

Razvoj mobilne aplikacije FERI

Maribor, januar 2013

**KAZALO**

1 Uvod 1

2 Pametne mobilne naprave 3

3 MOBILNE PLATFORME 4

3.1 Android, Google 4

3.2 IOS, Apple 6

3.3 Windows Phone, Microsoft 7

3.4 Blackberry OS, RIM 8

3.5 Ostale platforme 8

4 Tržni delež platform 9

4.1 V svetovnem merilu 9

4.2 Razširjenost v Sloveniji 11

5 Vrste mobilnih aplikacij 14

5.1 Responsive spletne strani 14

5.2 Spletne strani prilagojene za mobilne naprave 15

5.3 Aplikacije 17

5.4 Zaključek 18

6 Način razvoja 19

6.1 Native razvoj 19

6.1.1 IOS in ObjectiveC 20

6.1.2 Android in Java 21

6.2 Medplatformski razvoj 22

6.2.1 PhoneGap 22

6.2.2 Rhodes 23

6.2.3 Appcelerator Titanium 23

6.2.4 Corona 23

6.3 Primerjava medplatformskih ogrodij 23

7 Aplikacija FERI 26

7.1 Načrtovanje in funkcionalnosti 26

7.1.1 Funkcionalnosti 26

7.1.2 Wireframes 34

7.1.3 Userflow 36

7.1.4 Podatkovni model 37

7.2 Analiza aplikacij drugih fakultet 40

7.2.1 Cambridge University 40

7.2.2 Harvard University 41

7.2.3 University of Wales, Newport 41

7.2.4 University of Texas at Austin 42

7.2.5 University of Nebrska 42

7.3 Grafični vmesnik aplikacije 44

7.4 Opis uporabljenih tehnologij 47

7.4.1 Appcelerator Titanium 47

7.4.2 Javascript 47

7.4.3 HTML / CSS 48

7.4.4 Podatkovna baza SQLite 49

7.4.5 PHP za strežniški del 49

7.5 Razlike v kodiranju za iOS in Android 49

7.6 Strežniški del razvoja in izmenjava podatkov 51

7.7 Beta testiranje 52

7.8 Analiza uporabe in statistika 54

7.9 Nadaljnji razvoj in dostopnost materialov 54

8 Sklepne ugotovitve 56

**KAZALO SLIK**

Slika 4.1: Graf tržnega deleža mobilnih platform v Sloveniji. 13

Slika 7.1: Zaslonska slika modula »Oglasna deska«. 27

Slika 7.2: Zaslonska slika modula »Urniki«. 28

Slika 7.3: Zaslonska slika modula »Zaposleni«. 29

Slika 7.4: Zaslonska slika modula »Zemljevid«. 30

Slika 7.5: Zaslonska slika modula »Diplome«. 31

Slika 7.6: Zaslonska slika modula »Informacije«. 32

Slika 7.7: Zaslonska slika modula »Nastavitve«. 33

Slika 7.8: Zaslonska slika modula »FERI«. 34

Slika 7.9: Scan ročnih wireframe-ov 1. 35

Slika 7.10: Scan ročnih wireframe-ov 2. 35

Slika 7.11: Scan ročnih wireframe-ov 3. 36

Slika 7.12: Drevesni pogled vseh poti skozi aplikacijo. 37

Slika 7.13: Diagram podatkovnega modela. 38

Slika 7.14: Opis entitete modela zadnje\_diplome. 39

Slika 7.15: Preverjanje ali entiteta že obstaja. 39

Slika 7.16: Testirane aplikacije. 40

Slika 7.17: University of Wales, Newport. 41

Slika 7.18: University of Texas at Austin. 42

Slika 7.19: University of Nebraska. 43

Slika 7.20: Logotip v različnih velikostih. 44

Slika 7.21: Primerjava starega in novega »začetnega zaslona«. 45

Slika 7.22: Primerjava starega in novega »dashboarda«. 45

Slika 7.23: Primerjava stare in nove »oglasne deske«. 46

Slika 7.24: Primerjava starih in novih »informacij«. 46

Slika 7.25: Preverjanje trenutne platforme. 50

Slika 7.26: Ločevanje logike za različne platforme. 50

Slika 7.27: Seznam naročenih naprav v testnem okolju. 51

Slika 7.28: Dashboard v spletni aplikaciji TestFlight. 53

Slika 7.29: Activity okvir v spletni aplikaciji TestFlight. 53

Slika 7.30: Podatki o trajanje seje v spletni aplikaciji TestFlight. 54

**KAZALO TABEL**

Tabela 4.1: Operacijski sistemi, prodan naprav in tržni delež v drugem četrletju 2012. 10

Tabela 4.2: IDC napoved tržnega deleža za 2015. 11

Tabela 6.1: Pregled opisanih ogrodij v tabelarični obliki. 24

**SEZNAM AKRONIMOV**

API Application Programming Interface

GPS Global Positioning System

HTML HyperText Markup Language

JRE Java Runtime Environment

SDK Software Development Kit

XML Extensible Markup Language

WLAN Wireless Local Area Network

# **Uvod**

Pametne mobilne naprave se v zadnjih letih zelo hitro razvijajo, trend njihove uporabe narašča, z dostopnostjo naprav preko naročniških paketov pri mobilnih operaterjih, pa so dostopne tudi študentskemu žepu. V sklopu diplomskega dela se bomo osredotočili na razvoj mobilne aplikacije spletnega portala FERI, katerega študentje Fakultete za računalništvo in informatiko uporabljajo dnevno, a v mobilniku neprijaznem načinu. Aplikacija, ki je nastala kot praktični del diplomske naloge in je tudi javno dostopna ter na voljo za uporabo, je torej namenjena študentom in vsebuje najbolj pogosto uporabljene sklope iz spletne portala in tako ponudi lažji, hitrejši, predvsem pa prijaznejši pregled vsebin spletnega portala.

Po uvodu sledi kratek splošen pregled mobilnih naprav, ki so trenutno na trgu in vrste mobilnih naprav, ki so najbolj razširjene. Večji poudarek bomo namenili različnim načinom razvoja mobilnih aplikacij, saj nam sodobna orodja ponujajo razvoj aplikacije v t.i. nativno aplikacijo, mobilno spletno aplikacijo in nekakšen hibrid obeh.

Prav tako se na trgu pojavlja vedno več različnih mobilnih platform in čeprav analitiki napovedujejo obstoj le treh največjih[[1]](#footnote-1), bomo naslednje poglavje namenili opisu in primerjavi tudi tistih z trenutno manjšim tržnim deležem. Mobilne platforme se med seboj razlikujejo glede operacijskega sistema, svojih zmožnosti in funkcionalnostih, potrebnih programskih jezikov za razvoj aplikacij itd. Razvijanje aplikacije za vse platforme bi bilo potratno tako s stališča časa potrebnega za razvoj in testiranja, kot tudi stroškov in v ta namen smo pregledali analize razširjenosti platform in naprav v Sloveniji, s katerih lahko sklepamo na podoben vzorec uporabe pri študentih, kateri so ciljna skupina naše aplikacije.

Ko pridemo do same odločitve glede razvoja imamo na voljo več različnih pristopov, ki se razlikujejo tako v potrebnem znanju različnih programskih jezikov, kot tudi v času potrebnem za dosego končnega produkta na vseh platformah. Opisali bomo kako lahko obstoječo spletno stran ali aplikacijo prilagodimo za prikaz na mobilnih napravah, razvoj t.i. nativne mobilne aplikacije in pa uporabo medplatformskih orodij, ki nam omogočajo razvoj v enem programskem jeziku za več platform hkrati. Slednjemu pristopu se bomo posvetili največ, saj smo le-tega izbrali tudi za mobilno aplikacijo FERI. Podrobneje bomo opisali različna orodja, ki trenutno obstajajo na trgu, njihove prednosti in slabosti pa bomo opisali v besedilni, kot tudi v tabelarični obliki. Za medplatformski pristop smo se odločili tudi pri razvoju aplikacije FERI, zato bomo podali mnenje in razloge za odločitev tega pristopa in za izbiro specifičnega razvojnega okolja.

Sledi opis praktičnega dela diplomske naloge, kjer bomo šli skozi različne faze nastajanja aplikacije. Načrtovanje vsebuje opis potrebnih funkcionalnosti in specifikacije, žičnati model, potek in opis primerov uporabe, ter podatkovni model. Pogledali si bomo tudi aplikacije drugih fakultet, predvsem ameriških univerz, ter tako dobili občutek o strukturi podobnih mobilnih aplikacij in njihovih funkcionalnostih. Grafični vmesnik se je v času razvoja kar precej spremenil in prikazali bomo njegovo prvo verzijo in končni produkt, ter namenili nekaj besed o tem kako lahko grafična podoba temeljito spremeni uporabnikovo izkušnjo pri uporabi aplikacije. Sledi opis vseh uporabljenih tehnologij in nekaj primerov programske kode, da dobimo občutek o programski zgradbi aplikacije. Ker medplatformski razvoj potrebuje tudi nekaj prilagoditev za vsako platformo posebej, smo v nadaljevanju podali nekaj takih primerov in opisali kako se končna aplikacija razlikuje na različnih platformah. Sledi opis faze beta testiranja, kjer smo izbrali nekaj uporabnikov, ki so predčasno imeli dostop do mobilne aplikacije, ter so podali svoje mnenje, ugotovitve in prve izkušnje pri uporabi mobilne aplikacije, ki smo jih potem upoštevali in dodatno prilagodili nekatere dele aplikacije. Končna aplikacija je bila izdana že pred oddajo diplomskega dela in tako smo imeli možnost vključiti tudi prve odzive uporabnikov oz. študentov, ter analizirati uporabo na posameznih platformah.

V zadnjem poglavju smo predstavili sklepne ugotovitve in smotrnost uporabe medplatformskih ogrodij. Ker je sama izvorna koda aplikacije bila izdana pod odprto kodno licenco, diplomsko delo zaključujemo z možnostjo in spodbudo za nadaljni razvoj.

# **Pametne mobilne naprave**

Pametno mobilno napravo je pravzaprav težje definirati, kot se zdi na prvi pogled. Pomen besede „pameten“ se pri mobilnih napravah namreč skozi razvoj novih mobilnih naprav vedno spreminja. Naprava, ki bi jo uvrstili med „pametne mobilne naprave“ pred petimi leti, verjetno danes več ne bi spadala v to kategorijo. Danes pametni telefoni združujejo telefonijo z uporabo interneta v eni sami napravi, zraven tega pa ponujajo še različne druge funkcije, kot na primer prejemanje elektronske pošte, medijski predvajalnik, koledar opravil in dogodkov, kalkulator in še veliko drugih funkcij. Naštete funkcionalnosti so v večino telefonov že vgrajene, število funkcij telefona pa se dodatno poveča z namestitvijo novih aplikacij, katere so tudi tema te diplomske naloge in bomo v naslednjih poglavjih podrobneje opisali možnosti za njihov razvoj in distribucijo.

Če pogledamo pametne mobilne naprave s stališča strojnih komponent, pa večina naprav vsebuje vsaj digitalno kamero in fotoaparat, GPS navigacijski modul, digitalni kompas, pospeškometer, povezljivost z brezžičnim omrežjem Wi-Fi, bluetooth. Pri nekaterih napravah pa najdemo tudi možnost povezovanja s TV sprejemnikom oziroma večjim monitorjem, **senzor za detekcijo \_\_\_\_\_ (proximity sensor)**, FM radio, žiroskop, **NFC (near field communication)**,... Običajno pa imajo naprave še zaslon občutljiv na dotik, ki poenostavi uporabo opravil, kot je na primer brskanje po spletu.

Firtman[[2]](#footnote-2) je definiral pametno mobilno napravo, kot: »Naprava, ki ima večopravilni operacijski sistem, polno delujoč brskalnik, Wi-Fi in 3G povezljivost, multimedijski predvajalnik.« Dodatno omenja tudi funkcionalnosti, ki smo jih že našteli zgoraj. Wikipedijaprav tako opozarja, da je mejo med navadnimi in pametnimi napravami težko določiti, ter da uradna definicija ne obstaja, največjo razliko pa vidijo v tem, da slednji preko API vmesnika omogočajo razvoj dodatnih aplikacij, ki potem nadgradijo nameščen operacijski sistem z dodatnimi funkcijami in uporabnostjo[[3]](#footnote-3). Večina definicij pa vključuje tudi opis večjih operacijskih sistemov, kot sta Googlov Android in Appleov iOS in zato bomo operacijskim sistemom oz. mobilnim platformam namenili naslednje poglavje, kjer bomo le-te tudi opisali in prikazali možnosti za razvoj dodatnih aplikacij.

# **MOBILNE PLATFORME**

Mobilni operacijski sistem je operacijski sistem, ki nadzoruje pameten mobilni telefon, tablični računalnik, PDA ali kakšno drugo mobilno napravo. Najbolj znani mobilni operacijski sistemi so Android (Google), iOS (Apple), Windows Phone (Microsoft), Symbian (Nokia), BlackBerry OS (RIM), Bada (Samsung), webOS (Hewlett Packard) in manjše linux distribucije, kot sta Maemo in MeeGo. Operacijski sistemi so večinoma lahko nameščeni na različne modele telefonov in z izjemo Applovega iOS, tudi na naprave različnih proizvajalcev, Posamezna naprava pa je v svojem življenjskem ciklu deležna več posodobitev operacijskega sistema.

## Android, Google

Android je odprtokodni programski jezik in operacijski sistem za pametne telefone, ter ostale prenosne naprave. Zgrajen je na Linuxovem jedru. Android je razvilo manjše istoimensko startup podjetje, ki ga je pozneje prevzel Google Inc. leta 2005 in je sedaj tudi najzaslužnejši za nadaljnji razvoj in posodobitve[[4]](#footnote-4). Google je tako tudi ustanovil poslovno združenje več podjetij Open Handset Alliance (OHA), katerega so javnosti predstavili novembra 2007 s prizadevanjem skupnega razvoja odprtih standardov na področju telefonije ter ostalih prenosnih naprav, saj poslovno združenje teži k razvoju inovacij, prav tako pa želi približati te telefone uporabnikom z vedno cenejšimi in boljšimi pametnimi telefoni, ter prenosnimi napravami[[5]](#footnote-5).

Prva verzija, ki je prišla na trg februarja 2009 je vsebovala, le nekaj osnovnih funkcij, kot so budilka, testni prikaz uporabniškega vmesnika, pregledovalnik za internet, kamero, itd. Sledili sta verziji 1.5 (Cupcake) in 1.6 (Donut), ki sta prinesli dodatne funkcionalnosti kot so animacije na ekranu, možnost bluetooth povezave in pa tudi nov Android Market (danes Google Play), kateri je prinesel preglednejši in uporabnejši sistem za prenos programov. Skupaj z OHA so tudi pripravili telefon HTC Hero, s katerim so prvič pritegnili pozornost ostalih večjih proizvajalcev mobilnih telefonov.

Vse android verzije operacijskega sistema so poimenovane po slaščicah in konec leta 2009 so sledile verzije 2.0 (Eclair), 2.2 (Froyo oziroma Frozen Yogurt) in 2.3 (Gingerbread). Z verzijo 3.0 je Google podprl tudi tablične naprave, trenutno aktualna verzija 4.1 (Jelly Bean) je nasledila 4.0 (Ice Cream Sandwich) izdano oktobra 2011, ki sedaj podpira že ogromno novih funkcionalnosti. Poplava naprav, ki imajo nameščen Android operacijski sistem se je zgodila s predstavitvijo naprave HTC Dream. Od drugega četrtletja 2009 do drugega četrtletja 2010 je svetovni tržni delež Android operacijskega sistema zrasel za 850%, od 1.8% do 17.2%. Novembra 2011 pa je Android dosegel 52.5% in ima sedaj konec leta 2012 že 61%[[6]](#footnote-6). Skupaj z Applovim iOS, ki ga bomo predstavili kot naslednjega, torej pokrivata več kot 80% celotnega trga operacijskih sistemov na mobilnih napravah.

**Razvoj**

Nativne aplikacije za Android so napisane v programskem jeziku Java. Android ima svoj virtual machine imenovan Dalvik VM. Vsi java razredi so prevedeni v Dalvik in potem pognani s pomočjo Dalvik vritualne machine.

Za razvoj potrebujemo Java SE Development Kit (JDK), Android SDK in okolje za razvoj (IDE). Za lažji razvoj se priporoča uporaba Eclipse (IDE) in Android Development Tools (ADT) vtičnika za Eclipse za kompilanje in zagon v Android emulatorju.

Za objavo mobilne aplikacije preko uradnega Google Play marketa je potrebno včlaniti v Google Developer program, katerega naročnina znaša enkratnih $25 USD. V primeru, da aplikacijo ponudimo kot plačljivo, pa Google vzame 30%, ki so namenjeni za distribucijske partnerje in stroške vzdrževanja[[7]](#footnote-7).

