

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gorazd Porenta

**Razvoj mobilne aplikacije za uporabo
na različnih operacijskih sistemih
mobilnih naprav**

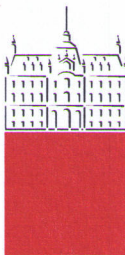
DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Rok Rupnik

Ljubljana, 2012

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 01843/2012

Datum: 02.04.2012

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

Kandidat: **GORAZD PORENTA**

Naslov: **RAZVOJ MOBILNE APLIKACIJE ZA UPORABO NA RAZLIČNIH
OPERACIJSKIH SISTEMIH MOBILNIH NAPRAV**
**THE DEVELOPMENT OF MOBILE APPLICATION FOR THE USE ON
VARIOUS OPERATING SYSTEMS ON MOBILE DEVICES**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Podrobno proučite razvojno platformo PhoneGap, ki omogoča razvoj mobilnih aplikacij za različne operacijske sisteme mobilnih naprav. Posebno pozornost posvetite možnostim, ki jih razvojna platforma nudi na področju dostopa do naslednjih sistemskih virov mobilne naprave: kamera, GPS enota in merilnik pospeškov. Z uporabo razvojne platforme PhoneGap nato razvijte mobilno aplikacijo, ki omogoča spremljanje delovanja sistema RipAdmin, ki omogoča asinhrono izmenjavo velikih količin sporočil.

Mentor:

doc. dr. Rok Rupnik

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Gorazd Porenta,

z vpisno številko 24940104,

sem avtor diplomskega dela z naslovom:

Razvoj mobilne aplikacije za uporabo na različnih operacijskih sistemih mobilnih naprav

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne 12.6.2012

Podpis avtorja:

Zahvala

Zahvaljujem se staršem, ki so mi študij omogočili, me spodbujali in podpirali. Hvala gre mentorju doc. dr. Roku Rupniku za pomoč in nasvete pri izdelavi diplomske naloge. Zahvala gre tudi prijateljem in sodelavcem, ki so sodelovali pri testiranju aplikacije. Največja zahvala gre Tini, ki me je spodbujala in mi pomagala pri oblikovanju diplomskega dela.

Kazalo

Povzetek	1
Abstract	2
1 Uvod	3
2 Opredelitev pojmov in uporabljenih tehnologij	5
2.1 Mobilna naprava	5
2.2 Mobilna aplikacija	7
2.3 HTML	8
2.4 CSS	8
2.5 JavaScript	9
2.6 Spletna storitev	9
3 Razvoj mobilnih aplikacij	10
3.1 Mobilni operacijski sistemi	11
3.1.1 iOS	11
3.1.2 Android	12
3.1.3 BlackBerry OS	13
3.1.4 Windows Phone	15
3.1.5 WebOS	16
3.1.6 Symbian OS	17
3.1.7 Bada OS	19
3.2 Orodja za razvoj mobilnih aplikacij, delujočih na več operacij- skih sistemih	19
3.2.1 PhoneGap	20
3.2.2 Rho Mobile	20
3.2.3 appMobi	21
3.2.4 appcelerator	21
3.2.5 DragonRAD	21

3.2.6	Sencha Touch	22
3.2.7	UNITY Mobile	22
3.2.8	MoSync	23
3.2.9	Qt	23
3.2.10	TotalCross	23
4	PhoneGap	25
4.1	Zgodovina	26
4.2	Podprti operacijski sistemi	26
4.3	Vzpostavitev razvojnih okolij	27
4.3.1	iOS	27
4.3.2	Android	27
4.3.3	BlackBerry	28
4.3.4	Windows Phone	28
4.3.5	WebOS	29
4.3.6	Symbian	29
4.4	Dostopnost sistemskih virov mobilnih naprav	29
4.4.1	Merilnik pospeška	31
4.4.2	Kamera	32
4.4.3	Avdiovizualne vsebine	32
4.4.4	Kompas	33
4.4.5	Omrežja	35
4.4.6	Stiki	35
4.4.7	Lastnosti naprave	36
4.4.8	Datotečni sistem	37
4.4.9	Geografska lokacija	38
4.4.10	Večpredstavnost	39
4.4.11	Opozorila	39
4.4.12	Podatkovno skladišče	40
4.5	Življenjski cikel aplikacije PhoneGap	40
5	Razvoj aplikacije z ogrodjem PhoneGap	42
5.1	Aplikacija RipAdmin	42
5.2	Arhitektura aplikacije	43
5.3	Primeri uporabe	44
5.4	Podatkovna baza	45
5.5	Uporabniki aplikacije	46
5.6	Razvoj mobilne aplikacije RipAdmin	46
5.6.1	Razvojno okolje	46

5.6.2	jQuery Mobile	47
5.6.3	Uporaba spletnih storitev	47
5.6.4	Razvoj programske logike in uporabniškega vmesnika	51
5.6.5	Prevajanje in testiranje aplikacije	54
5.6.6	Namestitev aplikacije	57
5.6.7	Uporaba aplikacije	58
6	Prihodnost PhoneGap	67
7	Sklepne ugotovitve	71
	Seznam slik	74
	Seznam tabel	76
	Literatura	77

Seznam uporabljenih kratic

- 3D** trirazsežni; ki ima tri razsežnosti, npr. trirazsežna grafika.
- API** (*angl.*) *Application Programming Interface*; programski vmesnik.
- ASCII** (*angl.*) *American Standard Code for Information Interchange*; ameriški standardni 7-bitni nabor znakov za izmenjavo informacij.
- CSS** (*angl.*) *Cascading Style Sheets*; stilna predloga na spletni strani, v kateri je zapisana oblika spletne strani.
- GPS** (*angl.*) *Global Positioning System*; sistem globalnega določanja lege.
- HSDPA** (*angl.*) *High Speed Downlink Packet Access*; tretja generacija telekomunikacijskega protokola, ki v omrežju UMTS omogoča hitrejši prenos podatkov.
- HTML** (*angl.*) *Hypertext Markup Language*; označevalni jezik za oblikovanje večpredstavnostnih dokumentov.
- HTTP** (*angl.*) *Hypertext Transfer Protocol*; protokol za prenos informacij na spletu. Prvotno je bil namenjen objavljanju in prejemanju HTML strani.
- HTTPS** (*angl.*) *Hypertext Transfer Protocol Secure*; protokol, ki omogoča varno internetno povezavo.
- IDE** (*angl.*) *Integrated Development Environment*; integrirano razvojno okolje.
- IMAP4** (*angl.*) *Internet Message Access Protocol version 4*; internetni protokol za dostop do elektronske pošte na oddaljenem strežniku.
- IPsec** (*angl.*) *Internet Protocol Security*; protokol za varno izmenjavo podatkov v medmrežju (omogoča overjanje in kodiranje).

- IrDa** (angl.) *Infrared Data Association*; interesna skupina, ki določa standarde protokolov za brezžično komunikacijo s pomočjo infrardečih žarkov.
- IS** (angl.) *Information System*; informacijski sistem.
- IT** (angl.) *Information Technology*; informacijska tehnologija.
- JDBC** (angl.) *Java Database Connectivity*; standard za dostop do baze podatkov v programskem jeziku Java.
- JS** *JavaScript*; objektni skriptni programski jezik.
- MIDP** (angl.) *Mobile Information Device Profile*; profil mobilne naprave.
- MMS** (angl.) *Multimedia Messaging System*; večpredstavnostni sporočilni sistem; nadgradnja SMS-sporočil preko katere prenašamo slike, zvok in video posnetke.
- MVC** (angl.) *Model-View-Controller*; Model-Pogled-Kontroler arhitektura aplikacije.
- NFC** (angl.) *Near Field Communication*; komunikacijska tehnologija kratkega dosega, ki omogoča izmenjavo podatkov na razdalji do 10 cm.
- OpenGL** (angl.) *Open Graphics Library*; specifikacija standarda, ki določa API za aplikacije, ki prikazujejo 2D in 3D računalniško grafiko.
- OS** (angl.) *Operating System*; operacijski sistem.
- PDA** (angl.) *Personal Digital Assistant*; osebni elektronski pomočnik, dlančnik.
- PDF** (angl.) *Portable Document Format*; odprt standard za izmenjavo elektronskih dokumentov.
- POP3** (angl.) *Post Office Protocol version 3*; internetni protokol, namenjen pridobivanju elektronske pošte s strežnika.
- SD** (angl.) *Secure Digital*; spominska kartica majhnega formata za potrebe prenosnih elektronskih naprav.
- SDK** (angl.) *Software Development Kit*; paket programskih orodij, uporabljen s strani programerja, da razvije aplikacijo za konkretno programsko opremo.

