanje vabil drugim študentom. Prikaz dodajanja je na Sliki 4.13. Vsaki povabljeni dobi vabilo po elektronski pošti, in se na vabilo ustrezno odzove. Prikaz vabila je na Sliki 4.14.



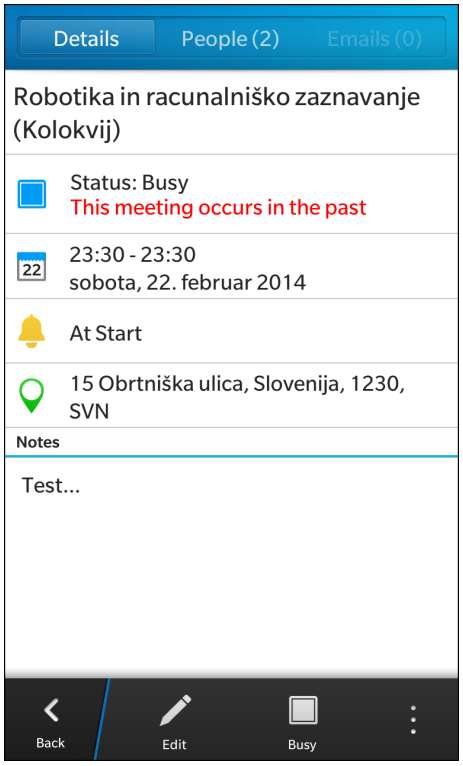
Slika 4.13: Prikaz izbire udeležencev.



Slika 4.14: Prikaz vabila v elektronski obliki.

Prikazana je lista udeležencev in njihovi kontaktni podatki. Z liste si lahko izberemo udeleženca in ga povabimo na določen dogodek. Tisti udeleženci, ki so izbrani, dobijo obvestilo na elektronski naslov. Obvestilo vsebuje podatke o dogodku, na podlagi tega pa se posamezniki potem lahko odločijo, ali se bodo dogodka udeležili. V kolikor potrdijo povezavo, se aplikacija sinhronizira z udeleženčevim koledarjem.

Slika 4.15: Prikaz dogodka v koledarju mobilne naprave.



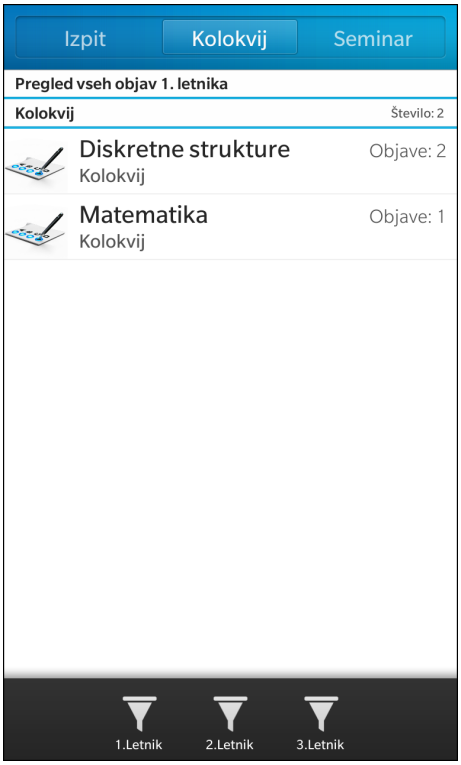
Slika 4.16: Podroben prikaz podatkov v koledarju.

Ko udeleženec potrdi dogodek v aplikaciji mSTUDY, se ta sinhronizira z koledarjem. Z sinhronizacijo dobimo novi ustvarjeni dogodek v mobilni aplikaciji telefona `Koledar`. V koledarju se avtomatsko zabeležijo podatki o dogodek. S klikom na dogodek v koledarju se izpišejo podrobnosti le-tega. Izpišejo se: opis dogodka, datum srečanja, lokacija srečanja ter komentarji za dogodek. Primer dogodka v koledarju je prikazan na Sliki 4.15 in 4.16.