## IOS, Apple

Appleov iPhone, iPod Touch, iPad in druga generacija Apple TV uporabljajo operacijski sistem imenovan iOS, ki izvira iz Mac OS X. Apple Inc. je prvo verzijo iOS-a, takrat imenovanega še iPhone OS, izdal januarja 2007, z verzijo 2.0 pa so julija 2008 tudi podprli razvoj dodatnih aplikacij. Do takrat je bil razvoj dodatnih funkcionalnosti sicer možen preko spletnih aplikacij, ki do določene mere lahko dostopajo tudi do strojne opreme v telefonu, a pravi razcvet novih aplikacij je iPhone doživel šele z uvedbo Software Development Kit-a (SDK) in pravim nativnim razvojem aplikacij.

V nasprotju z Googlovim Androidom, pa Apple ne dovoljuje uporabo in namestitev operacijskega sistema na naprave drugih proizvajalcev. Podobno filozofijo uveljavljajo tudi pri ostalih svojih produktih. Integracijo programske in strojne opreme postavljajo v ospredje in imajo tako nadzor nad celotnim procesom in kakovostjo končnega produkta.

Apple je operacijski sistem iOS vgradil tudi v njihove druge naprave. Leta 2007 v iPod Touch, leta 2010 v njihov prvi tablični računalnik iPad in pozneje istega leta tudi v Apple TV, ki je namenjen prikazovanju multimedijskih vsebin na televizijskem zaslonu.

Večje posodobitve operacijske sistema iOS si sledijo približno v ciklu enega koledarskega leta, preko programske opreme iTunes, od verzije 5.0 naprej pa tudi preko brezžične povezave. S to verzijo je Apple tudi dokončno omogočil uporabo in posodobitve mobilne naprave brez uporabe računalnika. Zadnja večja posodobitev operacijskega sistema je verzija 6.0, izdana Septembra 2012 in prinaša več kot 200 novih funkcionalnosti.

**Razvoj**

Kot smo že omenili, je bil prvi SDK izdan 6. Marca 2008 in s tem razvijalcem omogočen razvoj nativnih aplikacij tako za iPhone kot tudi za iPod Touch. Razvoj poteka v programskem jeziku Objective C, za razvoj potrebujemo Appleov računalnik z nameščenim MacOSX in programsko opremo Xcode. Testiranje poteka s programom iPhone Simulator, ali na sami napravi za, kar pa moramo biti včlanjeni v iPhone Developer Program, katerega letna članarina znaša $99.00 USD.

Ko razvijalec izda aplikacijo v App Store, ki je edini uraden kanal za prodajo in širitev novih aplikacij, lahko določi poljubno ceno. Apple od vsake prodane aplikacije vzame 30% delež za distribucijo aplikacije, če se pa razvijalec odloči za brezplačno aplikacijo, pa mu ni treba plačati dodatnih stroškov, razen seveda že omenjene letne članarine.

## Windows Phone, Microsoft

Microsoftov mobilni operacijski sistem Windows Phone je naslednik operacijskega sistema Windows Mobile. Windows Phone je bil uradno izdan leta 2010 na mobilni konferenci Mobile World Congress v Barceloni, s katerim je Microsoft naredil nov uporabniški vmesnik imenovan Metro. Microsoft naj bi prodal 1.5 milijonov naprav v prvih šestih tednih od izida Windows Phone 7, a ker so relativno pozno posodobili operacijski sistem in (pre)pozno priključili konkurenci, sedaj njihov delež na trgu operacijskih sistemov znaša le borih 5.2%[[8]](#footnote-8).

Oktobra leta 2010 je CEO Steve Ballmer naznanil 10 novih mobilnih naprav, ki jih bodo proizvedli HTC, Dell, Samsung in LG. Pozneje so se proizvodnji mobilnikov priključili še Acer, Fujitsu in ZTE. Naprave so na voljo preko 60 ponudnikov mobilne telefonije v 30 državah.

Na naslednji mobilni konferenci leta 2011 je Steve Ballmer predstavil novo pomembnejšo verzijo operacijskega sistema 7.5, s kodnim imenom „Mango“. Nova verzija je odpravila večje pomanjkljivosti z uvedbo nove mobilne verzije Internet Explorerja 9, možnostjo večopravilnosti z ostalimi nativnimi aplikacijami, integracijo s socialnimi omrežji in drugimi večjimi popravki. Verzija v letu 2012, izdana pod imenom „Tango“, je tudi znižala strojne zahteve za poganjanje operacijskega sistema. Mobilniki s slabšo strojno opremo imajo tako onemogočene določene funkcionalnosti, Microsoft pa s tem cilja tudi na naprave nižjega cenovnega razreda in prodoru na nove trge.

Steve Balmer in Stephen Elop sta 11 Februarja 2011 uradno oznanil partnerstvo Microsofta in Nokie, kjer bo Windows Phone postal primarni operacijski sistem za naprave, ki jih bo proizvedla Nokia. Dogodek je oznanjal nastanek novega globalnega mobilnega ekosistema in napovedoval tekmovanje z Androidom in iOS za prevlado na trgu mobilnih operacijskih sistemov.

**Razvoj**

Microsoft je razvoj za Windows Phone 7 vključil v svoj že obstoječ .NET framework in s tem poskrbel za enostaven prehod razvijalcev, ki so z platformo in programskim jezikom C# že seznanjeni. Silverlight in XNA predstavljata dve večji ogrodji v .NET okolju. Medtem ko je Silverlight bolj v uporabi za razvoj poslovnih aplikacij in 2D grafiki, je slednji namenjen 3D grafiki in razvoju iger. Za razvoj potrebujemo Windows Phone Developer Tools, ki je že vključen v paketu Visual Studio 2012 Express, Windows Phone Emulator, XNA Game Studio 4.0, Microsoft Expression Blend za Windows Phone, Silverlight, ter .NET Framework 4.

## Blackberry OS, RIM

BlackBerry OS je mobilni operacijski sistem podjetja Reasearch In Motion (RIM) in se uporablja v vseh telefonih BlackBerry istega podjetja. Prvi BlackBerry telefon je z optimiziranim načinom za prejemanje elektronske pošte in tipkovnico QWERTY postal uporaben predvsem v poslovnem svetu. Spletni brskalnik, ki je v napravah BlackBerry, se sooča z določenimi omejitvami (npr. prikazovanje naprednejših HTML5 lastnosti in SVG slik), vendar podjetje RIM pričakuje, da bodo odpravili večino omejitev z naslednjo izdajo OS in z uporabo brskalnika na osnovi WebKit-a. 3D grafična podpora je realizirana z uporabo knjižnice OpenGL ES, ki je vključena v BlackBerry SDK, in omogoča izgradnjo bogatih 3D vsebin[[9]](#footnote-9) [[10]](#footnote-10).

**Razvoj**

Aplikacije za BlackBerry razvijamo s pomočjo orodja BlackBerry Web Development ali z programskim jezikom Java. Za razvoj tako potrebujemo Sun JDK (angl. Java Development Kit), popularno razvojno orodje Eclipse, potreben pa je tudi BlackBerry dodatek za Eclipse in BlackBerry JDE.

## Ostale platforme

Na trgu so še ostali operacijski sistemi, kot na primer Open webOS (Hewlet Packard), Bada (Samsung), GridOS (Fusion Garage), Mer project (odprtokoden), MeeGo (odprtokoden), Brew (Qualcomm), LiMo (LiMo Foundation) in še nekateri manjši odprtokodni projekti. Kot bomo videli v nadaljevanju, pa ti operacijski sistemi ne predstavljajo večjega tržnega deleža, še posebej ne na slovenskem mobilnem trgu, zato jih tukaj tudi ne bomo posebej predstavljali.

# **Tržni delež platform**

## V svetovnem merilu

Po poročanju mednarodne korporacije IDC (International Data Corporation), ki je eno vodilnih globalnih podjetij na področju zbiranja in analize podatkov informacijske / telekomunikacijske industrije, pokrivata Android in iOS 85 % delež vseh prodanih telefonov v drugem četrtletju 2012. Medtem sta BlackBerry in Symbian, pionirja in nekdaj vodilna na trgu mobilnih telefonov, padla pod 5 odstotni delež. Če tako sedaj pogledamo tržni delež mobilnih operacijskih sistemov v drugem četrtletju 2012, je Android močno zasidran na prvem mestu s 68.1 odstotki, Applov iOS na drugem mestu s 16.9%, medtem ko imata BlackBerry in Symbian le 4.8% in 4.4%. Blizu za njimi je s 3.5% Windows Phone 7 (oziroma Window Mobile), kateremu pa, kot bomo videli v nadaljevanju analize, napovedujejo svetlo prihodnost[[11]](#footnote-11).

„Android nadaljuje s streljanjem na vse cilindre“ je dejal Ramon Llamas[[12]](#footnote-12), višji analitik pri IDC. „Trg je dobro sprejel modele vodilnih partnerjev Android platforme, cene pa so bile na dosegu uporabnikov z različnimi proračunskimi zmožnostmi. Uporabniška izkušnja, h kateri sta pripomogla tako Google, kot tudi njegovi partnerji je močno povečala uporabnost Android pametnih telefonov.“

„Trg mobilnih operacijskih sistemov je zdaj brez dvoma bitka med dominantnim Androidom in iOS“, je dejal Kevin Restivo[[13]](#footnote-13), prav tako višji analitik pri IDC-ju. „Ker pa je na mobilnem trgu še vedno ogromno uporabnikov, ki ne uporabljajo pametnih telefonov, je bitka za prevlado še daleč od končane. Čeprav bodo prizadevanja vedno težja, saj se prodor pametnih telefonov konstantno povečuje, je trg še vedno pripravljen na spremembo tržnih deležev trenutno nižje rangiranih operacijskih sistemov.“

Tabela 4.1: Operacijski sistemi, prodan naprav in tržni delež v drugem četrletju 2012[[14]](#footnote-14).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operacijski sistem** | **Q2 2012 Naprav** | **Q2 2012 Tržni delež** | **Q2 2011 Naprav** | **Q2 2011 Tržni delež** | **Sprememba** |
| **Android** | **104.8** | **68.1%** | **50.8** | **46.9%** | **106.5%** |
| **iOS** | **26.0** | **16.9%** | **20.4** | **18.8%** | **27.5%** |
| **BlackBerry OS** | **7.4** | **4.8%** | **12.5** | **11.5%** | **-40.9%** |
| **Symbian** | **6.8** | **4.4%** | **18.3** | **16.9%** | **-62.9%** |
| **Windows Phone 7 / Windows Mobile** | **5.4** | **3.5%** | **2.5** | **2.3%** | **115.3%** |
| **Linux** | **3.5** | **2.3%** | **3.3** | **3.0%** | **6.3%** |
| **Ostali** | **0.1** | **0.1%** | **0.6** | **0.5%** | **-80.0%** |
| **Skupaj** | **154.0** | **100.0%** | **108.3** | **100.0%** | **42.2%** |

Enote so izraže v milijonih.

Analitiki pa napovedujejo da rast, ki je bil Googlov Android v zadnjem obdobju deležen, ne more trajati v nedogled in leto 2012 napovedujejo kot zadnje leto, kjer bo Android dosegel svoj vrhunec. Predvidoma bi se številka prodanih pametnih mobilnikov v letu 2012 naj ustavila pri 1.8 milijarde, kar je le malenkost več kot v letu 2011, ko je bilo prodanih 1.7 milijarde pametnih telefonov.

Paradiranje pametnih telefonov letos ne bo tako živahno kot v preteklost, navaja IDC. Uporabniki bodo sicer še vedno prehajali iz enostavnejših naprav na bolj sposobne pametne mobilnike, a z bolj umirjenim, predvsem pa konstantnim tempom. Podobno rast pametnih mobilnih naprav lahko pričakujemo vse do leta 2016, drugače pa napovedujejo za tržni delež pri operacijskih sistemih. Konec leta 2012 bi naj bilo prelomno, kjer bo Android dosegel najvišjo točko, nato pa bo njegov delež in delež iOS-a začel padati. Predvsem Androidu napovedujejo padec na slabih 53%, kjer se bo Microsoft s svojo Window Platformo povzpel iz trenutnih 5.2% na kar 19.3%. Če so napovedi IDC-ja točne bi tako Microsoft do leta 2016 naj prevzel drugo mesto med mobilnimi operacijskimi sistemi, saj bi Applov iOS zdrsnil na 19.2 odstotka[[15]](#footnote-15).

Mogoče se zdijo napovedi IDC-ja pretirane in pa predvsem preveč natančne, kar se tiče rasti posameznikov v bitki za mobilni trg, a ko pogledamo podobne analize drugih vodilnih korporacij, kot je na primer Gartner, lahko opazimo podoben vzorec. V spodnji tabeli bomo tako primerjali analizi IDC-ja in Gartnerja:

Tabela 4.2: IDC napoved tržnega deleža za 2015.

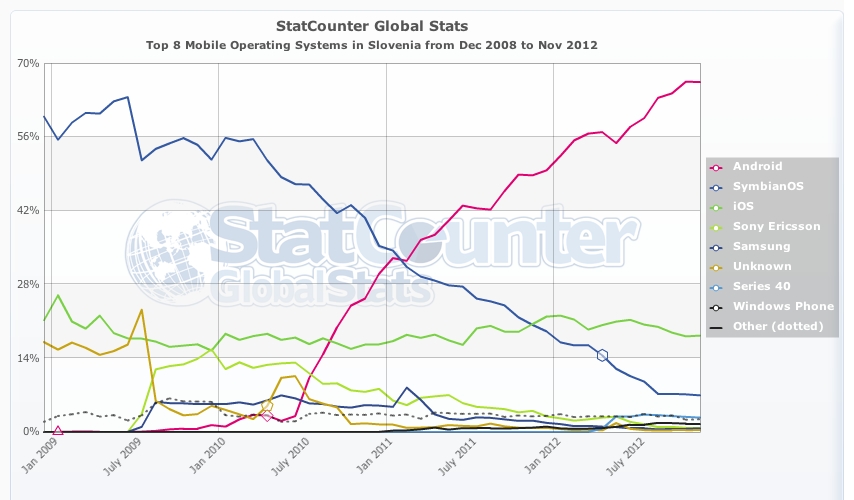
| Operacijski sistem | Napoved tržnega deleža za 2011 | Napoved tržnega deleža za 2012 | Napoved tržnega deleža za 2015 |
| --- | --- | --- | --- |
| Android | 38.5% | 49.2% | 48.8% |
| BlackBerry | 13.4% | 12.6% | 11.1% |
| iOS | 19.4% | 18.9% | 17.2% |
| Symbian | 19.2% | 5.2% | 0.1% |
| **Windows Phone 7 / Windows Mobile** | **5.6%** | **10.8%** | **19.5%** |
| Others | 3.9% | 3.4% | 3.3% |

**IDC-jeva napoved: skoraj 6% tržni delež v 2011, 21% do leta 2015[[16]](#footnote-16)**

**Gartner-jeva napoved: skoraj 6% v 2011, 11% v letu 2012, 20% do leta 2015[[17]](#footnote-17)**

## Razširjenost v Sloveniji

Ker pa pri tej diplomski nalogi razvijamo aplikacijo, ki je popolnoma namenjena slovenskemu trgu oziroma celo bolj specifično na študente, moramo bolj natančno preučiti trend uporabe pametnih telefonov in mobilnih operacijskih sistemov pri nas. Ker seveda korporacije kot sta IDC in Gartner slovenskega trga ločeno ne analizirata, si lahko pomagamo z članki objavljenimi v domačih revijah in spletnih straneh[[18]](#footnote-18), kot tudi s spletnim portalom podjetja StatCounter[[19]](#footnote-19), ki spremlja statistiko dostopov preko 3 milijonov strani po celem svetu. Poglejmo najprej delež spletnih brskalnikov. Trenutno trije najbolj priljubljeni mobilni spletni brskalniki so pri Androidu privzeti Browser, Opera, ki jo lahko uporabljamo tako na pametnih telefonih kot na bolj običajnih, in za iPhone privzeti Safari (podjetje ločeno obravnava delež naprave iPod Touch, ki sicer uporablja isti brskalnik kot iPhone). Na grafu deležev brskalnikov je gotovo najbolj zanimiva strma rast deleža Androida, ki iz meseca iz mesec za seboj pušča ostale tekmece. Tudi podatki o mobilnih operacijskih sistemih kažejo, da pri nas več kot polovica uporabnikov mobilnega spleta uporablja Android (dobrih 63 % v avgustu 2012), sledijo jim iOS uporabniki (okrog 20 %) in Symbian uporabniki (dobrih 7 %). Slednjih je iz meseca v mesec manj, iOS drži dokaj konstanten delež (nihanja med 20 in 25 % po posameznih mesecih), medtem ko uporaba Android operacijskega sistema izjemno narašča. Avgusta 2010 je tržni delež znašal le slabih 20 %, leto za tem se je povzpel na 45 %, sedaj pa že prevladuje z dobrimi 63 %. Iz grafa je razvidno tudi, da je Android po deležu uporabe iOS prehitel konec leta 2010, kar lahko povežemo z akcijskimi prodajami Android mobilnih naprav pri večjih mobilnih operaterjih pri nas.



Slika 4.1: Graf tržnega deleža mobilnih platform v Sloveniji.