- SDL** (angl.) *Simple DirectMedia Layer*; prosta odprtokodna knjižnica za dostop do grafičnih, zvočnih in vhodnih enot.
- SMS** (angl.) *Short Message Service*; sistem za pošiljanje krajših besedilnih sporočil z mobilnim telefonom.
- SMTP** (angl.) *Simple Mail Transfer Protocol*; protokol za prenos elektronske pošte.
- SOAP** (angl.) *Simple Object Access Protocol*; specifikacija za izmenjavo strukturiranih podatkov pri izvajanju spletnih storitev v računalniških omrežjih.
- SQL** (angl.) *Structured Query Language*; strukturirani povpraševalni jezik za delo s podatkovnimi zbirkami.
- SSL** (angl.) *Secure Socket Layer*; protokol za kodiranje podatkov v medmrežju.
- UMTS** (angl.) *Universal Mobile Telecommunications System*; univerzalni sistem za mobilno komunikacijo.
- URI** (angl.) *Uniform Resource Identifier*; enoten in enoličen niz znakov, s katerim se opredelita ime in/ali lokacija vira, navadno na medmrežju.
- W3C** (angl.) *World Wide Web Consortium*; mednarodni inštitut, kjer člani organizacije, osebje s polnim delovnim časom in javnost sodelujejo ter skupaj razvijajo standarde za splet.
- WAP** (angl.) *Wireless Application Protocol*; protokol za prilagajanje spletnih storitev mobilni telefoniji.
- WiFi** tehnologija brezžičnega prenosa podatkov preko računalniškega omrežja.
- WSDL** (angl.) *Web Services Description Language*; jezik, ki temelji na XML, uporaben za opis funkcionalnosti, ki jo nudi spletna storitev.
- WWW** (angl.) *World Wide Web*; svetovni splet.
- XDK** (angl.) *Cross Platform Development Kit*; paket programskih orodij, uporabljen s strani programerja, ki razvija aplikacijo za več različnih sistemov.

XHTML (*angl.*) *Extensible Hypertext Markup Language*; označevalni jezik, ki ima enak namen kot HTML, vendar je usklajen s sintakso XML.

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

Povzetek

Tehnološki napredek mobilnih naprav, razširjenost širokopasovnih mobilnih omrežij in nenazadnje cenovna dostopnost so razlogi za njihovo razširjenost. Vsebujejo zmogljive mikroprocesorje, velike pomnilniške kapacitete in komponente, ki razširjajo njihovo uporabnost (kamera, GPS-enota itd.). Na napravah oziroma pripadajočih mobilnih operacijskih sistemih so nameščene aplikacije, ki bodisi izkoriščajo vgrajene komponente bodisi bogatijo funkcionalnost. Uporabniki imamo možnost, da obstoječim dodamo nove, ki jih pridobimo v spletnih trgovinah mobilnih aplikacij ali jih razvijemo sami. Za razvoj potrebujemo razvojno okolje, primerno operacijskemu sistemu naprave. Razvita programska oprema ima omejitve, da bo tekla samo na napravah z operacijskim sistemom, za katerega je bila razvita. Alternativa je razvoj spletne aplikacije, prilagojene mobilnim napravam, vendar iz tovrstnih aplikacij ni mogoč dostop do sistemskih virov naprave, kot so kamera, enota GPS, stiki, koledar itd. Razvoj domorodnih aplikacij, ki bi jih bilo mogoče izvajati na več platformah, nam otežuje raznovrstnost mobilnih operacijskih sistemov. Posledično se v svetu pojavljajo skupnosti, ki se trudijo ponuditi enotno razvojno okolje, ki je izvorno kodo sposobno prevesti v domorodne aplikacije za več operacijskih sistemov.

V diplomskem delu bomo z enim izmed njih (PhoneGap) razvili mobilno aplikacijo. PhoneGap temelji na uporabi označevalnega jezika HTML, objektnega skriptnega jezika JavaScript in stilnih predlog strani CSS. Na ta način razvita aplikacija teče v istoimenskem izvajalnem okolju na sedmih različnih mobilnih operacijskih sistemih.

Ključne besede:

PhoneGap, mobilna naprava, razvoj mobilnih aplikacij

Abstract

Technological development of mobile devices, ripeness of broadband mobile networks and affordable price are the reasons that made them so widespread. They incorporate powerful microprocessors, large memory capacities and other components (camera or GPS unit) that even further expand their serviceability. Applications that take advantage of built-in components or improve their functionality are installed on such devices and their associated mobile operating systems. Users have an opportunity to add new applications, which can be obtained from online mobile applications markets, or to develop their own applications. In such cases, we need an environment applicable to the device's operating system. However, the developed software has a restriction and can run only on devices with an operating system for which it has been developed for. An alternative is to develop a web-application adapted to mobile devices, but in these applications, it is not possible to access the system resources, such as a camera, a GPS unit, contacts, a calendar, etc. The development of native applications, which could run on several platforms, prevents us from having various mobile operating systems. Consequently, some communities have been set up around the world trying to offer a unified development environment capable of compiling source codes into native applications that can be used on multiple operating systems. In my undergraduate thesis, we will use one of them – PhoneGap – and try to develop a mobile application. PhoneGap is based on the HTML markup language, JavaScript scripting language and cascading style sheets (CSS). The application developed in this way can run on seven different mobile operating systems in the same-named working environment.

Key words:

PhoneGap, mobile device, mobile application development

Poglavje 1

Uvod

V zadnjih letih smo priča neverjetnemu razvoju informacijsko-telekomunikacijske opreme. Telekomunikacijska podjetja so vlagala izdatna sredstva v razvoj omrežij, ki danes poleg govorne telefonije omogočajo tudi širokopasovni prenos podatkov. Vzporedno z razvojem in dostopnostjo telekomunikacijskih omrežij se je velik del aplikacij deloma ali v celoti selil v splet. Hkrati je razvoj prenosnih telefonov in drugih mobilnih naprav ponudil naprave z vse večjim poudarkom na povezljivosti v omrežja, ki omogočajo prenos podatkov. Preproste naprave, ki so v preteklosti omogočale zgolj telefonske klice, danes vsebujejo veliko število komponent, ki dopolnjujejo osnovno funkcionalnost (barvni zaslone, GPS-enota, kamera ...). Mobilne naprave dandanes vsebujejo zmogljive mikroprocesorje, obsežne pomnilniške kapacitete in barvne zaslone spodobnih ločljivosti. Prednost pred namiznimi in prenosnimi računalniki je v manjši velikosti, kar olajša njihovo prenosljivost. Uporabnost se je praktično približala prenosnim računalnikom, kar je sprožilo zahtevo po zmogljivejši programski opremi. Ob vse več napravah z vgrajenimi zaslone, občutljivimi na dotik, ki vse redkeje vsebujejo tipkovnico, se je spremenil tudi način njihove uporabe. Posledično se je na trgu mobilnih naprav pojavilo precejšnje število različnih operacijskih sistemov. Po navadi omogočajo hiter ter enostaven razvoj in namestitev namenskih aplikacij. Programsko opremo lahko razvijamo za lastno ali splošno uporabo. Slednjo lahko objavimo v spletnih trgovinah z mobilnimi aplikacijami brezplačno ali proti plačilu. Razvoj aplikacij, namenjenih širšemu krogu uporabnikov, omejuje veliko število raznovrstnih operacijskih sistemov. Burno dogajanje na trgu mobilnih naprav, kjer se delež naprav z nekim operacijskim sistemom dnevno spreminja, pa zahteva nenehno prilagajanje razvojnih okolij. Tabela 1.1 nam prikazuje trenutno stanje na trgu operacijskih sistemov.

Težavam se deloma izognemo, če imamo dostop do širokopasovnega mobil-

Operacijski sistem	4Q11 število	Delež 4Q11 %	4Q10 število	Delež 4Q10 %
Android	75.906,1	50,9	30.801,2	30,5
iOS	35.456,0	23,8	16.011,1	15,8
Symbian	17.458,4	11,7	32.642,1	32,3
RIM	13.184,5	8,8	14.762,0	14,6
Bada	3.111,3	2,1	2.026,8	2,0
Microsoft	2.759,0	1,9	3.419,3	3,4
Ostali	1.166,5	0,8	1.487,9	1,5
Skupno	149.041,8	100,0	101.150,3	100,0

Tabela 1.1: Svetovna prodaja pametnih telefonov po operacijskem sistemu v 4. četrtletju 2011 (v tisočih); Vir: Gartner (Februar 2012) [15].

nega omrežja in ne potrebujemo dostopa do sistemskih virov mobilne naprave. Ena izmed možnosti je nov razvoj ali prilagoditev spletne aplikacije, primerne okolju mobilnih naprav. Treba se je zavedati, da sodobne mobilne naprave zahtevajo prilagojen uporabniški vmesnik, s težnjo po minimalni uporabi tipkovnice in po uporabi zaslona, občutljivega na dotik. Razvoj z možnostjo izvajanja na več operacijskih sistemih zahteva vzpostavitev razvojnega okolja za vsakega izmed njih. Deloma je mogoče v istem razvojnem okolju razvijati za več mobilnih platform. Vseeno je nenehno spremljanje dogajanja na mobilnem trgu zaradi velikega števila ponudnikov izredno težavna naloga. Treba je posodabljanje razvojna okolja ter aplikacije in opuščati razvoj za okolja, ki jih na trgu ni več. Ob pojavu novega mobilnega operacijskega sistema je treba na novo razviti celotno aplikacijo. Ob vseh prej naštetih težavah se programerji v večini primerov odločijo razviti programsko opremo samo za najbolj razširjene platforme.