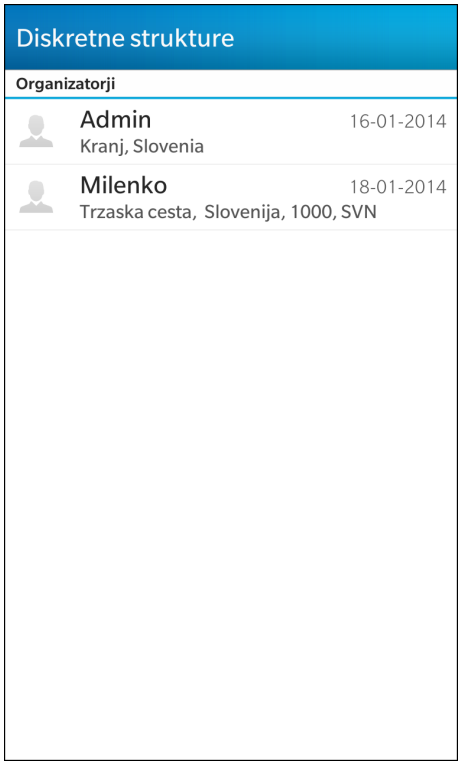
4.6.4 Prikaz objav

Omogočen je prikaz vseh oz. mojih objave izbranega letnika z izborom tipa dogodka. Torej: v tem primeru je izbran tip kolokvij, kjer lahko vidimo vse objave prvega letnika. S klikom na eno objavo, v našem primeru `Diskretne strukture`, lahko vidimo kdo so organizatorji dogodka, kje se bo dogodek

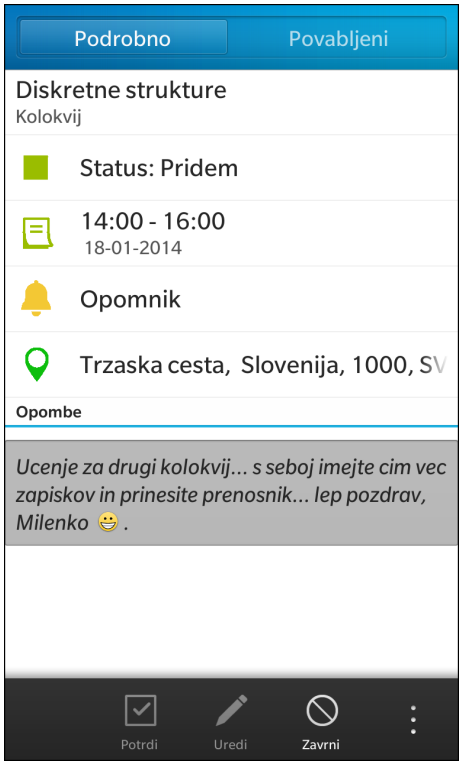
izvajal in datum dogodka. Za podrobne informacije uporabnik klikne na ustrezno objavo.



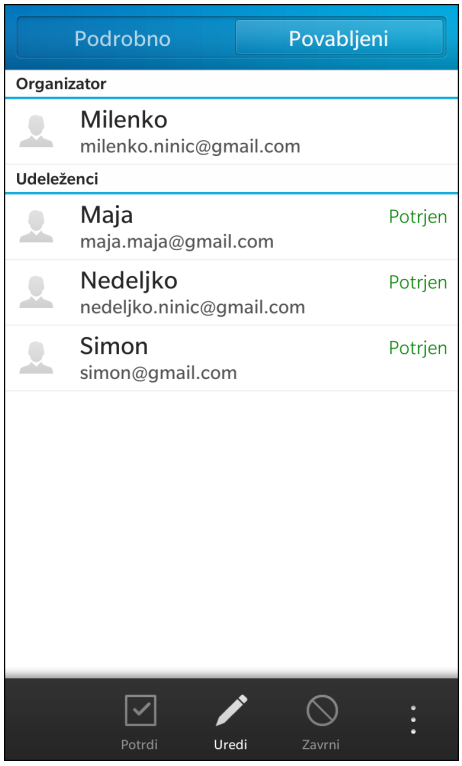
Slika 4.17: Prikaz objavljenih dogodkov v strukturi.