Statistika nam torej nazorno prikaže, da sta iOS in Android z skoraj 85 % najbolj priljubljena operacijska sistema uporabnikov pametnih mobilnih telefonov pri nas. Nobene izmed teh dveh platform pa seveda ne smemo zanemariti, zato smo se pri mobilni aplikaciji FERI odločili, da je potrebno podpreti obe prevladujoči platformi.

# **Vrste mobilnih aplikacij**

Najbolj pomembno vprašanje v mobilni poslovni strategiji podjetja je, ali je treba pripraviti kaj posebnega za mobilne platforme. Zadnje čase namreč veliko podjetij razvija lastne mobilne aplikacije, čeprav le ta nima neke dodane vrednosti napram že obstoječi spletni strani, ki predstavlja podjetje in njihove storitve. Tako se zgodi, da podjetje objavi aplikacijo, ki funkcionalno predstavi le novice iz spletnega mesta, kontaktne informacije in njihove poslovne enote. Gre za podatke, ki so že dosegljivi preko spletne strani in ne potrebujejo namenske aplikacije. Podjetja bi v takih primerih lahko bolj smiselno razmislila o razvoju oziroma prilagoditve spletne strani za prikaz na mobilnih napravah (t.i. responsive spletne strani). Tako pa se mnogokrat odločajo za namensko aplikacijo, ker tako narekuje trenutni trend in pri tem pozabijo da bi lahko na lažji, kot tudi verjetno cenejši način prišli do enakega rezultata. Kakšne so torej razlike med „normalnimi“ (tukaj so mišljene spletne strani za namizne računalnike), „responsive“ spletnimi stranmi, stranmi prilagojene za mobilne naprave in namenskimi aplikacijami, bomo opisali v nadaljevanju, skupaj s prednostmi, slabostmi, strojno podporo in možnosti distribucije.

## Responsive spletne strani

„Responsive“ (ang. odzivna) spletna stran se dinamično prilagaja glede na trenutno ločljivost in velikost zaslona. Uporabniški vmesnik se spreminja, skriva/prikazuje določene elemente in se avtomatsko prilagaja, tako da uporabniku zagotovi popolnoma funkcionalno in optimalno spletno izkušnjo na vseh napravah. Tako imamo spletno stran predstavljeno na namiznih računalnikih, mobilnikih in tabličnih računalnikih, medtem ko je potrebno razviti in vzdrževati le eno različico programske kode.

**Prednosti**

* Združljivost: Ker se strani prikazujejo v brskalniku je združljivost pogojena z podporo samega brskalnika, kateri pa sedaj že zelo dosledno podpirajo HTML5 in CSS3 standarde. Seveda nekateri bolj, drugi manj in tako so vseeno potrebne manjše prilagoditve.
* Razvoj: Verjetno največji plus napram nativnim in medplatformskim aplikacijam. Ko odkrijemo kakšno nepravilno delovanje lahko spletno stran enostavno popravimo na spletnem strežniku in vsi uporabniki bodo od takrat naprej dostopali do popravljene verzije.

**Slabosti**

* Strojna podpora: Je omejena, saj preko spleta vsaj zaenkrat ne moremo dostopati do nekaterih strojnih komponent kot je na primer kamera.
* Vmesnik: Platforme imajo začrtane smernice katerih bi se naj aplikacije držale in bi tako nekako zagotavljale enovitost in bi se uporabniki počutili takoj domače v vsaki novi aplikaciji. V primeru responsive spletnih strani to načeloma ni mogoče, saj bi potem morali prilagajati vmesnik ne samo za različne ločljivost, pač pa tudi za različne platforme. Tako ima uporabnik vedno občutek, da se nahaja na spletu in ne v aplikaciji, čeprav mu že spletna stran lahko ponudi velikost funkcionalnosti.

## Spletne strani prilagojene za mobilne naprave

V svetu razdrobljenih mobilnih naprav in operacijskih sistemov je splet edina stalnica. Splet je edina možnost, ki bo delovala na vseh napravah, operacijskih sistemih in platformah. Z napredkom HTML, CSS in Javascript tehnologij nam spletna aplikacija omogoča uporabniško izkušnjo, ki je že danes zelo podobna nativnim aplikacijam, kar pa se s hitrim razvojem tudi izboljšuje. Razvoj mobilne spletne aplikacije predstavlja način, kako podpreti na vse mobilne platforme hkrati. Spletni brskalnik (s podporo zadnjim tehnologijam) deluje na vseh platformah in tako imamo tudi zagotovljeno delovanje na vseh napravah. Seveda pa ima tudi takšen razvoj svoje prednosti in slabosti, ki jih bomo opisali v nadaljevanju.

**Prednosti**

* Združljivost: Spletne mobilne aplikacije niso omejene na platformo ali operacijski sistem, saj tečejo znotraj spletnega brskalnika, kateri pa je nameščen na vseh napravah. Spletni brskalniki po večini tudi podpirajo spletne standarde, ki jih predpisuje W3C tako da spletna stran prilagojena za mobilne naprave deluje povsod enako.
* Razvoj: Ko gradimo spletno mobilno aplikacijo v osnovi razvijamo samo za eno platformo, splet. Nismo pa seveda, tako kot pri nativnih aplikacijah, omejeni na en programski jezik ampak lahko izbiramo med različnimi jeziki kot so Java, PHP, Python,... Ker so ti programski jeziki tudi že dlje v uporabi, pa tudi lažje najdemo dobre razvijalce.
* Vzdrževanje: Podobno kot vzdržujemo nativno aplikacijo za eno platformo, vzdržujemo tudi spletno verzijo aplikacije, ampak tu gre seveda samo za eno verzijo za vse platforme skupaj.
* Distribucija: Omogoča nam hitrejše posodabljanje aplikacije z popravki in nadgradnjami, saj ni potrebno objaviti aplikacije preko dolgotrajnega postopka odobritve nove verzije (tukaj imamo v mislih predvsem Appleov App-Store, kjer lahko odobritev aplikacije ali nove različice traja tudi po več tednov).

**Slabosti**

* Strojna podpora: Medtem kot spletna aplikacija lahko dostopa do nekaterih strojnih komponent oziroma senzorjev, kot so na primer GPS, pospeškometer ali žiroskop, pa aplikacija nima dostopa do mikrofona ali kamere na napravi. Za večino primerov to ne predstavlja ovire, če pa potrebujemo tovrstne dostope, pa tak način razvoja ne pride v poštev.
* Vmesnik: Uporabniški vmesnik je sicer tekoč tudi na spletnih aplikacijah, a ne bo nikoli mogel doseči perfekcije, ki jo lahko dosežemo z optimalnim razvojem na vsaki platformi. Tukaj je spet odvisno kako pomemben je to za podjetje oz. njihovo storitev.

## Aplikacije

Ena izmed ugotovitev, ki so jo objavili v članku na spletni strani [www.useit.com](http://www.useit.com/)[[20]](#footnote-20), kjer se ukvarjajo z analizami uporabniške izkušnje, pravi: **„While a mobile site is good, a mobile app is even better. We measured a success rate of 76% when people used mobile apps, which is much higher than the 64% recorded for mobile-specific websites.“**

Seveda je razvoj za aplikacije dražji in kompleksnejši od razvoja spletne strani prilagojene mobilniku. Mobilne naprave so namreč zelo fragmentirane, kar otežuje že začetni razvoj, s poplavo novih mobilnih naprav pa tudi nadaljnje testiranje in vzdrževanje. Trenutno obstajajo štiri večje platforme na trgu pametnih mobilnikov: Android, iOS, RIM in Window Phone 7. V primeru tabličnih računalnikov je še malce bolj razdrobljeno. Tam sta Android in iOS trenutno največja igralca v igri, RIM ima sicer manjši delež, Amazonov Kindle Fire pa sicer poganja Android, ampak je tako zelo modificiran, da je potreben lasten market za distribucijo aplikacij. Da se stvari še dodatno zakomplicirajo, je na trg vstopil še Windows Phone OS s svojo hibridno politiko za mobilno napravo in namizni računalnik.

Razvoj za vse te platforme je torej izziv, še zlasti za razvijalce ali podjetja z omejenimi sredstvi. Tukaj pridejo v igro medplatformska orodja, kot so Appcelerator Titanium, Rhodes in PhoneGap. Kot vsaka razvojna strategija ima seveda tudi svoje prednosti in slabosti.

**Prednosti**

* Vmesnik: Je večinoma v skladu s priporočili platforme saj uporabljamo nativne gumbe in elemente.
* Strojna podpora: Imamo popoln dostop do vseh komponent telefona in tako ni omejitev glede razvoja in uporabe vseh strojnih funkcionalnosti mobilnika.
* Distribucija: Poteka preko uradnih marketov na katere so uporabniki navajeni in hitro najdejo nove aplikacije za namestitev. Pravtako lahko preko marketa enostavno ponudimo plačljive aplikacije ali kasneje plačevanje znotraj aplikacije.

**Slabosti**

* Vzdrževanje: Predstavlja v primeru iOS veliko slabost, saj smo v primeru tudi najmanjšega popravka aplikacije podvrženi ročnemu preverjanju aplikacije, kar lahko traja tudi nekaj dni.
* Združljivost: Tukaj ni nujno, da združljivost uvrstimo med slabosti. Vsaj v primeru iOS platforme to ne drži, saj je poskrbljeno za kompatibilnost med napravami. V primeru Androida je pa to večji problem saj imamo android nameščen na številnih napravah z različnimi zasloni, ločljivostmi in strojnimi komponentami.
* Razvoj: Za optimalne rezultate je potreben za vsako platformo posebej. Imamo pa tudi t.i. medplatformska ogrodja, ki jih bomo opisali v naslednjih poglavjih.

## Zaključek

Če bi morali na kratko povzeti za katero prisotnost na mobilnih napravah naj se podjetje odloči, bi lahko povzeli z naslednjo trditvijo: Če ima podjetje dovolj sredstev in ima namenska aplikacija dejansko neko dodano vrednost napram mobilni spletni strani, je razvoj lastne aplikacije smiselna odločitev. Vse raziskave potrjujejo, da imajo uporabniki večjo storilnost in boljšo uporabniško izkušnjo z namenskimi aplikacijami. Ko imamo opravka z majhnimi zasloni, počasnimi povezavami, težjim tipkanjem in manjšo natančnostjo je še toliko bolj pomembno, da je aplikacija v celoti prilagojena majhni napravi. Kar je pa seveda dosti težje doseči znotraj mobilnega brskalnika.

V primeru, da ima podjetje omejena sredstva ali ima namen s spletno aplikacijo le predstaviti že obstoječo vsebino (brez dodane vrednosti), ki se nahaja na spletni strani, pa razvoj dodatne aplikacije ni smiseln. Prilagojena spletna stran, naj si bo to posebna verzija spletne strani ali responsive stran, je v takem primeru mnogo boljša odločitev. Tehnologije, ki se uporabljajo znotraj modernih mobilnih brskalnikov, so dovolj napredovale in lahko z dokaj minimalnimi stroški in posegi na obstoječo stran dosežemo prisotnost podjetja na mobilnih platformah, ki pa je v današnjem času z vedno večjim deležem mobilnih uporabnikov, še kako pomembna.

# **Način razvoja**

Ko pridemo do razvoja programske opreme obstaja večno vprašanje ali je bolje voditi ločen razvoj za vsako platformo posebej (naj si to bo Microsoft Window, Linux, MacOS ipd. v primeru namiznih operacijskih sistemov, ali pa Android, iOS ipd. pri mobilnih napravah) ali pa uporabiti orodja katera omogočajo ti. razvoj „napiši enkrat, poženi povsod“ (primer Jave na namiznih računalnikih ali medplatformksih ogrodij za mobilne naprave). Oba načina razvoja imata svoje prednosti in slabosti, tako da se je pogosto težko odločiti katera smer je boljša, velja pa, da je odločitev odvisna od zahtevnosti in namena aplikacije.

Ko se vprašamo kaj dejansko pomeni „boljše“ ugotovimo, da lahko ta relativni pojem definiramo na več načinov. V določenih primerih je lahko zelo pomembno ali je razvoj dovolj hiter in ali že z malo truda dobimo želene rezultate. Ali se končen produkt dovolj hitro izvaja na vseh platformah, je dovolj robusten, ali lahko posamezno platformo izrabimo v celoti (npr. uporaba vseh strojnih zmožnosti naprave) in na koncu ali je končen produkt enostaven za nadaljnje vzdrževanje. Lahko bi seveda našli še več takih primerov, ki so od aplikacije do aplikacije različni, a dejstvo je da bo ena izmed teh definicij vplivala na končno odločitev kateri način razvoja izbrati.

Eden izmed najbolj znanih medplatformskih jezikov oziroma platform je Java. Ker Java kot platforma v operacijskem sistemu iOS ni podprta, seveda ta ne predstavlja ene izmed možnosti za tak način razvoja. Obstajajo pa ostala orodja kot so AdobeAir, PhoneGap, Appcelerator Titanium ipd. katere bomo v nadaljevanju tudi podrobneje pogledali, a vsaka izmed teh se razlikuje v določenih točkah, katere bomo medsebojno primerjali.

## Native razvoj

Native razvoj je izraz za programski razvoj, kjer razvijalec uporablja programski jezik in orodja za točno določeno platformo. Tako recimo razvoj za iOS pomeni uporabo programskega jezika ObjectiveC in orodja Xcode. V primeru razvoja za Android uporabljamo Javo in orodje Eclipse, medtem ko razvoj za BlackBerry in Window Phone poteka spet v drugem jeziku in orodju.

Velik plus takega razvoja je zmožnost v celoti uporabiti vse strojne in programske zmožnosti platforme. To pa seveda pomeni strokovnjaka za vsako platformo posebej, kar pogosto pomeni za podjetje zaposlovanje večjega števila razvijalcev.

Velikokrat se kot posledica razvoja za več platform slej ko prej pojavita dve (oz. več) različnih verzij mobilne aplikacije tako v številu funkcionalnosti, kot v zmožnosti njenega vzdrževanja. Tako zahtevnost, kot tudi znanje in izkušanje posameznih razvijalcev so različne, tako da so pogosto aplikacije pripravljene za produkcijo ob različnih časih oziroma z različnim naborom funkcionalnosti.

Veliko ljudi seveda zagovarjajo dejstvo, da lahko le z nativnim razvojem dosežemo aplikacijo katera bo popolnoma optimirana in pripravljena za vsako platformo posebej. S tem bi se seveda strinjal, a v določenih primerih to ni najbolj ključen faktor za končni uspeh mobilne aplikacije.

### IOS in ObjectiveC

Arhitektura iOS sestoji iz štirih plasti. Osnovni operacijski sistem (angl. Core OS) vsebuje Mach jedro, gonilnike in nizkonivojske UNIX vmesnike6. Njegova naloga je upravljanje s procesi in nitmi, pomnilnikom, mreženjem, datotečnim sistemom, vhodno-izhodnimi napravami in drugimi nizkonivojskimi storitvami.

Plast osnovnih storitev (angl. Core Services) vsebuje najbolj osnovne sistemske storitve, ki jih neposredno ali posredno uporabljajo vse aplikacije. V tej plasti najdemo ogrodja za sistemske nastavitve, trgovino z aplikacijami, mobilne storitve in telefonijo, knjižnice programskih jezikov, sistemske dogodke, lokacijske storitve, shranjevanje podatkov in stikov idr.

Medijska plast (angl. Core Media) je odgovorna za vse grafične, zvočne in video tehnologije operacijskega sistema. Zajema tehnologije za dva in tridimenzionalno grafiko, animacijo, zaslonske pisave, zvočni in video predvajalnik in snemalnik, pozicijski zvok, brezžično predvajanje idr.

Plast Cocoa Touch predstavlja osnovno filozofijo operacijskega sistema iOS. Medtem ko so prve tri plasti v svoji zgradbi podobne tistim v Mac OS X, je naloga Cocoa Touch definirati infrastrukturo aplikacij, ki delujejo na zaslonih na dotik. Cocoa Touch vsebuje osnovne vizualne gradnike, podporo večopravilnosti in obveščanju, tiskalniške gonilnike, P2P storitve, podporo zunanjim zaslonom ipd.

Razvojno orodje iOS SDK vsebuje vse vmesnike, orodja in vire, ki jih potrebujemo za razvoj iOS aplikacij na računalniku. Vmesniki in paketi so večinoma na voljo v obliki programskih ogrodij, ki vsebujejo knjižnice in druge programske vire, in jih povežemo v aplikacijo, ko jih potrebujemo.

Programska oprema zajema:

• integrirano razvojno okolje Xcode,

• orodje Interface Builder za vizualno izgradnjo uporabniškega vmesnika,

• orodje Instruments, ki nam omogoča analizo in razhroščevanje aplikacije med njenim izvajanjem,

• iOS simulator, s katerim lahko simuliramo delovanje aplikacije na računalniku in

• razvijalsko knjižnico, ki vsebuje dokumentacijo razvoja programske opreme.

Programsko ogrodje podpira razvoj domorodnih aplikacij. Aplikacije so napisane v enem izmed ali kombinaciji jezikov Objective-C, C++ in C ter uporabljajo knjižnice in vmesnike operacijskega sistema in druge zunanje knjižnice.