Največji izziv je, kako čim enostavneje razviti aplikacijo za čim več platform hkrati. Težimo k izrabi obstoječega znanja kadrov, ki v današnjih časih bolj ali manj razvijajo v spletnih tehnologijah, kakršne so HTML, CSS in JavaScript (JS). Na spletu je na voljo lepo število rešitev, ki se sicer razlikujejo, ponujajo pa natanko to, kar iščemo.

Poglavje 2

Opredelitev pojmov in uporabljenih tehnologij

Preden se lotimo razvoja mobilne aplikacije, delujoče na več operacijskih sistemih je primerno, da opredelimo nekaj osnovnih pojmov, o katerih bomo govorili v nadaljevanju.

2.1 Mobilna naprava

Mobilna naprava [5] je elektronska naprava. Njena velikost omogoča enostavno prenašanje. Po navadi je opremljena z zaslonom občutljivem na dotik in/ali miniaturno tipkovnico. V primeru dlančnika¹ (PDA) je vhodna/izhodna enota združena v obliki zaslona občutljivega na dotik. Pametni telefoni in dlančniki so priljubljeni med tistimi, ki potrebujejo funkcionalnost, podobno osebnim računalnikom, a jim je pomembna enostavna prenosljivost. Dlančniki so po navadi nadgrajeni z enotami za branje podatkov, kakršne so recimo čitalci črtne kode ali bralniki pametnih kartic. Glavni tipi današnjih mobilnih naprav so:

- mobilni telefon,
- pametni mobilni telefon,
- dlančnik,
- tablični računalnik,

¹osebni elektronski pomočnik

- prenosni računalnik,
- igralna konzola,
- predvajalnik večpredstavnih vsebin (angl. Mobile Internet Device),
- pozivnik (angl. pager),
- navigacijska naprava.

Naprave, ki imajo vgrajeno tipkovnico, so vse redkejše. Interakcija z napravo poteka na podlagi prepoznave gibov na zaslonu, občutljivem na dotik. Na tem področju je izredno inovativno podjetje Apple, ki je v zadnjih letih na svojih napravah ponudilo kar nekaj novih kretenj. Nekaj značilnih gibov, ki jih uporabljamo pri zaslonih na dotik je predstavljenih v tabeli 2.1.



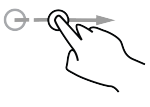

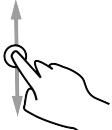
Naziv	Gib	Opis
tap / enojni pritisk		časovno kratek pritisk; običajno namenjen izbiri menija ali zagonu aplikacije
double tap / dvojni pritisk		dva hitra kratka pritiska; običajno namenjen približevanju prikazanega na ekranu
flick / hiter poteg		hiter poteg; v vertikalni smeri običajno premik prikazanega v smeri potega, v horizontalni običajno prehod na drugo aplikacijo
touch and hold / pritisni in drži		pritisni in drži; običajno prikaže namig ali predlog
pan / pomik		z enim ali več prsti pritisni in premakni; običajno omogoča pomik preko zaslonov z drsnikom

Tabela 2.1: (se nadaljuje)

Tabela 2.1: (nadaljevanje)


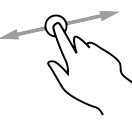


Naziv	Gib	Opis
rotate / obrni		dvoprstno obračanje; obrne vsebino na zaslonu
click & drag / izberi in povleci		izberi element na ekranu in povleci; premakne izbran element na zaslonu na izbrano mesto
pinch open / izberi in raztegni		z dvema ali več združenimi prsti izberi in raztegni; običajno postopna povečava
pinch close / izberi in združi		z dvema ali več razdruženimi prsti izberi in združi; običajno postopna pomajšava

Tabela 2.1: Primeri gibov na zaslonih občutljivih na dotik [23].

2.2 Mobilna aplikacija

Mobilna aplikacija [6] je vrsta programske opreme, namenjena delovanju na mobilnih napravah, kakršne so pametni telefoni ali tablični računalniki. Mobilne aplikacije pogosto uporabnikom nudijo storitve, podobne tistim na osebnih računalnikih. V splošnem so take aplikacije majhne programske enote z omejeno funkcionalnostjo. Zanje je značilno, da jih pridobimo preko spletnih trgovin, specializiranih za njihovo prodajo. Mobilne aplikacije se razlikujejo od integriranih sistemov programske opreme, ki jih najdemo na osebnih računalnikih, v omejenem številu posameznih funkcij. Najenostavnejše mobilne aplikacije enostavno prenesemo iz osebnih računalnikov v mobilno napravo, kar pa za stabilnejše aplikacije ni primerna tehnika. Primernejši način je samostojen razvoj, namenjen izdelavi mobilne aplikacije, v kateri upoštevamo tako omejitve kakor prednosti mobilne naprave. Eden izmed primerov je pristop do razvoja aplikacije, ki uporablja enoto GPS (določanje geografske lege),

in se na mobilnih napravah razlikuje od razvoja na osebni računalnikih.

2.3 HTML

Je označevalni jezik za oblikovanje večpredstavnostnih dokumentov [7], ki omogoča povezave znotraj dokumenta ali med dokumenti. Predstavlja osnovo spletnega dokumenta. Z jezikom HTML, poleg prikaza dokumenta v brskalniku, hkrati določimo tudi strukturo in semantični pomen delov dokumenta. Izdelamo ga lahko v vsakem urejevalniku besedil (npr. Beležnici). Zapisan je v obliki HTML-elementov, ki so sestavljeni iz značk (angl. tag), zapisanih v lomljenih oklepajih (npr. kot `<p>`) znotraj vsebine spletne strani. HTML-značke so običajno zapisane v parih, kot npr. `<p>` in `</p>`. Prva značka se imenuje tudi začetna značka in druga značka končna značka. Znotraj para značk lahko oblikovalec spletne strani vpiše poljubno besedilo in tudi druge značke.

2.4 CSS

Stilna predloga na spletni strani, poznana pod kratico CSS [8], predstavljena v obliki preprostega slogovnega jezika, ki skrbi za predstavitev spletnih strani. Definira stil elementov HTML oz. XHTML v smislu pravil, kako naj se ti prikažejo na strani. Določamo lahko barve, velikosti, odmike, poravnave, obrobe, pozicije in vrsto drugih atributov, prav tako pa lahko nadziramo aktivnosti, ki jih izvajamo nad elementi strani (npr. prekritje povezave z miško). Predloge so bile razvite z namenom konsistentnega načina podajanja informacij o stilu spletnim dokumentom.

Poleg določanja pravil je bistvo uporabe CSS ločitev strukture strani, ki jo podaja označevalni jezik skupaj z vsebino, od njene predstavitve. S tem omogočimo lažje urejanje in dodajanje stilov ter poskrbimo za večjo preglednost dokumentov, temelječih na HTML-sintaksi. Prav tako zmanjšamo ponavljanja kode, saj omogočimo množici strani uporabo istih podlog. Tako lahko bistveno zmanjšamo njihovo velikost.

Specifikacija in vzdrževanje CSS je v rokah skupine CSS Working Group znotraj organizacije World Wide Web Consortium (W3C). Specifikacijo uradno potrdijo člani W3C (podjetja, kot so Adobe, IBM, Microsoft idr.) in tako nastanejo CSS priporočila. Ta pa niso omejena samo na spletne brskalnike. Uporabljajo se tudi v programih za pisanje dokumentov in elektronskih preglednic ter na napravah, kakršne so prenosni telefoni in dlančniki.

2.5 JavaScript

JavaScript [9] 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 lahko sodeluje s HTML-kodo in s tem poživi stran z dinamičnim izvajanjem. JavaScript podpirajo velika programska podjetja. Kot odprt jezik ga lahko uporablja vsakdo, ne da bi pri tem potreboval licenco. Podpirajo ga vsi novejši spletni brskalniki. Sintaksa jezika JavaScript ohlapno sledi programskemu jeziku C. Prav tako kakor programski jezik C JavaScript nima vgrajenih vhodno-izhodnih funkcij, zato je njihova izvedba odvisna od gostitelja.

JavaScript se veliko uporablja za ustvarjanje dinamičnih spletnih strani. Program se vgradi ali pa vključi v HTML, da opravlja naloge, ki niso mogoče na statični strani. Primeri so odpiranje novih oken, preverjanje pravilnosti vnesenih podatkov, enostavni izračuni ipd. Na žalost različni spletni brskalniki izpostavljajo različne objekte za uporabo. Za podporo vseh brskalnikov je zato treba napisati več različic funkcij. Izven spleta se JavaScript uporablja v različnih orodjih. Adobe Acrobat in Adobe Reader ga podpirata v datotekah PDF. Podpirata ga tudi operacijska sistema Microsoft Windows in Mac OS X. Programi imajo svoje objektne modele, ki zagotavljajo dostop do gostiteljevega okolja. Samo jedro jezika JavaScript pa je v vseh programih večinoma enako.