Slika 4.18: Prikaz objav istega tipa.



Slika 4.19: Podroben prikaz dogodka.



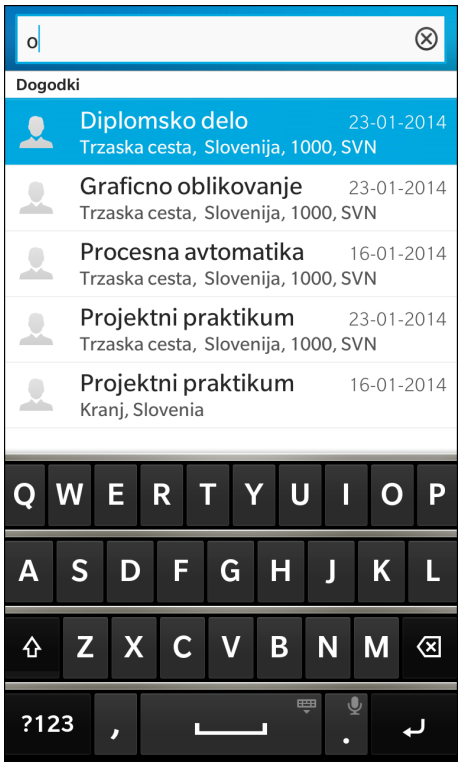
Slika 4.20: Prikaz povabljenih oseb na dogodek.

Na Sliki 4.19 in 4.20 je prikazano skreirano vabilo na učno srečanj namenjeno uterjevanju znanja, za kolokvij prvega letnika pri predmetu `Diskretne strukture`. Vsebuje podatke o statusu, torej: ali se bo udeleženec dogodka udeležil, ura dogodka. Za opomnik je urejena povezava z koledarjem ter naslov, kjer se bo dogodek odvijal. Prav tako so dodane `Opombe`, kjer lahko avtor objave napiše določene komentarje, zahteve. Vse to se lahko vidi v zavihku `Podrobno`. V zavihku povabljeni lahko vidimo kdo je organizator dogodka ter vse udeležence, ki so potrdili svojo udeležbo. Prikazan je status udeležencev ter njihov elektronski naslov.

4.6.5 Filter objav

Vse objave se lahko iščejo z iskalnikom. Na Sliki 4.22 je prikazano iskanje po ključni besedi. Tako lahko vidimo vse objave, ki vsebujejo določeno ključno besedo.

Slika 4.21: Prikaz filtriranja objav.

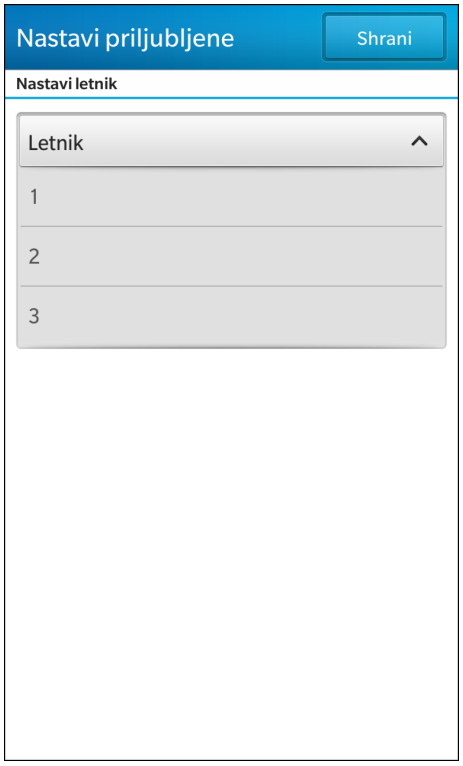


Slika 4.22: Prikaz filtriranih objav.