Ogrodje iOS SDK nam omogoča popoln razvoj aplikacij in preizkus teh aplikacij v simulatorju. Če želimo aplikacije preizkusiti tudi na mobilnih napravah, potrebujemo letno naročnino na Apple Developer račun, ki nam omogoča tudi distribucijo aplikacij v App Store.

### Android in Java

Arhitektura operacijskega sistema je zgrajena iz več komponent. Osnovne sistemske storitve, kot so varnost, upravljanje s pomnilnikom in procesi, mreženje in gonilniki, so prepuščene Linux jedru. Jedro skrbi tudi za navidezno napravo Dalvik, ki je temeljna komponenta Android izvajalnega okolja za aplikacije.

Android vsebuje vrsto knjižnic, ki nudijo dostop do različnih komponent operacijskega sistema. To so knjižnice za delo s programskimi jeziki, medijske knjižnice za grafiko, zvok in video, pogon za spletno brskanje (WebKit), knjižnice za dvo in tridimenzionalno grafiko, knjižnice za zaslonske pisave ter pogon za relacijske podatkovne baze (SQLite). Do knjižnic operacijskega sistema dostopamo preko aplikacijskega ogrodja, ki nam omogoča razvoj programske opreme.

Operacijski sistem že v osnovi vsebuje tudi prenaložene aplikacije za elektronsko pošto, kratka sporočila (SMS), koledar, zemljevide, brskanje po internetu, stike idr. Skupaj z morebitno programsko opremo tretjih oseb tvorijo aplikacijski sloj.

Android SDK deluje na operacijskih sistemih Windows, Linux in Mac OS X. Za razvoj potrebujemo Java izvajalno okolje (JRE) in razvojno okolje po naši izbiri. Najbolj pogosta je uporaba uradnega vtičnika za razvojno okolje Eclipse. Ta vtičnik omogoča izdelavo aplikacije in uporabniškega vmesnika, testiranje aplikacije v simulatorju ter izvoz aplikacije na napravo.

Aplikacije za Android so napisane v programskem jeziku Java. Celotna programska koda in viri so zapakirani v Android paketu (datoteka s končnico .apk), ki predstavlja aplikacijo.

## Medplatformski razvoj

Medplatformski razvoj nam omogoča zapis izvorne kode v enem jeziku z isto programsko kodo in potem to pognati na različnih platformah. V teoriji to sicer drži, a kot bomo videli pri praktičnem primeru so vseeno potrebna rahla odstopanja in prilagoditve. Prednost takega pristopa prav gotovo predstavlja ravno zmožnost razvoja in popravljanja enotne izvorne kode. Tako so morebitni kasnejši popravki na voljo za vse platforme, potrebnih pa je tudi manj razvijalcev za razvoj in vzdrževanje. Ker kar nekaj medplatformskih orodij deluje na osnovi spletnih programskih jezikov (HTML, CSS, Javascript) tako tudi lažje najdemo razvijalce, saj je za prehod na razvoj za mobilne platforme potrebno dosti manj znanja in učenja novih tehnologij.

### PhoneGap

PhoneGap je razvilo podjetje Nitobi, katerega je pozneje oktobra 2011 prevzelo podjetje Adobe[[21]](#footnote-21). Z uporabo spletnih tehnologij kot so Javascrip, HTML5 in CSS3 nam omogoča razvoj nekakšne hibridne aplikacije. Aplikacije za prikaz namreč uporabljajo t.i. Web-view, namesto platformskih knjižnic za prikaz uporabniškega vmesnika. Web-view lahko definiramo kot nekakšen prikaz strani, vendar brez uporabe vrstic, ki so klasične za brskljalnik. Uporabnik tako niti ne opazi, da gre za prikaz spletne strani oz. spletne aplikacije. Končna aplikacija pa vendarle je videti kot samostojna aplikacija, saj jo namestimo preko uradnih distribucijskih kanalov in se nato tudi namesti in posodablja kot klasična aplikacija. Omogočen pa nam je tudi dostop do API metod platforme na kateri poganjamo aplikacijo.

PhoneGap trenutno podpira razvoj za Apple iOS, Google Android, HP webOS, Microsoft Windows Phone, Nokia Symbian OS in RIM BlackBerry 5/6/7.

### Rhodes

Rhodes je odprtokodno ogrodje, izdano pod MIT licenco, podjetja Motorola. Za razvoj uporabijo Model-View-Controller, pri čemer je controller napisan v Ruby-u, view pa v HTML-ju.

Ogrodje podpira operacijske platforme iOS, Android, Windows Mobile, Research in Motion (BlackBerry) in Windows Phone 7, medtem ko so podporo za Symbian ukinili po verziji Rhodes 1.2.

### Appcelerator Titanium

Appcelerator Titanium je medplatformsko ogrodje za razvoj mobilnih, tabličnih in namiznih aplikacij z uporabo spletnih tehnologij. Razvil ga je Appcelerator Inc. in je za uporabo na voljo od decembra 2008. Podpora za razvoj iPhone in Android je bila omogočena junija 2009, podpora za razvoj na iPad pa leto kasneje. Gre za zelo priljubljeno ogrodje, saj lahko spletni razvijalci, ki načeloma že imajo znanje programske jezika javascript, le-to uporabijo sedaj pri razvoju mobilnih aplikacij. Potrebno je le spoznavanje Titanium API-ja in hitro lahko pridemo do svoje prve aplikacije. Titanium dokumentacija je zelo dobra, na njihovi uradni strani pa najdemo tudi veliko primerov delujoče kode in video posnetkov nekaterih pogostih problemov in pristopov pri razvoju. Titanium ponuja tudi vključitev dodatnih modulov, ki jih najdemo na uradnem marketu[[22]](#footnote-22). Najdemo tako brezplačne kot tudi plačljive module, ki nam dodatno še dodatno povečajo paleto možnosti uporabe ogrodja.

### Corona

Corona SDK je ustvaril Walter Luh, so-ustanovitelj Corona Labs Inc., ki je bilo prej znano pod imenom Ansca Mobile. Podpira razvoj za iPhone, iPad, Kindle Fire, Nook in Android naprave, primeren pa je predvsem za razvoj grafično zahtevnih aplikacij oziroma iger. OpenGL-ES pogon za prikazovanje nam namreč omogoča strojno pospeševanje grafike in tako lahko grafično procesorsko enoto (GPU) v popolnosti izrabimo. Razvijalci aplikacije razvijajo s programskim jezikom Lua, ki predstavlja sloj nad C++/OpenGL.

Corona ima t.i. naročniški model, kjer razvijalci plačajo letno naročnino $199 USD za posamezno platformo, ali se odločijo za paket s podporo vsem platformam za $349 USD.

## Primerjava medplatformskih ogrodij

Tabela 6.1: Pregled opisanih ogrodij v tabelarični obliki.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **PhoneGap** | | **Rhodes** | **Appcelerator Titanium** | **Corona** | |
| **Spletna stran** | [phonegap.com](http://phonegap.com/) | | [rhomobile.com](http://rhomobile.com/) | [appcelerator.com](http://appcelerator.com/) | coronalabs.com | |
| **Odprtokodna licenca** | [Apache Public License v2](http://en.wikipedia.org/wiki/Apache_License) | | [MIT](http://en.wikipedia.org/wiki/MIT_License) | [Apache Public License v2](http://en.wikipedia.org/wiki/Apache_License) | ✕ | |
| **Brezplačno?** | ✓ | | ✓ | ✓ | ✕ | |
| **Programski jeziki za razvijanje** | [HTML](http://en.wikipedia.org/wiki/HTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript)and [CSS](http://en.wikipedia.org/wiki/CSS" \o "CSS) | | [HTML](http://en.wikipedia.org/wiki/HTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript),[Ruby](http://en.wikipedia.org/wiki/Ruby_(programming_language)) | [HTML](http://en.wikipedia.org/wiki/HTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript), ([PHP](http://en.wikipedia.org/wiki/PHP), [Ruby](http://en.wikipedia.org/wiki/Ruby_(programming_language)) & [Python](http://en.wikipedia.org/wiki/Python_(programming_language)) namizne aplikacije) | LUA | |
| **Platforme** | |  | | | |
| [**iOS**](http://en.wikipedia.org/wiki/IOS)**podpora** | ✓ | | ✓ | ✓ | ✓ | |
| [**Android**](http://en.wikipedia.org/wiki/Android_(operating_system)) **podpora** | ✓ | | ✓ | ✓ | ✓ | |
| [**Windows Phone**](http://en.wikipedia.org/wiki/Windows_Phone)**podpora** | ✓ | | ✓ | ✕ | ✕ | |
| [**BlackBerry**](http://en.wikipedia.org/wiki/BlackBerry) **podopra** | ✓ | | ✓ | ✕ | ✕ | |
|  | **PhoneGap** | | **Rhodes** | **Appcelerator Titanium** | **Corona** | |
| [**Symbian**](http://en.wikipedia.org/wiki/Symbian) **podpora** | ✓ | | ✓ | ✕ | ✕ | |
| [**Palm WebOS**](http://en.wikipedia.org/wiki/WebOS) **podpora** | ✓ | | ✕ | ✕ | ✕ | |
| [**WAC**](http://en.wikipedia.org/wiki/Wholesale_Application_Community)**podpora** | ✕ | | ✕ | ✕ | ✕ | |
| **Ostalo** | |  | | | |
| [**SQLite**](http://en.wikipedia.org/wiki/SQLite)**podpora** | ✓ vendar ne na [BlackBerry](http://en.wikipedia.org/wiki/BlackBerry" \o "BlackBerry)-u | | ✓ | ✓ | ✓ | |
| **Razvoj v nativnem programskem jeziku platforme** | ✕ | | ✓ z  [Rhodes](http://en.wikipedia.org/wiki/Rhodes_Framework" \o "Rhodes Framework) dodatki | ✓ | ✕ | |

Bolj podroben tabelarični prikaz najdemo tudi na spletnem naslovu wikipedije[[23]](#footnote-23).

# **Aplikacija FERI**

## Načrtovanje in funkcionalnosti

### Funkcionalnosti

Aplikacija je že z osnovnim zaslonom, kjer nam prikazuje različne module razdeljena na jasno ločene in vsebinsko različne enote. Začetni zaslon je tako tudi pripravljen na dodatne module v primeru kasnejšega razvoja in dopolnjevanja aplikacije. Moduli, ki jih podrobneje opisujemo spodaj predstavljajo najbolj pogosto uporabljene oz. zahtevane strani iz spletnega portala FERI.

Kot zanimivo funkcionalnost, ki jih ostale tuje fakultete nimajo, velja omeniti možnost uporabe aplikacije v offline načinu. Tako so vprašanja študenta kot je »Kje že imam naslednja predavanja?« ali »Kdaj ima profesor govorilne ure?« odgovorjena brez potrebe povezovanja.

Aplikacija je na voljo tudi v angleškem jeziku, ki sicer na začetku ni bil planiran. Aplikacija je bila namreč pri prvem poizkusu zavrnjena s strani Apple in njihovega ročnega procesa odobritve vsake aplikacije posebej, saj je bila za privzeti jezik aplikacije izbrana angleščina (slovenščine žal ni ne seznamu možnosti), zaslonske slike, ki jih je potrebno prav tako oddati, pa niso prikazovale angleškega jezika. Tako je bilo potrebno aplikacijo dodatno urediti in dodati možnost izbire jezika in je sedaj na voljo v slovenskem in angleškem jeziku.

* **Oglasna deska** – Predstavlja zadnje objave na oglasni deski, možnost pregleda po kategorijah, katere si lahko označimo tudi kot priljubljene in imamo tako še hitrejši dostop do njih. A tukaj najbolj pomembna funkcionalnost je zagotovo možnost push obveščanja na posamezno kategorijo, kjer si lahko uporabnik določi kategorijo in nato za vsako novo objavo dobi t.i. push obvestilo. Primer: Študent 1. letnika UN RIT se naroči na kategorijo „Redni študij“ > „UN RIT“ > „1. letnik“ in tako dobi obvestila takoj ko so objavljena na oglasni deski tudi na svoj mobilnik. Še bolj pomembno, takoj dobi tudi rezultate izpita na katere že tako nestrpno čaka.

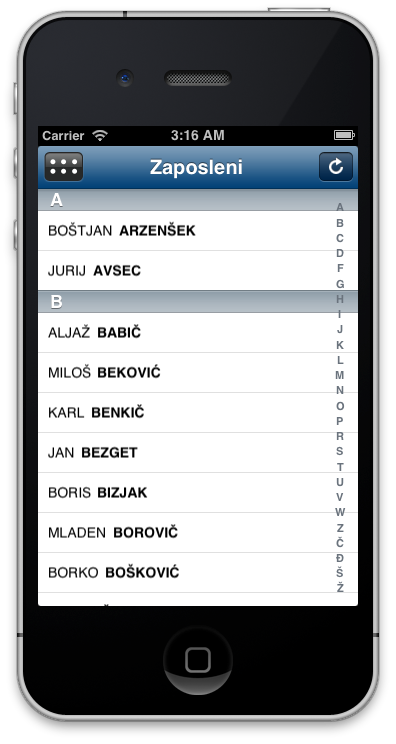
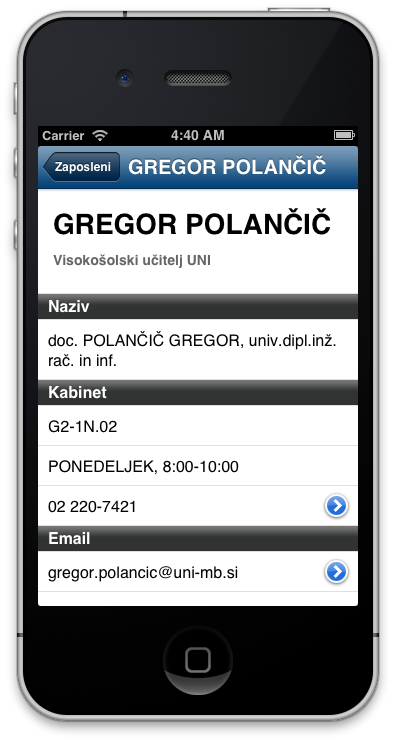


Slika 7.1: Zaslonska slika modula »Oglasna deska«.

* **Urniki** - Kot so objavljeni na FERI spletni strani, a bolj priročni na mobilnem telefonu za hiter pogled v kateri učilnici je naslednje predavanje. Urniki se tudi shranijo v medpomnilnik na mobilniku in tako ni potrebna povezava z internetom vedno ko hočemo na hitro pogledati kdaj se nam začne naslednje predavanje.

  
Slika 7.2: Zaslonska slika modula »Urniki«.

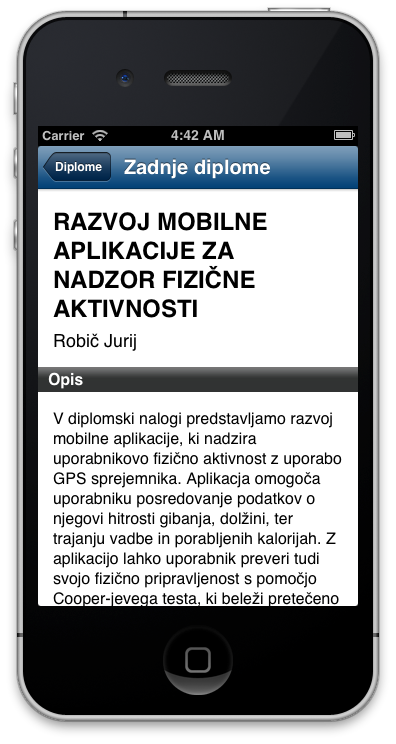
* **Zaposleni** – Študentje velikokrat iščemo kdaj ima določen profesor govorilne ure in v katerem kabinetu ga lahko najdemo. Modul zaposleni nam omogoča hiter dostop do informacij o govorilnih urah, kabinetu, s klikom na telefonsko ali e-mail pa lahko tudi hitro vzpostavimo kontakt s profesorjem.

  
Slika 7.3: Zaslonska slika modula »Zaposleni«.

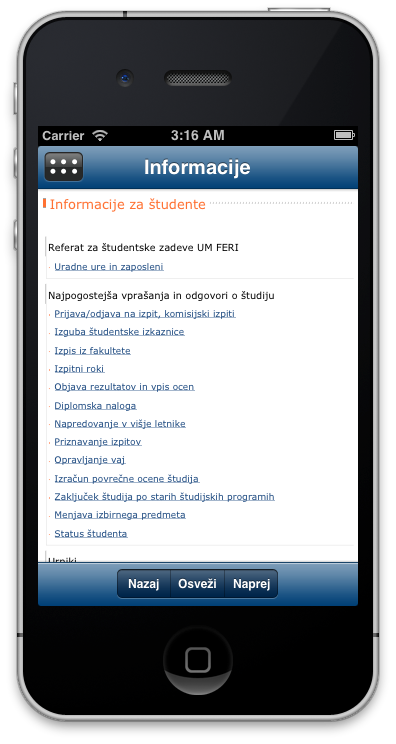
* **Zemljevid** - Grafično predstavljen objekt G2, z tlorisom po vseh nadstropjih in medetaži.

  
Slika 7.4: Zaslonska slika modula »Zemljevid«.