2.6 Spletna storitev

Spletna storitev [10] je način komunikacije med dvema elektronskima napravama preko spleta. W3C določa "spletno storitev" kot aplikativni sistem, izdelan za podporo medsebojni interakciji med dvema napravama preko omrežja. Vmesnik, ki ga ponuja spletna storitev, je opisan z jezikom WSDL. Sistemi komunicirajo s spletnimi storitvami z uporabo SOAP-sporočil, ki se običajno prenašajo preko HTTP-protokola z uporabo XML in drugih, s spletom povezanih standardov.

Poglavje 3

Razvoj mobilnih aplikacij

Programska oprema je bila na prvih prenosnih telefonih sorazmerno preprosta, zato je bila v veliki večini primerov zapisana na nekem čipu. Tehnološki razvoj je napravam omogočil, da so vsebovale vse več dodatnih komponent (fotoaparat, modul za brezžično omrežje, GPS-enota itd.), ki so jih vse bolj oddaljevale od njihovega prvotnega namena, služiti le vzpostavljanju telefonskih klicev. Vzporedno je programska oprema zanje postajala vse kompleksnejša. Pametni telefoni (angl. smart phones) in druge mobilne naprave so v danem trenutku postali alternativa manjšim prenosnikom. Novi način uporabe in pojav zaslonov, občutljivih na dotik, je ponujal nov segment pri razvoju programske opreme kot premostitvenega dejavnika med strojno opremo in uporabniškim vmesnikom. Področje je hitro zbudilo zanimanje pri proizvajalcih mobilnih naprav (npr. Apple) in drugih, ki so predvidevali, da se bo trend obrnil v prid mobilnosti (npr. Google). Resnici na ljubo je treba priznati, da nekateri niso videli take prihodnosti in se v zadnjem času trudijo nadoknaditi zamujeno (npr. Microsoft). Operacijski sistem nudi uporabnikom, da si okolje mobilnih naprav prilagodijo svojim željam. Tako na svoj način izkoristijo zmožnosti naprav ob namestitvi aplikacij, ki so jih razvili sami ali pa so jih razvili drugi in jih ponudili v spletnih trgovinah z mobilnimi aplikacijami. Razvijalci se trudijo proti plačilu ali brezplačno ponuditi čim večje število aplikacij, ki so jih bodisi uporabljali na osebnih računalnikih bodisi izkoriščajo nove možnosti, ki jih ponujajo mobilne naprave. Ob tem trčijo ob prepreko raznovrstnosti tako strojne opreme kot operacijskih sistemov. Razvoj domorodne (angl. native) programske opreme je močno odvisen od strojne opreme in operacijskega sistema, za katera se razvija. S širjenjem uporabe pametnih telefonov in drugih mobilnih naprav postaja jasno, da je tendenca, da z razvojem pokrijemo kar se da širok krog uporabnikov. Ena izmed težav je vsekakor buren trg mobil-

nih naprav, na katerem skoraj ni mogoče predvideti smeri gibanja razširjenosti posameznih mobilnih operacijskih sistemov. Lahko se zgodi, da v nekem trenutku postane podpora enemu izmed operacijskih sistemov odveč, saj mogoče ni več prisoten. V nadaljevanju bomo spoznali nekaj mobilnih operacijskih sistemov in osnovnih značilnosti, proti koncu poglavja pa rešitve, ki so na voljo, da premostimo potrebo po razvoju za vsak operacijski sistem posebej.

3.1 Mobilni operacijski sistemi

3.1.1 iOS

iOS [11] (do junija 2010 znan tudi kot iPhone OS) je Applov operacijski sistem za mobilne naprave. Sprva je bil namenjen prenosnemu telefonu iPhone, danes pa se uporablja na napravah, kakršne so iPod touch, iPad in Apple TV. Najnovejša različica iOS nosi oznako 5.

Uporabniški vmesnik sistema iOS temelji na konceptu direktne manipulacije za uporabo večprstnih dotikov¹ (angl. multi-touch). Elementi vmesnika so sestavljeni iz drsnikov, stikal in gumbov. Odziv na dotik uporabnika je takojšen, kar omogoča tekoče delovanje sistema. Vgrajeni senzorji za zaznavanje pospeška (angl. accelerometer) služijo aplikacijam za zaznavo tresljajev naprave in tridimenzionalno orientacijo. V tabeli 3.1 je prikaz različic iOS in naprav, na katerih je mogoča namestitvev.

Različica	Podprte naprave
3.1.3	iPhone; iPod Touch (1. generacija)
4.2.1	iPhone 3G; iPod Touch (2. generacija)
5.0.1	iPhone 3GS; GSM/CDMA iPhone4, iPhone 4S; iPod Touch (3. in 4. generacija); iPad, iPad 2
5.1	iPhone 3GS; GSM/CDMA iPhone4, iPhone 4S; iPod Touch (3. in 4. generacija); iPad, iPad 2

Tabela 3.1: Različice iOS in podprte mobilne naprave.

Podjetje Apple uveljavlja strogo (omejujočo) licenčno politiko, zato za razvoj potrebujemo njihovo strojno in programsko opremo. Če želimo lastne

¹Lastnost vhodne enote naprave, da zazna tri ali več dotikov hkrati.

aplikacije prenesti na eno izmed njihovih platform, potrebujemo razvojno digitalno potrdilo (angl. iOS developer certificate). Razvojni certifikat stane 99 ameriških dolarjev za javne brezplačne in plačljive aplikacije ter 299 ameriških dolarjev za lastniške poslovne programe za interno rabo. Aplikacijo lahko brez potrdila poganjamo v posnemovalniku (angl. emulator). Aplikacije razvijamo v programskem jeziku objektnem C (angl. Objective-C), ki je objektno orientiran in izhaja iz C++. Okolje temelji na aplikacijskem vmesniku Cocoa Touch, ki s paketom UIKit omogoča hiter in enostaven razvoj programske opreme za iOS. Paket namreč vsebuje elemente grafičnega vmesnika, nadzor nad njimi in možnost proženja akcij ob različnih dogodkih. Razvojno okolje je sestavljeno iz orodja Xcode in iOS SDK (angl. Software Development Kit).

3.1.2 Android

Android [13] je mobilni operacijski sistem, katerega avtor je skupnost Open Handset Alliance. Skupina pod vodstvom Googla združuje vodilna svetovna podjetja na področju mobilnih tehnologij ter strojne in programske opreme. Za razliko od iOS je Android odprta platforma, kar udeležencem v skupnosti in razvijalcem omogoča vpogled v operacijski sistem, uporabljene knjižnice in aplikacije. Največja prednost pristopa je v tem, da zaradi ažurnosti sistema v celoti izkoristimo tudi najnovejše funkcionalnosti mobilnih naprav. Dostop do sistemskih virov naprave (kamere, pošiljanje SMS-sporočil, opravljanje telefonskih klicev itd.) omogoča razvoj bogatejših in uporabniku prijaznejših aplikacij. Osnovan je na operacijskem sistemu Linux in njegova odprtost tudi za vnaprej zagotavlja, da se bo sproti prilagajal prihajajočim novim tehnologijam. Android poganja precejšnje število mobilnih naprav različnih proizvajalcev, medtem ko je Apple edini proizvajalec naprav za iOS. Google okvirno priporoča obliko naprav, ki naj bi jih poganjal njihov sistem, med drugim recimo gumbi "Home", "Menu" in "Back". V tabeli 3.2 so predstavljene različice operacijskega sistema in razširjenost (merjeno v številu dostopov do spletnega portala Android Market).

Google omogoča orodja za programiranje v programskem jeziku C/C++. Prednost je v hitrosti izvajanja, optimalnejšem izkoriščanju virov in neposredni uporabi strojne opreme. Slaba plat takega načina je v zmanjšani možnosti prenosa med napravami in bolj zapleten razvoj. Priporočljiva je uporaba programskega jezika Java, ki nam omogoča popoln dostop do komponent naprave in razvojnega okolja Eclipse IDE [14]. To je odprtokodna rešitev, ki jo razvija skupnost Eclipse Foundation. Projekti skupnosti so usmerjeni v oblikovanje odprte razvojne platforme, sestavljene iz razširljivega ogrodja, orodij in iz-

Različica	Kodno ime	API nivo	Delež
1.5	Cupcake	3	0,6%
1.6	Donut	4	1,0%
2.1	Eclair	7	7,6%
2.2	Froyo	8	27,8%
2.3– 2.3.2	Gingerbread	9	0,5%
2.3.3– 2.3.7		10	58,1%
3.0	Honeycomb	11	0,1%
3.1		12	1,4%
3.2		13	1,9%
4.0– 4.0.2	Ice Cream Sandwich	14	0,3%
4.0.3		15	0,7%

Tabela 3.2: Različice platforme Android, kodna imena, API nivo in delež dostopov do Android Market [16] (Februar 2012).

vajalnih okolij (angl. runtimes). Orodje je izredno prilagodljivo in omogoča enostavno dopolnjevanje z novimi možnostmi v obliki vtičnikov (angl. plugin), kar lajša pisanje programov, razhroščevanje (angl. debug) in testiranje v posnemovalniku.