4.6.6 Priljubljene objave

V aplikaciji je omogočena nastavitev priljubljenih vsebin oz. priljubljenih objav. To lahko nastavimo s filtrom po letniku. Z izborom ustreznega letnika imamo prikaz vseh objav za določen letnik, ki smo ga izbrali.

Slika 4.23: Prikaz priljubljenih dogodkov.



Slika 4.24: Nastavitve priljubljenih dogodkov.

Poglavje 5

Sklepne ugotovitve

Danes si zelo težko predstavljamo življenje brez mobilne telefonije. Z razvojem pametnih telefonov pa je mobilna telefonija postala nepogrešljiv del našega življenja. Zelo pomembni so tudi za študente, saj ob napornem tempu življenja želimo, da so vse pomembne stvari hitro dosegljive. Tehnologija se razvija, tako da je ves čas treba razvijati orodja in programe, ki olajšajo delo in življenje človeku. Zaradi tega smo tudi mi razvili in predstavili mobilno aplikacijo mSTUDY. V diplomski nalogi smo že predstavili vse tehnologije in orodja, ki so nam pomagala, da dobimo takšno aplikacijo, ki deluje in je pripravljena na takojšnjo uporabo.

Predstavili smo mobilno aplikacijo, ki je zasnovana na operacijskem sistemu Blackberry. Uporabili smo razvojni paket Blackberry 10 Native SDK. Na strežniški strani smo uporabili storitev Microsoft Windows Azur, kjer smo implementirali spletno storitev in podatkovno bazo.

Namen mobilne aplikacije je, da imajo študentje ob vsakem trenutku dostop do nje in do takojšnjih objav učnih urnikov. Vsi uporabniki Blackberry telefonov lahko dostopajo do spletne aplikacije. Potrebna je registracija, ko uporabnik vnese svoje kontaktne podatke, kamor bo tudi dobival obvestila. Ko se registriramo, je potrebno vnesti uporabniško ime in geslo. V glavnem meniju so predstavljene opcije za izbiro, tako študentje lahko dodajo nov dogodek, kjer lahko povabijo določene udeležence, ali pa drugi študentje lahko

vidijo objavo in potrdijo svojo navzočnost. Omogočena je tudi sinhronizacija s koledarjem ter opominjanje po elektronski pošti.

S takšno aplikacijo študentje lažje navežejo stik, hitreje dobijo določene informacije, pridružijo se učnim skupinam ali pomagajo drugim v primeru težav pri učenju.

Kot možnost za izboljšavo bi lahko predlagali opcijo prijave s certifikatom, saj bi bila s tem omogočena bolj varna uporaba aplikacije. Prav tako smo razmišljali tudi o uvedbi diskusije o določeni objavljeni temi. Študentje bi lahko objavljali svoje predloge in mnenja. Opcija bi torej lahko bila tudi uvedba foruma za izpite, kolokvije ali seminarje.

Aplikacija je lahka za uporabo, poda veliko informacij ter lahko združuje veliko študentov z željo po učinkovitem učenju in deljenju znanja.

Slike

Tabele

Literatura

[1] (2014) QNX Momentics Tool Suite. Dostopno na: http://www.qnx.com/products/tools/qnx-momentics.html

[2] (2012) BlackBerry razvijalci imajo boljša orodja. Dostopno na: http://www.bbdevbalkan.com/home/2012/11/blackberry-razvijalci-imajo-boljsa-orodja/?lang=sl

[3] (2014) Razvoj v okolju AIR. Dostopno na: http://developer.blackberry.com/air/documentation

[4] (2014) Razvoj z HTML5 WebWorks. Dostopno na: https://developer.blackberry.com/html5/documentation/beta

[5] (2014) BlackBerry. Dostopno na: http://en.wikipedia.org/wiki/BlackBerry

[6] (2012) RIM stavi na BlackBerry. Dostopno na: http://www.monitor.si/novica/rim-stavi-na-blackberry-10/140578/?xURL=301