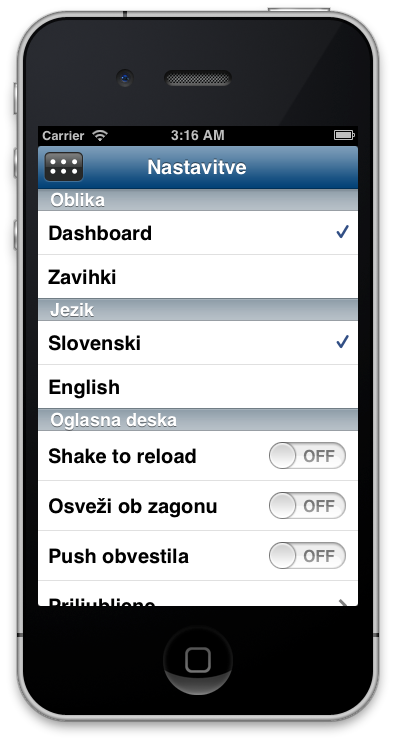
* **Diplome** - Pogledamo lahko, kdaj se odvijajo aktualni zagovori diplom, njihove teme in kandidate. Že zagovorjene zadnje diplome se nahajajo pod razdelkom „Zadnje diplome“ kjer imamo tudi kratek opis diplomskega dela in povezavo na DKUM, kjer si lahko celotno delo tudi pogledamo. Kot zadnje v tem modulu pa imamo še povezavo na celotno zbirko diplom s strani Digitalne Knjižnice Univerze Maribor DKUM.

  
Slika 7.5: Zaslonska slika modula »Diplome«.

* **Informacije –** Prikaže študentom dobro znano stran, do katere dostopamo na spletnem portalu preko klika v levem meniju na povezavo „Informacije za študente“. Stran in njene povezave pa se odprejo v mobilniku bolj prijaznem načinu, saj s pomočjo CSS-a izklopimo nepotreben levi meni in glavo spletne strani z informacijami.

  
Slika 7.6: Zaslonska slika modula »Informacije«.

* **Nastavitve** - Uporabnik si lahko tudi malce prilagodi mobilno aplikacijo. Na iOS platformi lahko tako izbiramo med oblikama „Dashboard“ in „Zavihki“, medtem ko Android verzija podpira le „Dashboard“ pogled. Aplikacija je na voljo tudi v angleškem jeziku in tukaj lahko spremenimo jezik vmesnika. Sledi še nekaj nastavitev glede oglasne deske, kjer vklopimo/izklopimo push obvestila ter priročneje urejamo Priljubljene in Push kategorije.

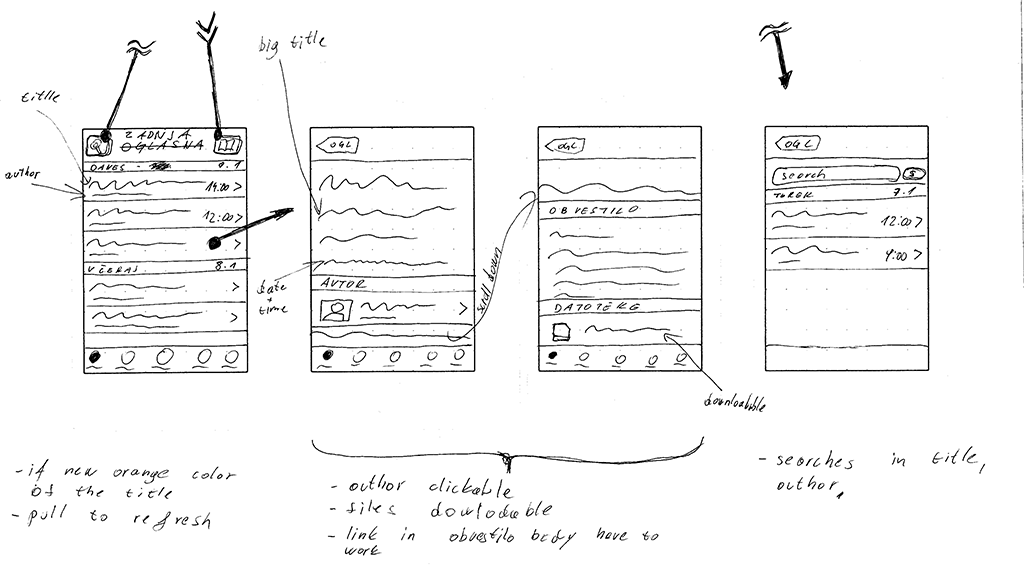
  
Slika 7.7: Zaslonska slika modula »Nastavitve«.

* **FERI –** Prikaz osnovnih podatkov Fakultete za elektrotehniko, računalništvo in informatiko kot so naslov, telefon, telefaks, e-mail naslov itd. Prikažemo tudi kratek opis FERI mobilne aplikacije in seznam ljudi, ki so kakorkoli sodelovali pri nastanku ali poznejšem razvoju. Opozorimo tudi, da gre za odprtokoden projekt pri katerem vabimo študente k nadaljnjem razvoju in izboljšanju aplikacije.

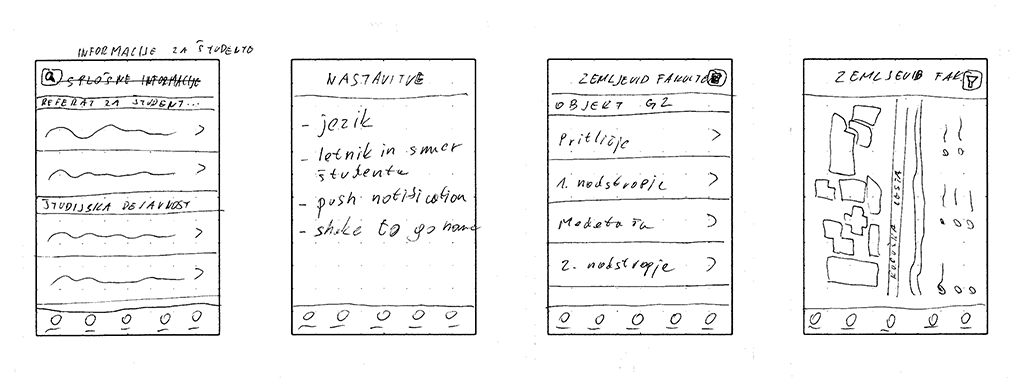
  
Slika 7.8: Zaslonska slika modula »FERI«.

### Wireframes

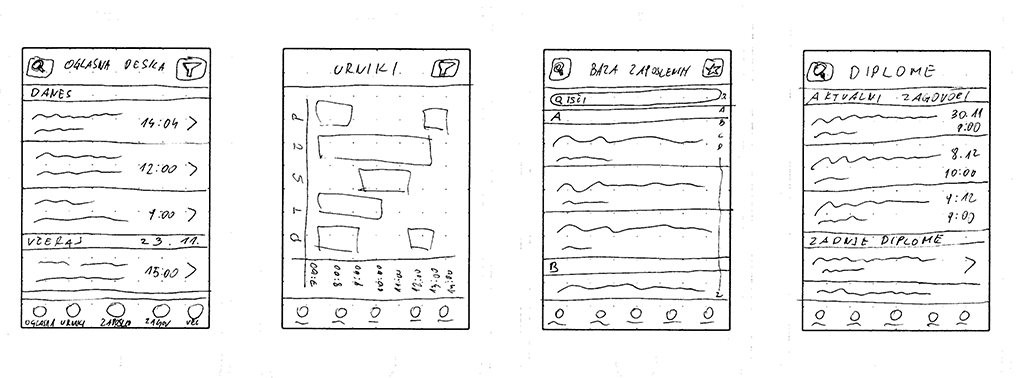
Po tem, ko smo analizirali in ocenili katera področja FERI portala so najbolj obiskana je bilo potrebno izdelati wireframe-e oz. žičnate modele. Ti so bili izdelani kar ročno, saj natančnost in možnost skupne rabe, ki ga ponujajo moderna orodja, ni bilo treba. S pomočjo wireframov je bila aplikacije predstavljena tudi nekaterim poznejšim beta uporabnikov in tako smo dobili kvaliteten feedback in smo lahko že v fazi načrtovanja določene zadeve popravili in prilagodili. Prav tako je v tej fazi padlo ogromno novih idej in možnosti glede funkcionalnosti aplikacije. Vseh v okviru diplomskega dela žal ni bilo možno izdelati, bomo pa jih predstavili v zaključku kot potencialne ideje za nadaljnji razvoj. Določene podrobnosti se sicer iz začetnih wireframov in v končni aplikaciji razlikujejo, saj smo pozneje pri sami uporabi ugotovili določene pomanjkljivosti oz. možnosti za izboljšanje vmesnika.



Slika 7.9: Scan ročnih wireframe-ov 1.



Slika 7.10: Scan ročnih wireframe-ov 2.



Slika 7.11: Scan ročnih wireframe-ov 3.

Kar mogoče manjka pri ročno izrisanih prototipih je možnost interakcije. Tako mogoče ni najbolj jasno kam točno nas klik na gumb pripelje. Zato smo naknadno uporabili mobilno aplikacijo POP (Prototyping on Paper)[[24]](#footnote-24), kjer lahko prototip narišemo ali v aplikacijo ali pa poslikamo že obstoječe skice. Slednje smo tudi naredili in povezali gumbe, tako da lažje prikažemo potem aplikacije. Spletna predstavitev se nahaja na spletnem naslovu strani aplikacije[[25]](#footnote-25). Tako lahko tudi laičnemu uporabniku oziroma ne-strokovnjaku za uporabniške vmesnike že vnaprej prikažemo kako izgleda potek skozi aplikacijo.

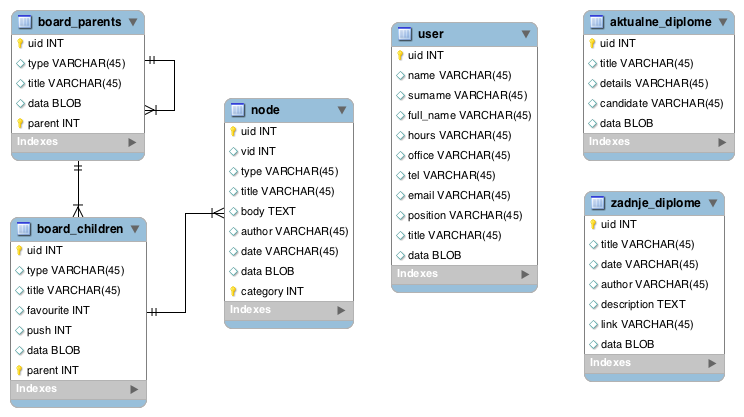
### Userflow

Zelo koristno si je tudi ogledati celotno zgodbo in vse možne poti uporabnika skozi zaslone aplikacije. V ta namen smo izdelali enostaven diagram, kjer lahko na hitro odkrijemo skozi katera okna, je uporabnik pride do končnega pogleda. Tako lahko na Zelo koristno si je tudi ogledati celotno zgodbo in vse možne poti uporabnika skozi zaslone aplikacije. V ta namen smo izdelali enostaven diagram, kjer lahko na hitro odkrijemo skozi katera okna je uporabnik pride do končnega pogleda.

****Slika 7.12: Drevesni pogled vseh poti skozi aplikacijo.

### Podatkovni model

Ker je aplikacije narejena, da omogoča brskanje po podatki o oglasni deski, zaposlenih in diplomah tudi brez internetne povezave, smo morali poskrbeti tudi za lokalno shranjevanje na mobilniku. V ta namen je bila uporabljena podpora podatkovni bazi SQLite, ki jo Titanium podpira. V spodnji sliki si lahko ogledamo podatkovni model baze, ki je sicer zelo enostaven, a zadosti potrebam aplikacije.



Slika 7.13: Diagram podatkovnega modela.

Podatki o kategorijah oglasne deske so že »zapečeni« v samo aplikacijo, ker FERI portal ne ponuja ustreznega servisa od kjer bi podatke o kategorijah in njeni strukturi bilo možno dobiti. To ne predstavlja večjega problema, saj se kategorije ne spreminjajo oziroma če že, se spreminjajo zelo redko, tako da lahko podatke posodobimo tudi z izdajo nove verzije aplikacije.

Podatki o zaposlenih se črpajo iz spletnega servisa[[26]](#footnote-26)ko uporabnik uporabi gumb »osveži« v aplikaciji. Pridobijo se vsi podatki in se ustrezno posodobijo v podatkovni bazi. Tako imamo podatke pozneje na voljo tudi če nimamo dostopa do interneta, kar je prednost pred podobnimi aplikacijami tujih univerz, kar bomo opisali tudi v naslednjem poglavju.

Enako velja za aktualne in zadnje diplome. Podatke pridobimo iz dveh spletnih naslovov[[27]](#footnote-27) [[28]](#footnote-28) in so nato na voljo brez povezave. Gre za manjšo količino podatkov, zato osveževanje traja precej manj časa, kot v primeru baze zaposlenih.

Zadnje objave na oglasni deski pridobivamo preko RSS feeda[[29]](#footnote-29). Ne posodobimo le seznama naslovov in osnovnih podatkov, ampak tudi vsebino obvestila. Obvestila so praviloma zelo kratka, tako da tudi tukaj pridobivanje svežih podatkov preko interneta ne predstavlja težav, saj gre le za goli tekst, morebitne datoteke pa v obvestilo niso vključene.

Vredno je omeniti še način, kako se začetna struktura tabel ustvari. Tukaj ne uporabljamo klasičnih SQL stavkov za kreiranje tabele, temveč smo uporabili delček kode iz odprtokodnega projekta Codestrong [vir], kjer entiteto preprosto opišemo z javascript array-em in se tabelo preko tega kreira oz posodobi v primeru že obstoječe tabele.

fields : function() {

return {

fields : {

uid : {

type : 'INTEGER'

},

title : {

type : 'VARCHAR'

},

details : {

type : 'VARCHAR'

},

candidate : {

type : 'VARCHAR'

}

},

indexes : {

uid\_idx : ['uid']

}

};

},

Slika 7.14: Opis entitete modela zadnje\_diplome.

if (!Database.db.getConnection('main').tableExists('aktualne\_diplome')) {

Database.entity.db('main', 'aktualne\_diplome').initializeSchema();

}

Slika 7.15: Preverjanje ali entiteta že obstaja.

Možnost shranjevanja podatkov za t.i. »offline« način smatramo kot prednost pred podobnimi aplikacijami. Študentje, ki so naša cilja skupina, namreč nimajo vedno na voljo neomejenih količin zakupljenih podatkov s strani mobilnih operaterjev, pa tudi sprejem mobilnega signala je na nekaterih delih univerze malenkost slabši in tako oteženo pridobivanje podatkov s spletnih servisov.

## Analiza aplikacij drugih fakultet

Mobilna aplikacija FERI je prva aplikacija med slovenski univerzami oziroma fakultetami. Medtem ko se pri nas tovrstne aplikacije še niso razvile, pa imajo večje ameriške in angleške univerze svoje aplikacije in jih uspešno uporabljajo za komunikacijo s študenti. Aplikacije imajo številčne funkcionalnosti. Tako imamo skoraj povsod oglasno desko oziroma novice, seznam zaposlenih in karto kampusa. To sta nekako klasični deli aplikacije in se med seboj ne razlikujejo prav veliko. Podobno smo naredili tudi pri FERI aplikaciji, vendar uporabili dodatno možnost pogleda (zadnje objave ali kategorizirane objave) ter push funkcionalnost, kar pri drugi univerzah ni bilo opaziti. Nekaj aplikacij tujih univerz smo tudi testirali in zapisali točke, ki so najbolj izstopale pri posamezni aplikaciji in bi lahko pomagale pri razvoju lastne aplikacije.

****

Slika 7.16: Testirane aplikacije.

### Cambridge University

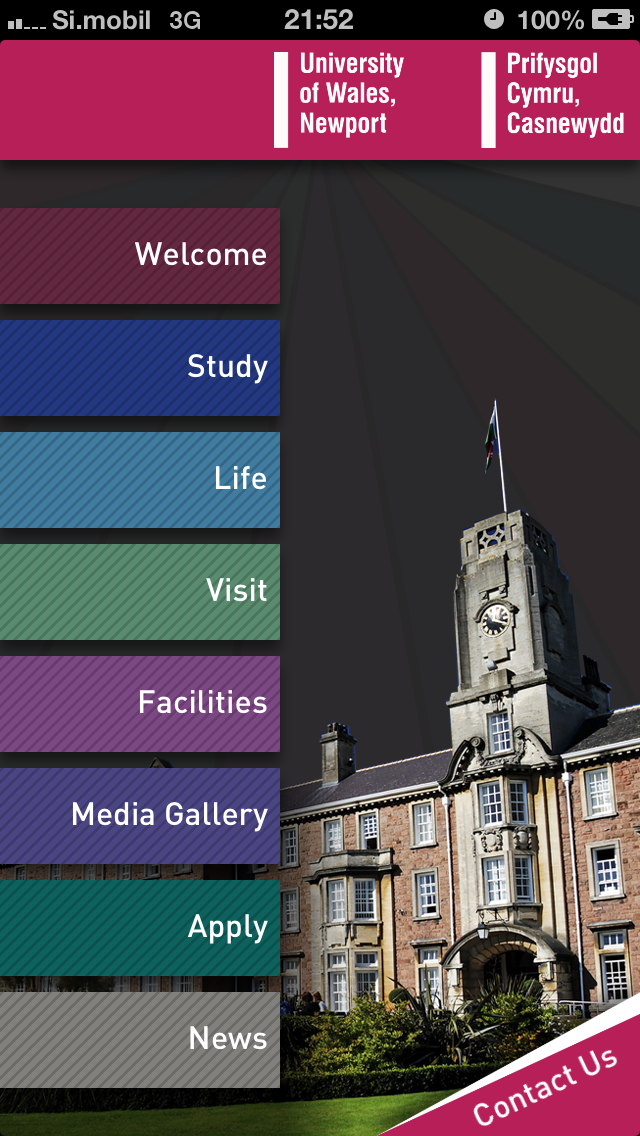
* Dogodki predstavljeni v koledarskem pogledu, ki prikazujejo dogajanje na univerzi.
* Media modul, kjer najdemo nekaj posnetkov iz univerze, ter njihov uradni Youtube kanal.
* Knjižnica z možnostjo iskanju po gradivih.
* Možnost deljenja novic in drugih informacije preko socialnih omrežij Twitter in Facebook.