3.1.3 BlackBerry OS

Operacijski sistem BlackBerry OS je proizvod kanadskega podjetja Research In Motion (RIM) od leta 1999. Kot prvi je ponudil kontekstno odvisno mobilno aplikacijo elektronske pošte, ki v trenutku prispetja elektronskega sporočila letga posreduje v mobilno napravo. Operacijski sistem podpira večopravnost (angl. multitasking) in posebne vhodne enote, sprejete s strani RIM, zlasti sledilno kroglico (angl. trackball), sledilno kolesce (angl. trackwheel), sledilno ploščico (angl. trackpad) in zaslon na dotik (angl. touch screen). Platforma je najbolj poznana po domorodni podpori elektronski pošti po standardu MIDP 1.0 in v zadnjih različicah MIDP 2.0, ki omogoča brezžično sinhronizacijo z elektronsko pošto (Microsoft Exchange, Lotus Domino ali Novell GroupWise),

koledarjem, opravili, zapiski in imenikom ob uporabi aplikacijskega strežnika BlackBerry Enterprise Server. Podpira tudi WAP 1.2. V tabeli 3.3 so predstavljene različice BlackBerry OS in podprte mobilne naprave.

Različica	Podprte naprave: BlackBerry serije 9000
5.0	BlackBerry Bold 9000, BlackBerry Storm, BlackBerry Storm 2 and BlackBerry Tour
6.0	BlackBerry Bold 9650/9700/9780, BlackBerry Curve 9300/9330, BlackBerry Pearl 9100, BlackBerry Style 9680, and BlackBerry Torch 9800
7.0	BlackBerry Bold 9790/9900/9930, BlackBerry 9350/9360/9370/9380, BlackBerry Torch 9810/9850/9860.
7.1	V beta testiranju

Tabela 3.3: Različice BlackBerry OS in podprte naprave BlackBerry.

BlackBerry [17] je odprta platforma, ki omogoča obilo možnosti za razvoj aplikacij, in sicer:

- Razvoj z uporabo domorodne SDK, ki nudi razvoj v programskem jeziku C/C++. Uporabimo ga v primeru, ko želimo najvišji nivo učinkovitosti pri izrabi virov (primer 3D igre) ali ko želimo dostopati do enot naprave, ki iz drugih razvojnih okolij ni možna.
- Razvojno okolje BlackBerry WebWorks omogoča razvoj samostojnih aplikacij z uporabo spletnih tehnologij HTML5, JavaScript in CSS. Omogoča tudi razvoj z uporabo sodobnih mobilnih spletnih ogrodij, kakršna so Sencha, PhoneGap, jQuery ali Dojo. Tak način ponuja hitrejši razvoj s ponovno uporabo obstoječega znanja o spletnih tehnologijah.
- BlackBerry Tablet OS SDK za Adobe Air je razvojno orodje, ki omogoča uporabo tehnologij Adobe Action Script in Adobe Flex. SDK API razširja (angl. extend) Action Script 3, ki ponuja nekatere napredne gradnike uporabniškega vmesnika in vnaprej definirane tematske videze uporabniškega vmesnika (angl. skins). Ob tem podpira tudi upravljanje dogodkov (angl. event listener), ki so specifični operacijskemu sistemu BlackBerry Tablet OS. OS lahko zazna, ko uporabnik s prstom zdrsne (angl. swipe) z zgornjega okvira na zaslonu. V orodju je na voljo tudi

Adobe Flash API, s katerim na enak način uporabimo enote, specifične mobilnim napravam (npr. merilec pospeška in geografski položaj).

- Izvajalno (angl. runtime) okolje BlackBerry lahko uporabimo za poganjanje aplikacij, namenjenih operacijskemu sistemu Android 2.3.3. Treba je zagotoviti, da aplikacijo Android ponovno sklopimo (angl. repackage) v BAR-datoteko, katere format je kompatibilen zahtevanemu formatu aplikacij, ki jih želimo poganjati na platformi BlackBerry. API ne podpira vseh funkcij, ki so na voljo v Androidu, zato obstaja možnost, da nekatere izmed njih ne bodo ali bodo le deloma delovale.
- RIM omogoča razvoj tudi v programskem jeziku Java ob uporabi BlackBerry Java SDK, ki vsebuje API-pakete za posamezno področje (funktionalnost), ki ga pokriva: življenjski cikel aplikacije, uporabniški vmesnik, integracijo aplikacij, PIM² (angl. Personal Information Management), upravljanje s sporočili, omrežno komunikacijo, upravljanje s sistemskimi resursi, dostop do lastnosti naprave, shranjevanje podatkov (datotečni sistem, podatkovna baza SQLite, trajna hramba (angl. Persistent Store), začasna hramba (angl. Runtime Store), hramba zapisov (angl. Record Store)), upravljanje z večpredstavnostnimi vsebinami (angl. multimedia), geografsko lokacijo in varnost. BlackBerry SDK vsebuje vse Java ME (Micro Edition) API-je in specifične API-je, namenjene izključno njihovim napravam. BlackBerry je do nedavna ponujal svoje razvojno okolje za razvoj v Javi (Legacy BlackBerry JDE), vendar je nadaljnji razvoj opustil in priporoča uporabo orodja Eclipse z vtičnikom (angl. plug-in) BlackBerry.

3.1.4 Windows Phone

Mobilnim napravam in njihovim operacijskim sistemom se ni mogel izogniti niti Microsoft. Windows Phone je naslednik operacijskega sistema Windows Mobile. Razvojni oddelek ni uspel pripraviti združljivega (angl. compatible) naslednika zaradi časovne stiske, saj je imel predhodnik preveč pomanjkljivosti, da bi uspel vzdržati v boju s konkurenco. Zdi se, da je podjetje na področju mobilnosti zaspalo in se močno trudi, da bi ujelo stik z naprednimi tehnologijami, ki so na drugih platformah že podprte. Uporabniški vmesnik se deli na dva dela: začetni zaslon in programski zaslon. Na prvega dodajamo večkrat uporabljene bližnjice, funkcije in osebe, na drugem pa so bližnjice do

²upravljanje osebnih podatkov (imenik, koledar, opravila, zapiski)

vseh nameščenih programov. Sistem z različico 7.5 omogoča večopravilnost, uvoz oziroma povezovanje stikov s profili na Facebooku³ in Windows Live⁴.

Prva različica operacijskega sistema Windows Phone z oznako 7 je bila predstavljena javnosti v februarju 2010. Doslej je doživela eno samo posodobitev, in sicer v februarju 2011, ko je bila predstavljena različica 7.5 s kodnim imenom Mango.

Razvoj za platformo zahteva Windows Phone SDK 7.1 [18], s katerim je možno graditi aplikacije za obe različici operacijskega sistema. Paket vsebuje Visual Studio 2010 Express za Windows Phone, posnemovalnik mobilne naprave Windows Phone, XNA Game Studio, Expression Blend za Windows Phone, primere aplikacij in dokumentacijo. Če je Visual Studio 2010 Professional ali poznejši že nameščen, se v razvojno okolje namestijo samo moduli, ki omogočajo razvoj mobilnih aplikacij. V orodju je mogoče pisanje kode z avtomatskim zaključevanjem kode (angl. autocomplete), prevajanje, razhroščevanje, testiranje in poganjanje aplikacij v posnemovalniku. Programsko kodo pišemo v programskem jeziku C#, objektne jeziku, izhajajočem iz C. Dodatno si lahko pri razvoju pomagamo še z naslednjima orodjema:

- *Silverlight*; Ogrodje, ki je neodvisno od platforme, na kateri se izvaja, ali spletnega brskalnika. Omogoča razvoj uporabniku prijaznih interaktivnih programov za splet, kot namizne aplikacije (angl. desktop applications) ali kot aplikacije Windows Phone.
- *XNA*; Ogrodje sestavljajo programska oprema, storitve in viri (angl. resources), ki so namenjeni razvoju iger na igralnih platformah Microsoft. Tehnologija profesionalnim razvijalcem omogoča hitro gradnjo iger za Windows Phone, Xbox 360, Zune HD in Windows 7.

3.1.5 WebOS

WebOS je izdelek podjetja Palm, ki je danes v lasti podjetja HP. Je eden izmed najmlajših mobilnih operacijskih sistemov, prvič izdan v letu 2009. HP je v avgustu 2011 napovedal ukinitve izdelave mobilnih naprav z njihovim operacijskim sistemom. Razvojni oddelki operacijskega sistema so bili namreč v negotovosti o njihovi prihodnosti. A je februarja 2012 doživel izdajo prvega dela (Enyo) kot odprte kode pod licenco Apache 2.0, kar naj bi se nadaljevalo do konca leta 2012, ko naj bi bil na voljo celoten sistem WebOS 1.0. Sistem

³<http://www.facebook.com>

⁴<https://home.live.com>

[19] je sestavljen iz treh delov: Enyo (razvojno ogrodje JavaScript), WebKit/Isis (standardno jedro Linux X) in upravitelj sistema WebOS (angl. system manager).