[7] (2014) Prednosti in slabosti BlackBerry-ja. Dostopno na: http://our-techblog.blogspot.com/2011/06/advantages-and-disadvantages-android.html

[8] (2011) Prednosti in slabosti BlackBerry-ja. Dostopno na: http://www.blazkos.com/blackberry.php

[9] (2013) Operacijski sistem BlackBerry. Dostopno na: http://www.monitor.si/novica/blackberry-10-je-tu/141343/?xURL=301

[10] (2013) Orodje Qt. Dostopno na: http://qt.digia.com/About-Us/

[11] (2013) QML deklaracijski jezik. Dostopno na: http://qt-project.org/doc/qt-4.8/qdeclarativeintroduction.html

[12] (2014) QML deklaracijski jezik. Dostopno na: http://developer.blackberry.com/native/documentation/cascades

[13] (2014) Operacijski sistem BlackBerry. Dostopno na: http://en.wikipedia.org/wiki/BlackBerry#Operating_system

[14] (2014) Različice OS BlackBerry. Dostopno na: http://en.wikipedia.org/wiki/Research_In_Motion

[15] (2014) BlackBerry operacijski sistem. Dostopno na: http://en.wikipedia.org/wiki/BlackBerry_OS

[16] (2014) JSON. Dostopno na: http://json.org/json-sl.html

[17] (2014) Microsoft Windows Azure. Dostopno na: http://blogs.technet.com/b/microsoft_blog/archive/2010/02/01/windows-azure-general-availability.aspx

[18] (2014) Microsoft Windows Azure. Dostopno na: http://en.wikipedia.org/wiki/Windows_Azure#cite_note-avail-1

[19] (2014) Microsoft Windows Azure. Dostopno na: http://dne.enaa.com/Internet-in-programi/Internet/Windows-Azure-tudi-uradno-v-Sloveniji.html

[20] (2014) Microsoft Windows Azure. Dostopno na:
http://www.diventic.si/Articles/UseCases

[21] (2014) Visual Studio. Dostopno na:
http://msdn.microsoft.com/en-us/library/fx6bk1f4

[22] (2014) .NET orodje. Dostopno na:
http://en.wikipedia.org/wiki/.NET_Framework

[23] (2014) .NET platforma. Dostopno na:
http://msdn.microsoft.com/en-us/library/zw4w595w.aspx

[24] (2014) Oblačno računalništvo. Dostopno na:
http://student.fnm.uni-mb.si/zmocivnik/index.php

[25] (2014) BlackBerry 10 Native SDK. Dostopno na:
http://scrapps.org/blackberry/rimsetup/native.htm

[26] (2014) BlackBerry 10 Native SDK. Dostopno na:
http://developer.blackberry.com/native/downloads/releasenotes_momentics/

[27] (2014) QNX Operacijski sistem. Dostopno na:
http://www.qnx.com/

[28] (2014) Arhitektura BlackBerry operacijskega sistema. Dostopno na:
http://www.skymobilemedia.com/rim/operating-system/architecture-of-blackberry-os.html

[29] (2014) QNetworkAccessManager. Dostopno na:
http://peter.hartmann.tk/blog/2012/02/inside-the-qt-http-stack.html

[30] (2014) QNetworkReplay. Dostopno na:
http://qt-project.org/doc/qt-4.8/qnetworkreply.html

[31] (2014) ASP.NET MVC. Dostopno na:
http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework

[32] (2014) Web API. Dostopno na:
http://msdn.microsoft.com/en-us/library/hh833994

[33] (2014) REST in RESTful. Dostopno na:
http://mauriziostorani.wordpress.com/2008/07/27/rest-representational-state-transfer-and-restful-web-services-concepts-and-examples/

[34] (2014) JavaScript. Dostopno na:
http://www.egradiva.net/drugo/javascript/01_mapa/01_datoteka.html

[35] (2014) Web API. Dostopno na:
http://en.wikipedia.org/wiki/Web_API

[36] (2014) Programski jezik C++. Dostopno na:
http://www.cplusplus.com/