### Harvard University

* Sledenje trenutnim javnim prevozom, ki peljejo na kampus.
* Informacije o prehrani in trenutni jedilniki.
* Modul, kjer so prikazane vse novice s področja športnih dogodkov.
* Aktivnost uradnih računov iz socialnih omrežij Youtube, Twitter in Facebook.
* Nujna obvestila kampusa.

### University of Wales, Newport

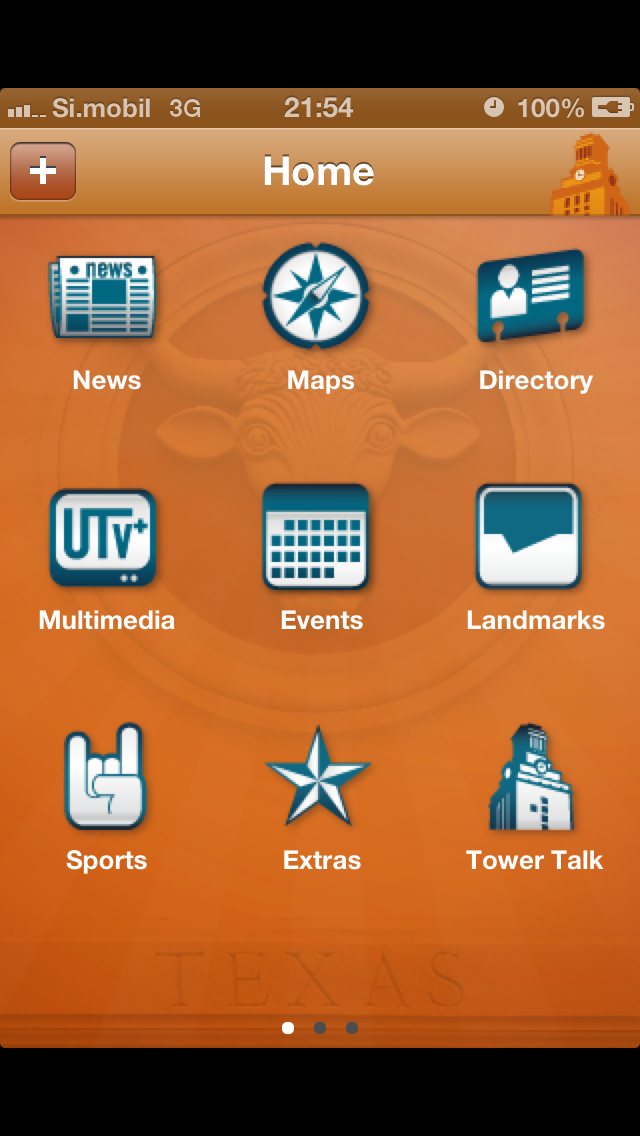
* Zelo dovršena aplikacij s strani grafičnega dizajna. Uporablja sicer svojo logiko navigacije med zasloni, ki ni v sladu s platformo, a je zelo dobro izpeljana.



Slika 7.17: University of Wales, Newport.

### University of Texas at Austin

* Študijski koledar, ki prikaže vse pomembne datume.
* Podatki o zaključnih izpitih.
* V skoraj vse module se je potrebno prijaviti kot njihov študent in tako ni bilo možno bolj podrobno analizirati delovanje.



Slika 7.18: University of Texas at Austin.

### University of Nebrska

* Uporablja tehnologijo podjetja Straxis.
* Ankete za pridobivanje anonimnih podatkov na različne teme, ki se tičejo življenja na kampusu.
* Poslušanje študentskega radia v živo.
* Himna univerze, ki jo lahko predvajamo na telefonu.

****

Slika 7.19: University of Nebraska.

Testirali smo še nekaj ostalih aplikacij, a zaradi omejitve prikaza informacij, ali razloga ker, niso izstopale iz povprečja, jih nismo tukaj posebej obravnavali. Pri aplikacijah opazimo, da pri marsikateri sama univerza ni avtor, temveč je razvoj izvajalo zunanje podjetje. Tako je recimo zadnja izmed opisanih aplikacij, University of Nebraska, izdelana s pomočjo ogrodja, ki ga ja razvilo podjetje Straxis. Gre za že vnaprej izdelane šablone in module, ki jih nato prilagajajo vsaki univerzi posebej.

Funkcionalnost, ki jo pri večini aplikacijah pogrešamo je možnost pregled informacij v načinu brez internetne povezave. Že mogoče res, da imajo študentje vedno povezavo z internetom preko mobilnega operaterja ali preko brezžičnega omrežja na področju kampusu, a za na primer pregled urnika ali karte, se lahko povezavi na internet popolnoma izognemo s pomočjo lokalnega medpomnjenja na mobilniku. Tukaj je recimo za to funkcionalnost v FERI aplikaciji poskrbljeno, saj se skoraj vsa vsebina posameznih modulov shrani na samem mobilniku.

## Grafični vmesnik aplikacije

Pomemben del pri vsaki aplikaciji predstavlja tudi njen grafični videz. Za grafiko je pri FERI aplikaciji poskrbel kolega Matej Grušovnik, grafični dizajner z večletnimi izkušnjami na področju web dizajna, 3D modeliranja ter izdelave logotipov. V spodnjih slikah si lahko ogledamo, kako je izgledala aplikacija pred in po grafični prenovi. Pri vključevanju novih grafik smo ugotovili tudi določene pomanjkljivosti v sami zasnovi vmesnika in tako prenovili določene zaslone in gumbe, ter tako poskrbeli, da je aplikacija dobila bolj domač vmesnik tako za iOS kot tudi za Android. Mobilne aplikacije namreč sledijo določenim navodilom in načelom (**Human Interface Guidelines link**), ki so skupne večini aplikacij in tako poskrbijo, da se uporabnik počuti domače že pri prvi uporabi aplikacije.

Grafični dizajn pa se seveda začne pri ikoni, ki ni le predstavlja vhod v aplikacijo, ampak je grafika s katero se prikazujemo tudi drugod na spletu in mora biti dovolj enostavna in razpoznavna, kar je v primeru FERI aplikacije tudi uspelo:



Slika 7.20: Logotip v različnih velikostih.



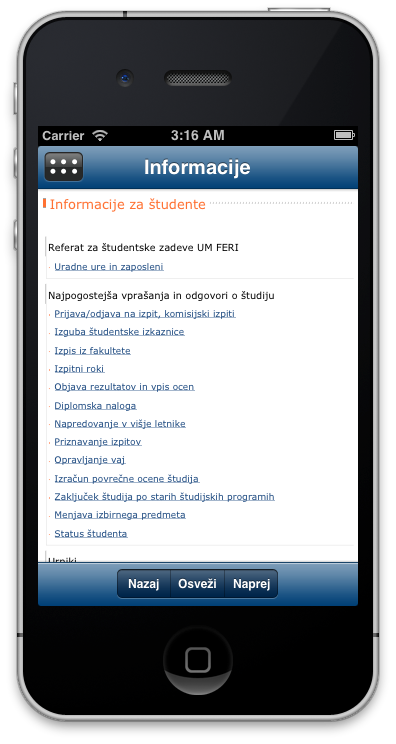
Slika 7.21: Primerjava starega in novega »začetnega zaslona«.



Slika 7.22: Primerjava starega in novega »dashboarda«.



Slika 7.23: Primerjava stare in nove »oglasne deske«.



Slika 7.24: Primerjava starih in novih »informacij«.

## Opis uporabljenih tehnologij

### Appcelerator Titanium

Appcelerator Titanim smo malce že opisali v prejšnjem poglavju, sedaj pa bomo dodali nekaj podrobnosti in podali zakaj smo se za razvoj odločili ravno z tem ogrodjem. Titanium podpira le platformi Android in iOS, vendar glede na uporabo mobilnikov na našem trgu in oceno uporabe pri študentih, sklepamo da je večina uporabnikov pametnih mobilnih naprav ravno na teh dveh platformah. Zelo pomemben faktor pri odločitvi je bilo tudi način kako Titanium genrira končne aplikacije. Ostala ogrodja namreč uporabljajo CSS in HTML za izgradnjo vmesnika, čeprav razvijalec tega ne opazi saj izvorno kodo piše prav tako na primer v javascriptu. Posledično je vmesnik lahko tudi malce drugačen in ne daje pravega občutka pri navigaciji in odzivnosti pri sami navigaciji. Medtem pa Titanium uporablja izvorne funkcionalnosti naprave in tako generira vmesnik z uporabo nativnih metod in knjižnic platforme. V ozadju sicer še vedno poteka del transformacije iz javascripta v nativno kodo (Objective C v primeru iOS in Java v primeru Androida) vendar je končen rezultat kljub temu zelo hitra in odzivna aplikacija, kar je pomemben faktor za zadovoljstvo končnega uporabnika. Imamo tudi zelo koristno aplikacijo[[30]](#footnote-30) z pripadajočo izvorno kodo, ki nam predstavi skoraj vsako možno uporabo različnih oken, pogledov, gumbov itd., kjer smo našli skoraj vse možnosti, ki jih bomo potrebovali pri FERI aplikaciji. Tako smo se lahko v naprej prepričali, da bomo lahko vse načrtovane funkcionalnosti tudi implementirali. Titanium ima tudi nekaj pomanjkljivosti, ki so se najbolj izkazali med samim razvojem, kot je na primer različno delovanje enakih metod, če prav bi le te morale dajati enake rezultate. Določene težave so se sicer odpravile z novo verzijo 3.0, vendar ker je začetni razvoj potekal še z verzijo 2.x je marsikje v izvorni kodi ostalo pisaje majhnih delčkov kode za obe platformi ločeno. Več o teh težavah in ali je Titanium primeren za prav vsako aplikacijo si bomo lahko prebrali še v zaključnem poglavju.

### Javascript

JavaScript je objektni skriptni programski jezik, ki ga je razvil Netscape, da bi spletnim programerjem pomagal pri ustvarjanju interaktivnih spletnih strani. Jezik je bil razvit neodvisno od Jave, vendar si z njo deli številne lastnosti in strukture. JavaScript podpirajo velika programska podjetja in kot odprt jezik ga lahko uporablja vsakdo, ne da bi pri tem potreboval licenco. Sintaksa jezika JavaScript ohlapno sledi programskemu jeziku C. Prav tako kot C,

Kar 95 % izvorne kode mobilne aplikacije FERI je zapisane s pomočjo javascripta. S pomočjo javascripta tako kličemo metode v API-ju Titaniumu. Javascript je relativno enostaven za uporabo in ne potrebuje veliko učenja in prilagajanja, da bi razvijalec lahko začel z delom. Kot vsak drug jezik je tudi tukaj zelo pomembno kakovost in struktura kode. Gre za jezik, ki dovoljuje številne napake oziroma ne najbolj optimalno napisno izvorno kodo, tako da lahko hitro zabredemo v težave, če nimamo pravilne strukture in pogleda kako bo razvoj aplikacije potekal naprej. Izvorna koda FERI aplikacije je tako ločena na različne datoteke, ki predstavljajo vsaka svoj modul oz. okno tudi znotraj modula. Imamo še datoteke s knjižnicami in funkcijami, ter ločeno izvorno kodo za podatkovno bazo, entitete, push obvestila in izgradnjo začetnega dashboarda. Izvorna koda je tako relativno lepo razdeljena v logične enote in omogoča nemoten nadaljnji razvoj novih modulov.

### HTML / CSS

Hyper Text Markup Language (slovensko jezik za označevanje nadbesedila, kratica HTML) je označevalni jezik za izdelavo spletnih strani. Predstavlja osnovo spletnega dokumenta. S pomočjo HTML razen prikaza dokumenta v brskalniku hkrati določimo tudi strukturo in semantični pomen delov dokumenta. Cascading Style Sheets (sl. kaskadne stilske podloge) poznane pod kratico CSS so podloge, predstavljene v obliki preprostega slogovnega jezika, ki skrbi za predstavitev spletnih strani. Z njimi definiramo stil HTML oz. XHTML elementov v smislu pravil, kako se naj ti prikažejo na strani. Določamo lahko barve, velikosti, odmike, poravnave, obrobe, pozicije in vrsto drugih atributov, prav tako pa lahko nadziramo aktivnosti, ki jih uporabnik nad elementi strani izvaja (npr. prekritje povezave z miško). Podloge so bile razvite z namenom konsistentnega načina podajanja informacij o stilu spletnim dokumentom.[[31]](#footnote-31) [[32]](#footnote-32)

Obeh tehnologij pri izdelavi mobilne aplikacije pretirano nismo uporabljali. Uporabo lahko opazimo recimo pri modulu Informacije, kjer s pomočjo CSS-a skrijemo nepotrebne dele spletnega portala, saj nas pri prikazu na mobilniku le motijo. Tako na primer ni viden levi meni in glava spletnega portala. Enako velja za prikaz urnikov, kjer izključimo zgornji okvir z izbiro letnika in študija, saj je ta izbira že integrirana v samo aplikacijo. HTML se uporablja še pri prikazu modula Zemljevid in prikazi splošnih informacij o fakulteti in aplikaciji.

### Podatkovna baza SQLite

SQLite je relacijska podatkovna baza, velikosti le približno 350 KB. V nasprotju z ostalimi podatkovnimi bazami, SQLite ni ločen proces, ampak je integriran že v samo aplikacijo. Podatki se zapisujejo neposredno v navadno datoteko na sistemu. Ta datoteka je potem kompatibilna med vsemi operacijskimi sistemi kot tudi med 32 in 64 bitno arhitekturo. Deluje v skladu z ACID (Atomicity, Consistency, Isolation, Durability) in implementira večina SQL standarda, tako da je razvijalec z izkušnjami iz ostalih relacijskih baz že seznanjen z načinom povezovanja in pridobivanja podatkov. Najbolj je popularen v lokalnih aplikacijah kot je na primer webbrowser. Nekateri zatrjujejo, da gre celo za najbolj uporabljeno podatkovno bazo, saj ga uporablja veliko sodobnih brskalnikov in operacijskih sistemov.

### PHP za strežniški del

Za strežniški del, ki je zadolžen za klicanja spletnega servisa za pošiljanje push obvestil, bi lahko pravzaprav uporabili kateri koli programski jezik. Izbrali smo PHP, razširjen odprtokodni programski jezik, ki se uporablja večinoma za razvoj dinamičnih spletnih strani. Pri projektu FERI aplikacije smo potrebovali le enostavno skripto, ki vsako minuto preveri ali imamo kakšen nov zapis v oglasni deski na FERI portalu. V primeru, da obstaja novo obvestilo, si le to zabeležimo kot že prebrano in pokličemo spletni servis na strani Appcelerator Titanium-a, ki na to poskrbi, da obvesti uporabnike, naročene na posamezno kategorijo.

## Razlike v kodiranju za iOS in Android

Čeprav nam medplatformska ogrodja obljubljajo enako delovanje na vseh platformah pa se v praksi izkaže, da ni vedno tako. Določeni klici metod namreč delujejo nekoliko drugače, kar je večinoma v dokumentaciji sicer zabeleženo, a kljub temu najdemo primere kjer je delovanje drugačno od pričakovanega. Na splošno velja, da je treba pri samem razvoju večkrat preveriti delovanje na vseh platformah in tudi na različnih napravah. Tako se dovolj zgodaj izognemo nepredvidenemu delovanju in težavnemu odkrivanju in odpravljanju napak.

Da poskrbimo za različno delovanje glede na platformo v izvorni kodi uporabljamo preproste kontrolne stavke. Funkcija, ki jo večkrat uporabimo in nam vrne ali gre za Android platformo izgleda tako:

feri.isAndroid = function() {

if (feri.\_\_isAndroid === undefined) {

feri.\_\_isAndroid = (Ti.Platform.osname == 'android');

}

return feri.\_\_isAndroid;

}

Slika 7.25: Preverjanje trenutne platforme.