Prvo različico je Palm predstavil javnosti januarja 2009 kot PalmOS. Sledil je intenziven razvoj do današnje različice 3.0.5 (izšel januarja 2012).

Za razvoj na platformi potrebujemo HP WebOS SDK (zadnja različica 3.0.5) in PDK (angl. Plug-in Development Kit), ki je že vključen v sam SDK. Orodje že vsebuje vse knjižnice, potrebne za razvoj v spletnih tehnologijah (HTML, JavaScript in CSS) in programskem jeziku C/C++. SDK je del, ki podpira JavaScript, in vključuje orodja, ki jih poganjamo iz ukazne vrstice (angl. command line tools), potrebne za pakiranje, nalaganje (angl. download) in razhroščevanje aplikacij. Najdemo tudi primere aplikacij, referenčno kodo in dokumentacijo. Razvijalcem v C/C++ ponuja PDK, ki vključuje prevajalnik GCC (angl. compiler), SDL in knjižnice OpenGL, predloge (angl. template) projektov, primere izvorne kode, skripte, pomožne programe (angl. utility) in dokumentacijo. Mogoča je namestitev na osebne računalnike z operacijskimi sistemi Windows, OS X in Ubuntu Linux. Opcijsko se pri razvoju v jeziku C/C++ SDK integrira v okolje Visual Studio na platformi Windows in Xcode na OS X. SDK lahko uporabljamo iz ukazne vrstice oziroma kot vtičnik za Eclipse IDE, v kolikor nam je ljubši razvoj v integriranem razvojnem okolju.

3.1.6 Symbian OS

Operacijski sistem Symbian OS [20] je v nekaj letih kar nekajkrat zamenjal lastnika. Prvotni lastnik je podjetje Symbian Ltd, ki je potomec podjetja Psion, ustanovljenega že leta 1980. V letu 2008 je družbo prevzela Nokia in ustanovila novo neodvisno in neprofitno skupnost, imenovano Symbian Foundation, katere ustanovni člani so podjetja Nokia, Sony Ericsson, NTT DoCoMo, Motorola, Texas Instruments, Vodafone, LG Electronics, Samsung Electronics, STMicroelectronics in AT&T. Kmalu po ustanovitvi sta zaradi sprememb pri strategiji razvoja naprav odstopili podjetji LG in Motorola. Ustanovljeno združenje si je zadalo cilj, da postane sistem odprtokodni, kar se je zgodilo leta 2010. Leto pozneje je Nokia sklenila zaveznitvo z Microsoftom in s tem napovedala uporabo Windows Phone OS na svojih mobilnih napravah, zaposlene na projektu Symbian pa premestila v podjetje Accenture. Symbian OS je namenjen mobilnim napravam s procesorji ARM. Arhitektura sistema je namenjena mobilnim napravam, ki vsebujejo omejene systemske vire, a se od njih pričakuje visoka zanesljivost. Posebni programerski pristopi in ukazi, značilni samo za Symbian OS, omogočajo nizko porabo pomnilnika in redke odpovedi zaradi napak v po-

mnilniškem naslavljanju. Čeprav naprave s Symbian OS običajno uporabljajo pomnilnike Flash, platforma vsebuje posebne metode za upravljanje diskovnih virov. Okolje je usmerjeno k dogodkom, zato se lahko procesor izklaplja, ko ni v uporabi. S tem se zmanjša poraba energije. V tabeli 3.4 so prikazani različice Symbian OS, leto izdaje in na novo podprte tehnologije.

Različica	Leto izdaje	Novo podprte tehnologije
6.0	2001	Bluetooth ⁵ , IrDA in podpora zunanjemu pomnilniku, XHTML, MMS, Java MIDP 1.0, SMTP, IMAP4 in POP3
7.0	2003	Java MIDP 2.0, možnost več hkratnih omrežnih povezav
8.0	2004	OpenGL, razširitev pomnilnika SD, prepoznavanje prstnih odtisov
9.1	2005	HSDPA, vgrajen WiFi, Bluetooth 2.0, podpora sprejemanju IP-klicev preko WiFi
Symbian ^3	2007	SQL-podpora (SQLite), digitalna mobilna TV (DVB-H, USDB-T)
Symbian Anna	2011	IPSEC, SSL, NFC
Nokia Belle	2012	Integriran NFC, izboljššan uporabniški vmesnik

Tabela 3.4: Različice Symbian OS, leto izdaje in na novo podprte tehnologije [21].

Symbian OS omogoča razvoj v Symbian C++, OPL, Phyton, Visual Basic, Simkin, Perl, Java ME in PersonalJava. Razvojno okolje Qt (vsebuje Qt Quick, Qt Mobility in Qt WebKit), ki je del Qt SDK, podpira tako naprave Symbian in MeeGo/Meamo, pa tudi operacijske sisteme za namizne računalnike (Microsoft Windows, Apple Mac OS X in Ubuntu Linux). Qt omogoča razvoj, testiranje in razhroščevanje mobilnih aplikacij. Nudi podporo razvoju bogatih uporabniških vmesnikov z uporabo jezika QML, ki je podoben jeziku JavaScript, in komponentami Qt Quick Components. S paketom Qt WebKit lahko razvijemo aplikacijo s spletnimi tehnologijami in prevedemo v domorodno aplikacijo. Razvojni okolji za aplikacije Java ME sta NetBeans IDE in Eclipse IDE,

⁵tehnologija za brezžično komunikacijo na majhni razdalji med napravami

kamor namestimo vtičnik Nokia SDK za Javo. Podobno kakor pri namiznih računalnikih tečejo tovrstne aplikacije v navideznih strojih, zato so praviloma počasnejše. Symbian je ena redkih platform, ki podpira razvoj aplikacij Adobe Flash Lite. Za prenos aplikacije na mobilno napravo uporabimo produkt Nokia Suite.

3.1.7 Bada OS

Bada OS [22] je operacijski sistem, namenjen pametnim telefonom in tabličnim računalnikom podjetja Samsung Electronics. Ime, ki izhaja iz besede "bada" (v korejskem jeziku pomeni ocean oziroma obala), naj bi odražalo odprte možnosti platforme. OS je bil prvič predstavljen v februarju 2010 na Mobile World Congress 2010 v Barceloni. Moto razvijalcev je "Pametni telefon za vsakogar" (angl. Smartphone for Everyone). Namen Bada OS ni, da bi konkuriral obstoječim platformam, temveč je namenjen konvencionalnim strankam, ki bi jim ponudili cenovno ugodne pametne telefone. S tem naj bi odprli novi segment tržišča pametnih telefonov, ki naj bi se v prihodnosti močno razširil. Podpira večino funkcij, ki jih dandanes pričakujemo na mobilnih napravah: bogat uporabniški vmesnik, nadzor nad porabo virov, spletno integracijo in podporo tehnologiji Flash. Večdotičnost (angl. multitouch), vibriranje, senzor premikanja (angl. motion sensor) in prepoznavanje obrazov (angl. face recognition) je le nekaj bolj sofisticiranih orodij, ki so na voljo razvijalcem. Tehnologija OpenGL ES je namenjena igram s trirazsežno grafiko (3D).

Trenutna različica operacijskega sistema je 2.0, ki je bila javnosti predstavljena v letu 2011. Značilnosti zadnje izdaje so večopravnost (angl. multitasking), funkcija kopiraj/prilepi in deljenje datotek (angl. file sharing) preko omrežja WiFi.

Samsung ponuja razvojna orodja Bada SDK, v katerih je integrirano razvojno orodje Bada IDE, ki vsebuje vse potrebno za razvoj, testiranje, razhroščevanje in poganjanje v posnemovalniku. Domorodne aplikacije je mogoče razvijati v C++ in tehnologiji Adobe Flash. Spletne različice pa z uporabo HTML, JavaScript in CSS.

3.2 Orodja za razvoj mobilnih aplikacij, delujočih na več operacijskih sistemih

Razvoj mobilnih aplikacij za različne platforme v splošnem zahteva takšno število različnih razvojnih okolij, kolikor različnih platform želimo podpreti.

Dodatno omejitev pri takem pristopu predstavlja zahteva po poznavanju različnih programskih jezikov (Objective C, Java, C/C++ itd.). Ob teh omejitvah nastajajo projekti, ki se trudijo premostiti razliko med različnimi platformami in ponuditi enotno orodje, ki je sposobno ustvariti domorodno aplikacijo za več operacijskih sistemov. To pomeni, da je identično izvorno kodo mogoče prevesti za več različnih platform. V naslednjih točkah je predstavljenih nekaj primerov s kratkimi opisi [24].