Poglejmo še primer uporabe te funkcije, ki poskrbi za dodaten gumb v meniju na Android platformi. Tega gumba na iOS recimo ne potrebujemo, saj se gumb za nazaj vedno nahaja v zgornjem levem kotu vmesnika.

*// android back button listener and refresh menu*

if (feri.isAndroid()) {

diplomeWindow.addEventListener('android:back', function() {

feri.navGroup.close(feri.iconWin, {

animated : true

});

*// re-enabling the icons on the dashboard*

feri.dashboardActive = true;

});

var activity = diplomeWindow.activity;

activity.onCreateOptionsMenu = function(e) {

var menu = e.menu;

var menuItemRefresh = menu.add({

title : lang['osvezi']

});

menuItemRefresh.addEventListener("click", function(e) {

Ti.App.fireEvent('feri:update\_data\_diplome');

});

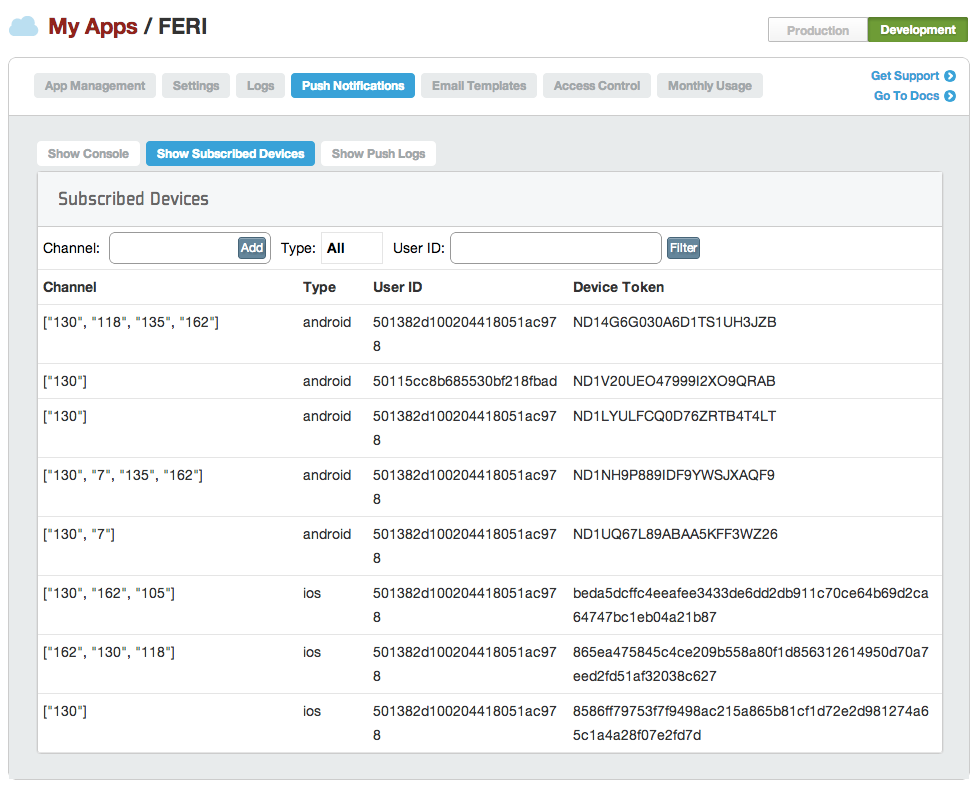
};

}

Slika 7.26: Ločevanje logike za različne platforme.

## Strežniški del razvoja in izmenjava podatkov

Za delovanje push sistema je bilo potrebno razviti preverjanje novih obvestil na oglasni deski FERI portala in obveščanje uporabnikov, ki so naročeni na določene kategorije. V ta namen je bila razvita dokaj enostavna PHP skripta, ki v določenem časovnem intervalu prebere RSS vir in ugotovi ali imamo nova obvestila. Obvestila so kategorizirana in tako lahko glede na podano kategorijo obvestimo mobilne uporabnike, da obstaja novo obvestilo v kategoriji na katero so naročeni. Obveščanje poteka preko Appcelerator cloud storitve, ki je na voljo brezplačno za do 5.000.000 obvestil, kar pa bi za potrebe FERI portala in obveščanje študentov moralo zadostovati. Skripta tako v primeru novega obvestila izvede zahtevo na storitev in poda identifikator kategorije ter tekst obvestila. Pred tem se še izvede avtentikacija preko 32 mestnega ključa, ki prepreči morebitno zlorabo in nekontrolirano pošiljanje obvestil uporabnikom.



Slika 7.27: Seznam naročenih naprav v testnem okolju.

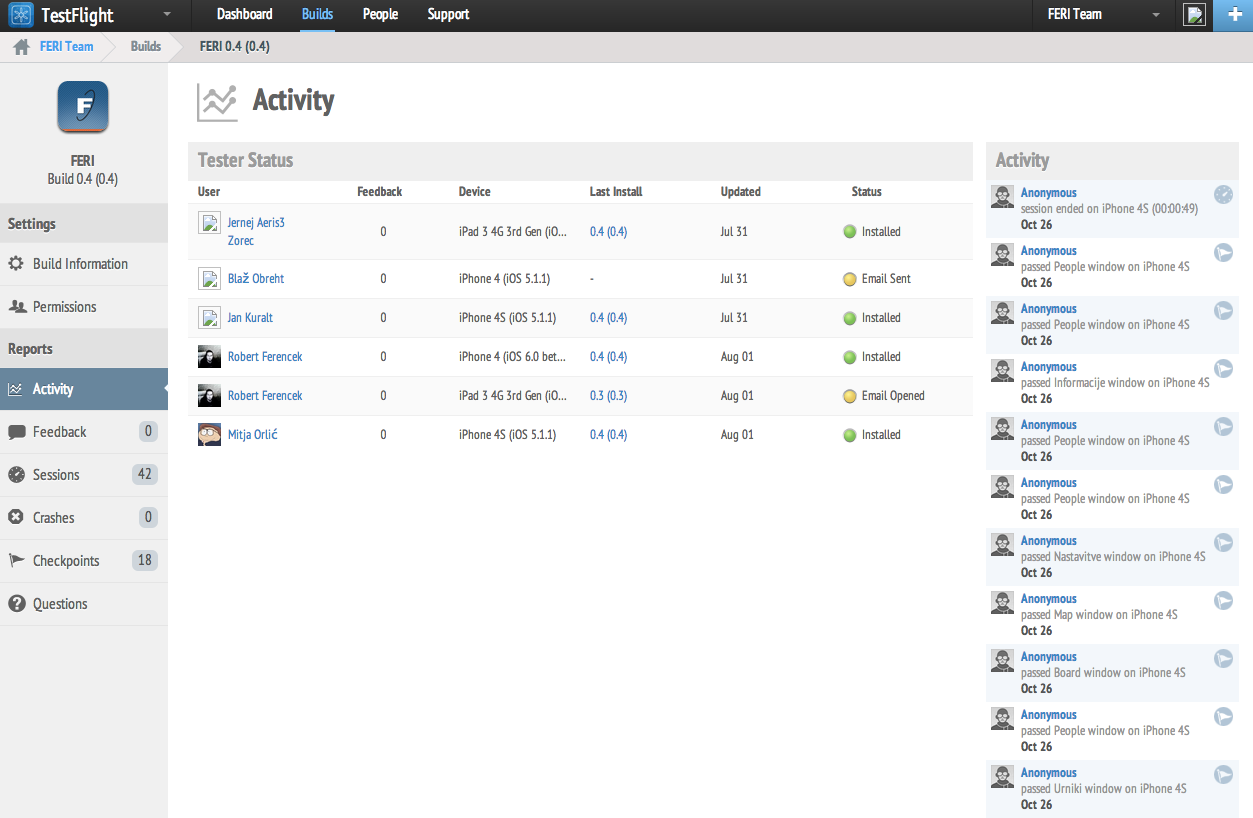
Tukaj bi lahko skripto oziroma zahtevo za obvestilo implementirali tudi na strežniku, kjer se obvestila dejansko vpisujejo in bi tako zagotovili obvestilo brez zakasnitve. Trenutno se skripta namreč izvaja na ločenem strežniku v časovnem intervalu ene minute.

## Beta testiranje

Ko je bila mobilna aplikacija razvita do te mere, da je bila pripravljena na beta testiranje uporabnikov smo poiskali nekaj uporabnikov, ki so nato podali mnenje o strukturi podatkov in modulov, vmesniku, izkušnji med uporabo in pomanjkljivostih.

Najprej smo opazovali nekaj uporabnikov, ki so sicer vešči uporabe pametnih mobilnih telefonov. Tukaj je šlo večinoma za študente FERI-ja tako, da so že na prvi pogled začetnega zaslona vedeli, katere sklope informacij aplikacija ponuja. Brez navodil so začeli uporabo aplikacije medtem, ko smo opazovali njihove interakcije z vmesnikom. Po nekaj testnih uporabnikih se je hitro pokazal vzorec, kako uporabniki na primer zgrešijo določene gumbe za nastavitev urnikov ali prilagojen prikaz oglasne deske. Na podlagi teh ugotovitev je bilo potem izvedenih nekaj popravkov samega vmesnika, da bi zagotovili čim bolj kvalitetno izkušnjo nadaljnjim uporabnikom.

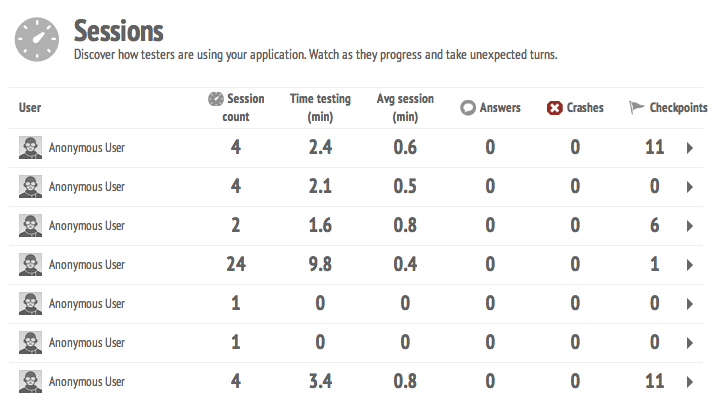
Beta uporabniki so aplikacijo dobili nameščeno tudi predčasno in tako smo lahko preko orodja Test Flight[[33]](#footnote-33) zbrali nekaj podatkov o časovni uporabi posameznih modul in morebitnih napakah, ki bi pomenile sesutje aplikacije. Aktivnost je s pomočjo tega orodja mogoče spremljati tudi v živo, vendar pri manjšem številu uporabnikov ne pride toliko do izraza. Tukaj smo zbrali predvsem podatke o tem, kateri modul je največ v uporabi s uporabo kontrolnih točk. Preverili smo lahko tudi ali se aplikacija pravilno namesti na različnih napravah.



Slika 7.28: Dashboard v spletni aplikaciji TestFlight.



Slika 7.29: Activity okvir v spletni aplikaciji TestFlight.



Slika 7.30: Podatki o trajanje seje v spletni aplikaciji TestFlight.

## Analiza uporabe in statistika

**TODO Koliko download je bilo po objavi aplikacije, analiza iphone vs android, naprave ki imajo nameščeno aplkacijo,...** … Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse aliquam feugiat lectus a lacinia. In pharetra feugiat hendrerit. Suspendisse tincidunt aliquet lorem, eu iaculis purus commodo ac. Aliquam sit amet ante ante, ac placerat nibh. Praesent erat massa, mattis pulvinar iaculis nec, commodo in lacus. Morbi rutrum gravida quam, rutrum interdum felis interdum ac. Quisque ut urna nulla, a pellentesque risus. Donec at lorem vitae ante ornare egestas id in leo. Vestibulum euismod dignissim leo, tempus lobortis massa semper eu. Maecenas venenatis sagittis vestibulum. Morbi feugiat eros a massa feugiat congue. Integer quam nunc, convallis vitae ornare ut, adipiscing et justo. Curabitur mollis feugiat turpis a tristique.

## Nadaljnji razvoj in dostopnost materialov

Izvorna koda aplikacije je izdana pod odprto kodno licenco, celoten projekt pa je na voljo na Githubu[[34]](#footnote-34). Zelo spodbujamo njen razvoj tudi s strani drugih razvijalcev in študentov, tako da je vsak, ki bi si želel obstoječo aplikacijo nadgraditi, dodati novo funkcionalnost ali samo odpraviti kakšen bug[[35]](#footnote-35), več kot dobrodošel. Izvorna koda ni kompleksna, načeloma napisana berljivo in tudi dokumentirana. Tako se lahko študentje, ki imajo vsaj osnovno znanje iz spletnih tehnologij in željo se naučiti razvoja z Appcelerator Titanium ogrodjem, hitro priključijo k razvoju.

Seznam materialov:

* Spletna stran z informacijami:
* Izvorna koda aplikacije:
* Izvorna koda strežniškega dela:
* Grafične datoteke:

Pozivamo tudi uporabnike aplikacije, da v primeru odkritja kakšne težave ali hrošča to sporočijo na avtorjev e-mail naslov [mail] ali, še bolje, vnesejo kar nov task v seznam pri projektu na Githubu.

Vabljeni torej k so-ustvarjanju mobilne aplikacije FERI!

# Sklepne ugotovitve

Mobilno aplikacijo FERI smo torej pripeljali do stabilne verzije in jo objavili na obeh najpopularnejših marketih. Študentje jo uporabljajo in imajo tako hitrejši in prijaznejši dostop do pogostih informacij spletnega portala FERI.

Pri razvoju smo se odločili za medplatformski način razvoja oziroma bolj podrobneje, razvoj s pomočjo ogrodja Appcelerator Titanium. Med samim procesom pa smo naleteli na nekatere ovire in se tako upravičeno začeli spraševati ali je medplatformski način primeren za razvoj take mobilne aplikacije. Marsikje namreč ogrodje ni delovalo po pričakovanjih in dobivali smo celo drugačne rezultate na različnih platformah. Kasneje so sicer te težave z novo verzijo SDK-ja deloma izginile, saj je očitno šlo za manjše hrošče, na katere pa smo pri našem razvoju naleteli. Pisanje izvorne kode v enem jeziku in takoj videti rezultate na večih platformah je sila dober občutek, saj smo na nek način ubili dve muhi na en mah. A takoj ko preidemo h kompleksijšim aplikacijam se lahko zgodba hitro obrne. Ko dobivamo napake, ki neposredno niso vezane na našo izvorno kodo in ne napišejo v kateri vrstici je napaka, pa je lahko nadaljni razvoj in odprava napak zelo utrujajoča. Ker imamo, v našem primeru, med napisanim javascriptom in končnim ObjectiveC oziroma Javo, dodatno plast pretvorbe izvorne kode, nam orodje seveda ne more podati zelo natančnega poročila o napaki. Žal ne moremo podati konkretnih izkušenj z uporabo drugih medplatformskih orodij, saj diplomsko delo ni pokrivalo praktičnega dela z primerjavo različnih ogrodij.

Če potegnemo črto pod celotno izkušnjo izgradnje FERI aplikacije, bi lahko mnenje ali se odločiti za medplatformski razvoj ali native razvoj za vsako platformo posebej, podali z to ugotovitvijo. Velik del odločitve predstavlja dejstvo za kakšno aplikacijo gre. Če bomo razvijali dokaj preprosto aplikacijo, ki ima klasične module in funkcionalnosti, je medplatformski način razvoja vsekakor primeren. Aplikacije, ki prikazujejo novice, programe prireditev, lokacije različnih storitev itd. so definitivno primerne za razvoj z enim izmed medplatforskimi ogrodji. To so namreč aplikacije, ki ne delajo preveč z grafiko, prikazujejo standardne vsebine in niso posebej požrešne performančno. Ko pa pridemo do malce večjega in zahtevnejšega produkta, pa se seveda zadeve malce obrnejo. Pod zahtevnejšo aplikacijo bomo tukaj uvrstili aplikacije, ki intenzivno delajo z grafiko, potrebujejo lasten vmesnik ali posebnosti v obstoječem vmesniku, in so seveda performančno zelo zahtevne. Z razvojem ločeno za vsako platformo pa seveda lahko dosežemo najbolj optimalno aplikacijo. Seveda je tukaj še več faktorjev, ki vplivajo ali je to pravilna ali napačna odločitev. Na primer izkušnje in sposobnost razvijalcev z delom v native okoljih. Teh je danes sicer vedno več, a vseeno najdemo več kvalitetnih razvijalcev s področja spletnega programiranja (kateri je povečini osnova za medplatformsko razvijanje mobilnih aplikacij) kot pa s področja native razvijalcev. Definitivno lahko tudi slaba koda napisana specifično za eno platformo na koncu dela počasneje kot pa koda, ki se mora še dodatno pretvoriti za nazivno delovanje.