3.2.1 PhoneGap

Je odprtokodna rešitev, temelječa na označevalnem jeziku HTML5. Aplikacija, razvita s spletnimi tehnologijami, je sestavljena iz dveh delov - domorodnega brskalnika in spletne aplikacije, ki ima dostop do sistemskih virov naprave. Dostop je mogoč do merilca pospeška, GPS-enote, stikov, kamere, datotečnega sistema, večpredstavnostnih vsebin in predvajalnikov, omrežne enote itd. Na spletu je poleg trgovine mobilnih aplikacij tudi tako imenovana storitev v oblaku⁶, ki razvijalcem omogoča prevajanje (angl. build) aplikacij v domorodno kodo. Med drugim na spletni strani najdemo še tehnična navodila za dostop do sistemskih virov mobilnih naprav preko domorodnih programskih vmesnikov (API). Podpira platforme iOS, BlackBerry, Android, Symbian, WebOS, Bada in Windows Phone 7.

3.2.2 Rho Mobile

Zagotavlja celovito paleto programja za upravljanje poslovnih mobilnih aplikacij in podatkov. Paket RhoMBUS vsebuje programske pakete Rhodes, RhoConnect, RhoHub in RhoGallery. Orodje Rhodes, podobno kakor PhoneGap, omogoča razvoj aplikacij s spletnimi tehnologijami, ki kodo HTML prevede (angl. compile) v pravo domorodno aplikacijo. Vsebuje programske vmesnike za dostop do sistemskih virov mobilnih naprav, kakršne so GPS enota, stiki, kamera, opozorila in koledar. RhoConnect skrbi za povezovanje (integracijo) z obstoječimi aplikativnimi rešitvami, razvitimi v programskem jeziku Objective-C ali katerem drugem. Temelji na programskem jeziku Ruby in je licenciran kot odprtokodna rešitev. Razvoj s paketom Rhomobile je mogoč za iOS, BlackBerry, Windows Mobile, Windows Phone 7 in Android [25].

⁶način uporabe aplikacij kot storitev preko omrežja (običajno preko svetovnega spleta)

3.2.3 appMobi

Razvoj aplikacij z uporabo appMobi XDK (angl. cross platform development kit) omogoča spekter razvojnih strategij, in sicer [26]:

- klasična spletna aplikacija; ne vsebuje grafičnega vmesnika naprave, deluje samo z vzpostavljeno internetno povezavo,
- mobilna spletna aplikacija; uporaba HTML5, dostop do nekaterih sistemskih virov mobilne naprave (GPS-enota, merilec pospeška),
- spletna aplikacija MobiUS; domorodni dostop do večine sistemskih virov mobilne naprave z uporabo programskega vmesnika appMobi API,
- hibridna domorodna aplikacija; zgrajena na osnovi HTML5 in JavaScript z uporabo storitve v oblaku appMobi.

Aplikacije razvijamo s spletnimi tehnologijami HTML5, JavaScript in CSS. Vsebuje podobne programske vmesnike kakor PhoneGap in je v neki smeri celo njegova nadgradnja. Ogrodje AppMobi JavaScript ima možnost zaviti (angl. wrap) aplikacije PhoneGap in na ta način zgladiti ostre robove razvoja mobilnih aplikacij. Orodja vsebujejo posnemovalnike različnih mobilnih naprav. Trenutno sta podprti mobilni platformi iOS in Android ter namizne različice v okoljih Windows, Linux in Mac.

3.2.4 appcelerator

Odpri tokodni projekt ponuja svoj paket Titanium Mobile SDK. Z orodjem Titanium Studio, osnovanim na Eclipse IDE, je mogoč razvoj domorodnih, hibridnih in mobilnih aplikacij za osebne računalnike in mobilne naprave. Razvoj aplikacij poteka z uporabo spletnih tehnologij, kakršne so JavaScript, HTML, CSS, Python, Ruby in PHP. Z razvojnim orodjem lahko gradimo mobilne aplikacije za operacijska sistema iOS in Android ter aplikacije na operacijskih sistemih Windows, Linux in Mac. V programskih rešitvah lahko uporabimo tudi njihovo storitev v oblaku, ki ponuja potisno obveščanje (angl. push notification), hrambo datotek, administracijo uporabnikov, urejanje in administracijo fotografij, socialna omrežja, klepetalnice, forume, sporočanje itd.[27].

3.2.5 DragonRAD

Komercialna rešitev ponuja orodja za razvoj, prevajanje, razhroščevanje in zaščito poslovnih mobilnih aplikacij. Rešitev zagotavlja enostavno načrtovanje,

namestitev (angl. deployment), vzdrževanje (angl. maintenance) in administracijo aplikacij s podporo raznovrstnim sistemom za upravljanje podatkovnih baz (IBM Lotus Domino, SQL Server, Oracle, MySQL, Sybase, PostgreSQL, & ZIM). Razvojno okolje omogoča razvoj brez kodiranja, zgolj z uporabo komponent, s katerimi gradimo aplikacijo na način primi in spusti (angl. drag & drop). V primeru, ko je treba vgraditi dodatno logiko orodje samo vsebuje lasten skriptni jezik. Po njihovem mnenju orodje pospeši razvoj mobilnih naprav, poenostavi izvedbo namestitev in vzdrževanje, omogoča povezljivost s podatkovnimi bazami in spletnimi storitvami ter omogoča enkraten razvoj in namestitev na več platform. V tem trenutku je mogoča namestitev na BlackBerry, Android in Windows Mobile. V prihodnosti pripravljajo podporo tudi platformi iOS [28].

3.2.6 Sencha Touch

Podjetje Sencha je plasiralo prvo ogrodje, namenjeno razvoju domorodnih aplikacij s spletnimi tehnologijami HTML5, CSS3 in JavaScript. Orodje je namenjeno razvoju visoko zmogljivih in prilagodljivih aplikacij s široko paleto možnosti optimizacije. Paket Sencha SDK je na voljo za operacijske sisteme Windows in Mac. Razvojno okolje spodbuja uporabo arhitekture model-pogled-krmilnik (angl. Model-View-Controller). Dodatno lahko uporabimo še razvojno tehniko AJAX (asinhroni JavaScript in XML), objekte DOM (angl. Document Object Model), avtomatsko zaznavanje prisotnosti sistemskih virov mobilnih naprav (GPS-enote, kamere, merilca pospeška itd.) ter podporo kretnjam, značilnih za ekrane, občutljive na dotik. Na voljo je veliko število ikon in primerov aplikacij, ki jih je mogoče uporabiti pri razvoju. Razvoj aplikacij je možen za operacijske sisteme iOS, Android, BlackBerry in Kidle Fire [29].

3.2.7 UNITY Mobile

UNITY Mobile je podjetje, ki ponuja okolje za razvoj mobilnih aplikacij, namenjenih malim podjetjem, ki potrebujejo učinkovite rešitve za opravljanje dejavnosti. Rešitev je na voljo na spletu v obliki storitve v oblaku. Produkt vsebuje čarovnike⁷ (angl. wizard) za začetnike, orodja za administracijo uporabnikov, okolje za urejanje vsebin, podporo mobilni prodaji in oglaševanju, podporo objavljanju in distribuciji ter analitična orodja. Podprta izvajalna okolja so iOS, Android, BlackBerry, Windows Phone [30].

⁷Grafični uporabniški vmesnik, ki korakoma vodi uporabnika po dobro zasnovanih zaslonskih maskah.

3.2.8 MoSync

MoSync SDK vsebuje orodja za razvoj domorodnih in hibridnih mobilnih aplikacij. Odprtokodna različica je namenjena razvoju nekomercialnih aplikacij. Razvoj poslovnih programov je licenčno mogoč samo s plačljivo različico. Podobno kakor druge rešitve razvoj poteka s spletnimi tehnologijami v jeziku JavaScript in HTML5. Razvojni programski vmesniki omogočajo dostop do sistemskih virov naprav, kakršne so kamera, razni senzorji, datotečni sistem in domorodni uporabniški vmesnik naprave. Vkolikor nam dostop do komponent naprave iz JavaScript predstavlja ozko grlo, MoSync podpira razvoj v programskem jeziku C/C++, ki je kompatibilen na več različnih platformah. Na tak način je mogoče razviti hibridno aplikacijo. Z uporabo njihove knjižnice dosežemo celo enostavno uporabo programov C/C++ dostopnih iz jezika JavaScript. Pri razvoju iger, kjer uporabljamo OpenGL, kodiranje ali procesiranje grafike, imamo možnost uporabiti vse standardne knjižnice C/C++. Nabor podprtih operacijskih sistemov je obširen: Android, BlackBerry, iOS, JavaME, Moblin, Symbian, Windows Mobile in Windows Phone [31].