Najbolj pomembna odločitev je torej vrsta aplikacije, ki jo izdelujemo. Igre na primer, razen s pomočjo ogrodja Corona SDK, niso najbolj optimalna aplikacija za razvoj z medplatformkim orodjem. Tukaj so performance namreč zelo pomembne in želimo si kar se da optimizirano aplikacijo. Aplikacijo izdelano v sklopu te diplomske naloge, bi lahko uvrstili ravno nekam v sredino odločitve o načinu razvoja. Pod črto, če imamo relativno enostavno aplikacijo (prikaz podatkov, tabel, webview-i) je smiselno izbrati medplatformska orodja, saj bomo tako hitro prišli do rezultatov na večih platformah hkrati. Če pa izdelujemo aplikacijo, kompleksnejšo in mogoče celo z lastnim grafičnim vmesnikom ali kakšnimi posebnostmi, pa je potrebno izbrati nativni razvoj. Le tako bomo natančno vedeli kaj se dogaja in dosegli maksimalno izrabo zmogljivosti mobilnika.

Vse aplikacije seveda tudi vsebujejo skrite hrošče, ki lahko povzročijo slabše delovanje ali celo ne-delovanje. FERI aplikacija tukaj ni izjema in tudi iz tega razloga je bila izvorna koda in vsi potrebni materiali za razvoj, objavljeni javno pod opensource licenco. Mobilno aplikacijo bomo sicer seveda tudi v naprej vzdrževali, odpravljali napake in dodajali manjše funkcionalnosti. A vendarle vabimo študente, da se razvoju pridružijo in pomagajo mobilno aplikacjo FERI narediti še boljšo.

VIRI

Crook, Jordan; Gartner IDC: *Windows Phone To Steal Second Place From iOS By 2015*, [Online] Dostopno na naslovu <http://techcrunch.com/2011/09/02/gartner-idc-windows-phone-to-steal-second-place-from-ios-by-2015>, 2. September 2011.

Firtman, Maximilian; *Programming the Mobile Web*, O’Reilly Media; 2010.

Elgin, Ben *"Google Buys Android for Its Mobile Arsenal"*, [Online] Dostopno na naslovu <http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>,17. Avgust 2005.

Industry Leaders Announce Open Platform for Mobile Devices (Press release), [Online] Dostopno na naslovu <http://www.openhandsetalliance.com/press_110507.html>, Open Handset Alliance; 5. November 2007.

Chowney, Vikki; *Android takes 52.5% of smartphone market*. Econsultancy, [Online] Dostopno na naslovu <http://econsultancy.com/si/blog/8279-android-doubles-market-share-to-take-52-5-of-smartphone-market>, 15. November 2011.

Google Play for Developers, [Online] Dostopno na naslovu <https://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=112622&topic=15867&ctx=topic>, 5. Januar 2013.

*Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year*, [Online] Dostopno na naslovu <http://www.gartner.com/newsroom/id/1689814>, 19. Maj 2011.

RIM Company – *Learn about Research in Motion*, RIM [Online] Dostopno na naslovu <http://www.rim.com/company/>, 23. September 2012.

»BlackBerry«, [Online] Dostopno na naslovu <http://www.blackberry.com/>, 3. December 2012.

Android and iOS Surge to New Smartphone OS Record in Second Quarter, IDC, [Online] Dostopno na naslovu <http://www.idc.com/getdoc.jsp?containerId=prUS23638712>, 8. Avgust 2012.

Android Expected to Reach Its Peak This Year as Mobile Phone Shipments Slow, BUSINESS WIRE, [Online] Dostopno na naslovu <http://www.sys-con.com/node/2291309>, 6. Junij 2012.

Devilla, Joey *The Windows Phone Predictions that IDC, Gartner and Pyramid Research Probably Hope You’ve Forgotten*, [Online] Dostopno na naslovu <http://www.globalnerdy.com/2012/05/07/the-windows-phone-predictions-that-idc-gartner-and-pyramid-research-probably-hope-youve-forgotten/>, 7. Maj 2012.

»Mobitel«, *Android pri nas uporablja že polovica uporabnikov mobilnega spleta*, [Online] Dostopno na naslovu <http://tehnik.mobitel.si/android-pri-nas-uporablja-ze-polovica-uporabnikov-mobilnega-spleta/>, 8. Marec 2012.

Morgan Stanley, Internet Trends [Online] Dostopno na naslovu <http://www.morgan-stanley.com/institutional/techresearch/pdfs/Internet_Trends_041210.pdf>, 10. Junij 2012

Boydlee Pollentine, Appcelerator Titanium Smartphone App Development Cookbook, Packt Publishing, 2011

»Statcounter«, [Online] Dostopno na naslovu <http://statcounter.com/>, 10. Januar 2013.

Nielsen, Jakob *Mobile Usability Update*, [Online] Dostopno na naslovu <http://www.useit.com/alertbox/mobile-usability.html>, 26. September 2011.

»Adobe« *Adobe Announces Agreement to Acquire Nitobi, Creator of PhoneGap*, , [Online] Dostopno na naslovu <http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html>, 3. Oktober 2011.

»Appcelerator Titanium Marketplace« [Online] Dostopno na naslovu <http://www.appcelerator.com/ecosystem/marketplace/>, 10. Januar 2013

»Appcelerator Titanium Kitchen Sink«, [Online] Dostopno na naslovu <http://developer.appcelerator.com/doc/kitchensink>, 10. Januar 2013

»W3C HTML Specifications«, [Online] Dostopno na naslovu (<http://www.w3.org/TR/html401/>, 20. December 2012

»W3C CSS Specifications«, [Online] Dostopno na naslovu <http://www.w3.org/TR/CSS2/>, 20. December 2012

Priloga A

Oblika začetnih strani

|  |
| --- |
| NASLOVNA STRAN DIPLOMSKEGA DELA |



Ime in priimek študenta

NASLOV DIPLOMSKEGA DELA

Diplomsko delo

Maribor, mesec LETO

|  |
| --- |
| PRVA NOTRANJA STRAN DIPLOMSKEGA DELA |

**NASLOV DIPLOMSKEGA DELA**

**Diplomsko delo**

|  |  |
| --- | --- |
| Študent(ka): | Ime in priimek |
| Študijski program: | vrsta študijskega programa[[36]](#footnote-36)  ime študijskega programa[[37]](#footnote-37) |
| Smer: | naziv smeri[[38]](#footnote-38) |
| Mentor(ica): | naziv ter ime in priimek[[39]](#footnote-39) |
| Somentor(ica): | naziv ter ime in priimek |
| Lektor(ica): | ime in priimek, naziv |
|  |  |



Številka: Številka

Datum in kraj: Datum, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 01/2010) izdajam naslednji

**SKLEP O DIPLOMSKEM DELU**

|  |  |
| --- | --- |
| 1. | **Ime Priimek,** študentu univerzitetnega študijskega programa #Program, smer #Smer, se dovoljuje izdelati diplomsko delo pri predmetu #Predmet. |
| 2. | MENTOR #Mentor  SOMENTOR #Somentor |
| 3. | **Naslov diplomskega dela:** |
|  | **#Naslov** |
| 4. | **Naslov diplomskega dela v angleškem jeziku:** |
|  | **#NaslovA** |
| 5. | Diplomsko delo je treba izdelati skladno z “Navodili za pisanje diplomskih del na dodiplomskih študijskih programih UM FERI ” in ga oddati v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) ter en izvod elektronske verzije do #Rok v referatu za študentske zadeve. |

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 3 delovnih dni.

|  |  |  |
| --- | --- | --- |
|  |  | Dekan: |
|  |  |  |
|  |  |  |

Obvestiti:

* kandidata/ko,
* mentorja,
* somentorja,
* odložiti v arhiv.

|  |
| --- |
| STRAN S SLOVENSKIM POVZETKOM |

Naslov diplomskega dela

**Ključne besede:** beseda1, beseda2, beseda3

**UDK:** xxxxxx

Povzetek

*Pred vrstilcem UDK navedite 3 do 5 ključnih besed. Dolžina povzetka naj ne presega 100 besed. Ta stran mora biti v diplomskem delu seveda oštevilčena z ustrezno rimsko številko.*

|  |
| --- |
| STRAN Z ANGLEŠKIM POVZETKOM |

Naslov diplomskega dela v angleškem jeziku

**Key words:** word1, word2, word3

**UDK:** xxxxxx

**Abstract**

*Na tej strani naj bo čim vernejši prevod strani s slovenskim povzetkom v angleščino. Tudi ta stran mora biti v diplomskem delu oštevilčena z ustrezno rimsko številko.*

Priloga B

Oblika končnih strani



**I Z J A V A O A V T O R S T V U**

|  |  |  |
| --- | --- | --- |
| Spodaj podpisani/-a |  | |
| z vpisno številko |  | |
| sem avtor/-ica diplomskega dela z naslovom: |  | |
|  |  | |
|  |  | |
| *(naslov diplomskega dela)* | | |
|  | | |
| S svojim podpisom zagotavljam, da: | | |
|  | | |
| * sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek) | | |
|  | |
| in somentorstvom (naziv, ime in priimek) | | |
|  | | |
|  | | |
| * 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 DKUM. | | |

|  |  |
| --- | --- |
| V Mariboru, dne |  |

|  |  |
| --- | --- |
|  | Podpis avtorja/-ice: |



**IZJAVA O USTREZNOSTI DIPLOMSKEGA DELA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Spodaj podpisani/-a | | |  | izjavljam, da je |
|  | | | (ime in priimek mentorja/-ice) |  |
| študent |  | | | izdelal diplomsko |
|  | | | (ime in priimek študenta/-ke) |  |
| delo z naslovom: | |  | | |
|  | |  | | |
|  | |  | | |
|  | | (naslov diplomskega dela) | | |

v skladu z odobreno temo diplomskega dela, Navodili za pisanje diplomskih del na dodiplomskih študijskih programih UM FERI in mojimi navodili.

|  |  |
| --- | --- |
| Kraj in datum: |  |

|  |  |
| --- | --- |
|  | Podpis mentorja: |



**IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA DELA IN OBJAVI OSEBNIH PODATKOV DIPLOMANTOV**

|  |  |
| --- | --- |
| Ime in priimek avtorja-ice: |  |
| Vpisna številka: |  |
| Študijski program: |  |
| Naslov zaključnega dela: |  |
|  |  |
| Mentor: |  |
| Somentor: |  |

Podpisani-a izjavljam, da sem za potrebe arhiviranja oddal elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna z elektronsko verzijo elektronski verziji, ki sem jo oddal za objavo v Digitalno knjižnico Univerze v Mariboru.

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, varstva industrijske lastnine ali tajnosti podatkov naročnika: ne sme biti javno dostopno do (datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela).

Podpisani izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zaključka študija, naslov zaključnega dela), na spletnih straneh in v publikacijah UM.

|  |  |
| --- | --- |
| Datum in kraj: | Podpis avtorja-ice: |
| Podpis mentorja:  (*samo v primeru, če delo ne sme biti javno dostopno)* |  |
| Podpis odgovorne osebe naročnika in žig:  (*samo v primeru, če delo ne sme biti javno dostopno)* |  |

1. Jordan Crook, Gartner, IDC: Windows Phone To Steal Second Place From iOS By 2015, (<http://techcrunch.com/2011/09/02/gartner-idc-windows-phone-to-steal-second-place-from-ios-by-2015>), 2. september 2011. [↑](#footnote-ref-1)
2. Firtman, Maximiliano, Programming the Mobile Web, O’Reilly Media, 2010. [↑](#footnote-ref-2)
3. Wikipedia, <http://en.wikipedia.org/wiki/Smartphone>, 4. Septmeber 2012 [↑](#footnote-ref-3)
4. Elgin, Ben, "Google Buys Android for Its Mobile Arsenal", <http://www.businessweek.com/stories/2005-08-16/google-buys-android-for-its-mobile-arsenal>, 17. Avgust 2005. [↑](#footnote-ref-4)
5. Industry Leaders Announce Open Platform for Mobile Devices (Press release), <http://www.openhandsetalliance.com/press_110507.html>, Open Handset Alliance. 5. November 2007. [↑](#footnote-ref-5)
6. Vikki Chowney, Android takes 52.5% of smartphone market. Econsultancy, <http://econsultancy.com/si/blog/8279-android-doubles-market-share-to-take-52-5-of-smartphone-market>, 15. November 2011. [↑](#footnote-ref-6)
7. Google Play for Developers, <https://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=112622&topic=15867&ctx=topic>, 5. Januar 2013 [↑](#footnote-ref-7)
8. Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year, <http://www.gartner.com/newsroom/id/1689814>,19. Maj 2011 [↑](#footnote-ref-8)
9. RIM Company – Learn about Research in Motion, RIM, <http://www.rim.com/company/>, 23. September 2012. [↑](#footnote-ref-9)
10. BlackBerry, <http://www.blackberry.com/>, 3. December 2012. [↑](#footnote-ref-10)
11. Android and iOS Surge to New Smartphone OS Record in Second Quarter, IDC, <http://www.idc.com/getdoc.jsp?containerId=prUS23638712>, 8. Avgust 2012. [↑](#footnote-ref-11)
12. Android and iOS Surge to New Smartphone OS Record in Second Quarter, IDC, <http://www.idc.com/getdoc.jsp?containerId=prUS23638712>, 8. Avgust 2012. [↑](#footnote-ref-12)
13. Android and iOS Surge to New Smartphone OS Record in Second Quarter, IDC, <http://www.idc.com/getdoc.jsp?containerId=prUS23638712>, 8. Avgust 2012. [↑](#footnote-ref-13)
14. Android and iOS Surge to New Smartphone OS Record in Second Quarter, IDC, <http://www.idc.com/getdoc.jsp?containerId=prUS23638712>, 8. Avgust 2012. [↑](#footnote-ref-14)
15. Android Expected to Reach Its Peak This Year as Mobile Phone Shipments Slow, BUSINESS WIRE, <http://www.sys-con.com/node/2291309>, 6. Junij 2012. [↑](#footnote-ref-15)
16. The Windows Phone Predictions that IDC, Gartner and Pyramid Research Probably Hope You’ve Forgotten, Joey Devilla, <http://www.globalnerdy.com/2012/05/07/the-windows-phone-predictions-that-idc-gartner-and-pyramid-research-probably-hope-youve-forgotten/>, 7. Maj 2012 [↑](#footnote-ref-16)
17. The Windows Phone Predictions that IDC, Gartner and Pyramid Research Probably Hope You’ve Forgotten, Joey Devilla, <http://www.globalnerdy.com/2012/05/07/the-windows-phone-predictions-that-idc-gartner-and-pyramid-research-probably-hope-youve-forgotten/>, 7. Maj 2012 [↑](#footnote-ref-17)
18. Android pri nas uporablja že polovica uporabnikov mobilnega spleta, Mobitel, <http://tehnik.mobitel.si/android-pri-nas-uporablja-ze-polovica-uporabnikov-mobilnega-spleta/>, 8. Marec 2012. [↑](#footnote-ref-18)
19. Statcounter, <http://statcounter.com/>, 10. Januar 2013. [↑](#footnote-ref-19)
20. Mobile Usability Update, Jakob Nielsen, <http://www.useit.com/alertbox/mobile-usability.html>, 26. September 2011 [↑](#footnote-ref-20)
21. Adobe Announces Agreement to Acquire Nitobi, Creator of PhoneGap, Adobe, <http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html>, 3. Oktober 2011 [↑](#footnote-ref-21)
22. Appcelerator Titanium Marketplace, <http://www.appcelerator.com/ecosystem/marketplace/> [↑](#footnote-ref-22)
23. Multiple phone web-based application framework, <http://en.wikipedia.org/wiki/Multiple_phone_web_based_application_framework>. [↑](#footnote-ref-23)
24. Prototyping on Paper, <http://popapp.in>. [↑](#footnote-ref-24)
25. FERI mobile app, <http://feri.mobile.nej.si/pop>. [↑](#footnote-ref-25)
26. FERI Osebje, <http://osebje.feri.uni-mb.si/cache/data/sm=7038.txt>. [↑](#footnote-ref-26)
27. Aktualne diplome, <http://dkum.uni-mb.si/Iskanje.php?type=napredno&niz0=&vrsta=dip&vir=3&chkFullOnly=on>. [↑](#footnote-ref-27)
28. Zadnje diplome, <http://www.feri.uni-mb.si/zadnjizagovori.html>. [↑](#footnote-ref-28)
29. Oglasna deska, <http://www.feri.uni-mb.si/odeska/getxml.aspx>. [↑](#footnote-ref-29)
30. Appcelerator Titanium Kitchen Sink, <http://developer.appcelerator.com/doc/kitchensink>. [↑](#footnote-ref-30)
31. W3C HTML Specifications, <http://www.w3.org/TR/html401/>. [↑](#footnote-ref-31)
32. W3C CSS Specifications, <http://www.w3.org/TR/CSS2/>. [↑](#footnote-ref-32)
33. TestFlight, <https://testflightapp.com/>. [↑](#footnote-ref-33)
34. nej/si.nej.mobile.feri github repozitorij, <https://github.com/nej/si.nej.mobile.feri>. [↑](#footnote-ref-34)
35. nej/si.nej.mobile.feri github issues, <https://github.com/nej/si.nej.mobile.feri/issues?state=open>. [↑](#footnote-ref-35)
36. npr. univerzitetni študijski program [↑](#footnote-ref-36)
37. npr. Elektrotehnika [↑](#footnote-ref-37)
38. samo, če ima program smeri [↑](#footnote-ref-38)
39. npr. red. prof. dr. Bojan Grčar [↑](#footnote-ref-39)