3.2.9 Qt

Qt SDK združuje Qt-ogrodje z orodji, katerih namen je racionalizirati razvoj mobilnih aplikacij. Namenjen je razvoju (pametnih) telefonov z operacijskima sistemoma Symbian in Nokia N9. Razvoj v orodju Qt Creator IDE poteka v programskih jezikih C++ in JavaScript. Dodatno omogoča razvoj za osebne računalnike na platformah Windows, Mac OS X in Linux. Qt-ogrodje ponuja intuitivne programske vmesnike (API) za C++. Za hiter (angl. rapid) razvoj uporabniškega vmesnika uporabimo Qt Quick, s tehniko, podobno programiranju JavaScript/CSS. V tem času so podprte platforme Windows, Windows CE, Windows Mobile, Linux/X11, Embedded Linux, Mac OS X, Symbian in Nokia N9 [32].

3.2.10 TotalCross

Orodje za razvoj mobilnih aplikacij za dlančnike in pametne telefone. Uporablja programski jezik Java, kar omogoča široko prenosljivost programov. Predpogoj je nameščeno izvajalno okolje TotalCross, ki je naslednik platforme SuperWaba. Migracijo aplikacij v novo okolje zlahka izvedemo z orodji, ki so del novega SDK. Projekt je zaradi lastnega izvajalnega okolja najbližji konceptu "razviti enkrat, izvajati povsod" (angl. write once, run anywhere). TotalCross SDK ponuja močna orodja za upravljanje omrežnih povezav, urejanje

komponent z grafičnimi vmesniki, podporo aplikacijam brez grafičnega vmesnika, širok nabor grafov, podporo SSL in implementacijo najbolj uporabljenih algoritmov za kodiranje [33].

Poglavje 4

PhoneGap

Viharno tržišče mobilnih naprav in operacijskih sistemov nas sili v iskanje optimalnejše rešitve, kakor je zgolj nenehno prilagajanje razvojnih okolij. Izdelava množice aplikacij za vsako izmed platform zahteva poznavanje različnih programskih jezikov in orodij. Še več, programski jeziki, namenjeni razvoju domorodnih mobilnih aplikacij, se razlikujejo od jezikov, ki se danes uporabljajo pri razvoju spletnih aplikacij in portalov. Ponudniki mobilnih naprav in operacijskih sistemov, ki jih poganjajo, imajo vsak svojo vizijo prihodnosti. To ovira uveljavljanje standardov na področju razvoja mobilnih aplikacij. Skupnost PhoneGap [3] ponuja rešitev, ki ob uporabi spletnih tehnologij omogoča razvoj aplikacij, delujočih na več platformah. Ogrodje predstavlja velik korak k premostitvi na eni strani hitro spreminjajočega se mobilnega trga in na drugi veliko množico izkušenih razvijalcev z odličnim poznavanjem spletnih tehnologij. Znanje in izkušnje razvoja za svetovni splet, deloma ali v celoti, koristno uporabimo pri razvoju mobilnih aplikacij [1].

PhoneGap je odprtokodna rešitev, ki vključuje odprte standarde¹. V marcu leta 2012 je bil izveden prenos okrilje skupnosti Apache Software Foundation pod imenom Apache Cordova. Projekt dolgoročno ostaja odprt in neodvisen. Uporabijo ga lahko razvijalci in podjetja za razvoj mobilnih aplikacij, ki so proste plačila, plačljive, odprtokodne ali v katerikoli kombinaciji le-teh. Aplikacije PhoneGap so hibridne. Niso niti popolnoma domorodne, saj se grafični vmesnik prikazuje preko spletnega pogleda (in ne v domorodnem uporabniškem vmesniku), niti niso povsem spletne, ker jim je omogočen dostop do sistemskih virov naprave.

¹Odpri standard (angl. open standard) je standard, ki je javno dostopen in ima različne pravice za uporabo in tudi različne možnosti razvoja standarda.

4.1 Zgodovina

Ime PhoneGap se je prvič pojavilo leta 2009 na konferenci iPhoneDevCamp v San Franciscu. Prvotni avtor in eden izmed primarnih podpornikov projekta je bilo podjetje Nitobi. Ustanovilo je skupnost, v kateri so s svojimi prispevki sodelovala tudi velika podjetja, kakršna so IBM, RIM in Microsoft. Prizadevanja združenja so šla v smeri odprtosti, preglednosti in sodelovanja. Kmalu po predstavitvi aprila 2009 je bil projekt nagrajen na konferenci Web 2.0 Expo Launch Pad. Število aplikacij, razvitih na ta način, odtlej strmo narašča. Ogrodje (angl. framework) je podprlo podjetje Apple, ki ga kljub spremenjeni različici (4.0) njihove razvojne licenčne pogodbe še vedno podpira. V letu 2008 sta bila podprta Android in BlackBerry, v naslednjem letu pa še Windows Mobile, Nokia WRT in PalmOS. Podjetje Sony Ericsson je v tistem letu vključilo PhoneGap v svoj razvojni paket WebSDK. Prva različica z oznako 1.0.0rc1 je bila predstavljena 12. julija 2011 in še istega meseca so v Portlandu organizirali konferenco PhoneGap Day. Oktobra leta 2011 je podjetje Adobe prevzelo podjetje Nitobi. Novi lastnik si je prizadeval, da bi se ekipa osredotočila na projekt PhoneGap in nadaljevala delo z učinkovitim načrtovanjem in razvojem za večino mobilnih platform. Podjetje Adobe še naprej gosti tako spletno skupnost, kakor storitev v oblaku PhoneGap Build, ki je trenutno v beta različici. V marcu 2012 je bil projekt pridružen odprtokodni skupnosti Apache Software Foundation pod imenom Cordova. V tem času je bila koda s spletnega strežnika prenesena več kakor 600.000-krat. V spletnih trgovinah mobilnih aplikacij je na voljo nekaj tisoč aplikacij PhoneGap.

4.2 Podprti operacijski sistemi

Število operacijskih sistemov, ki jih podpira PhoneGap, se iz leta v leto širi. Na sliki 4.1 je prikazano sedem logotipov mobilnih operacijskih sistemov, ki jih podpira ogrodje PhoneGap: iOS, Android, BlackBerry, Windows Phone, WebOS, Symbian in Bada.



Slika 4.1: Logotipi mobilnih operacijskih sistemov podprtih v ogrodju PhoneGap [34].

4.3 Vzpostavitev razvojnih okolij

Ogrodje PhoneGap sledi načelu razvij enkrat, prevedi (angl. compile) večkrat. Enotno izvorno kodo prevedemo posebej za vsak mobilni operacijski sistem. Razvoj poenostavi kakovostno orodje, ki omogoča kodiranje, razhroščevanje in testiranje razvitih aplikacij. Vkolikor želimo sami pripraviti aplikacije, delujoče na vseh sedmih operacijskih sistemih, moramo vzpostaviti prevajalna okolja za vsakega izmed njih. Za razvoj in testiranje v ogrodju PhoneGap z njihove spletne strani prenesemo (angl. download) še paket, ki vključuje knjižnice in ustrezno dokumentacijo za vzpostavitev novega projekta (nove aplikacije). Skupnost nam ponuja storitev v oblaku "PhoneGap: Build", ki omogoča prevedbo izvorne kode v JavaScript, HTML in CSS v domorodno kodo. Storitve daje možnost kreiranja domorodne kode, namenjene razhroščevanju (angl. debug). Ob uporabi storitve v oblaku prevajalnega okolja ne potrebujemo. Lastno okolje vseeno olajša razvoj, razhroščevanje in testiranje v posnemovalniku mobilnih naprav. Za vtis o razvojnih okoljih bomo v nadaljevanju po mobilnih operacijskih sistemih predstavili potrebno programsko opremo za razvoj s PhoneGap.

4.3.1 iOS

Zahteve:

- osebni računalnik z nameščenim operacijskim sistemom Mac OS X Snow Leopard (10.6).

Namestitev aplikacije na mobilno napravo zahteva:

- eno izmed mobilnih naprav Apple (iPhone, iPad, iPod Touch),
- razvojno elektronsko potrdilo.

Potrebna orodja in knjižnice:

- razvojno orodje XCode (potrebno članstvo na spletni strani Apple Developer Portal).

4.3.2 Android

Zahteve:

- razvojno okolje Eclipse 3.4 ali višji

Potrebna orodja in knjižnice:

- orodje Eclipse Classic,
- orodja Android SDK,
- vtičnik ADT (angl. plugin).

4.3.3 BlackBerry**Zahteve:**

- osebni računalnik, ki ima nameščenega enega od operacijskih sistemov: Windows XP (32-bitni) ali Microsoft Windows 7 (32- in 64-bitni) ali Apple Mac OS X 10.6.4. ali višje različice.

Potrebna orodja in knjižnice:

- Sun JDK (32-bitna različica),
- Apache Ant,
- BlackBerry WebWorks Smartphone SDK za BlackBerry in/ali BlackBerry WebWorks Tablet OS SDK za Playbook.

4.3.4 Windows Phone**Zahteve**

- osebni računalnik, ki ima nameščenega enega od operacijskih sistemov Microsoft Windows 7 ali Microsoft Windows Vista s SP2.

Namestitev aplikacije na mobilno napravo zahteva:

- članstvo v spletni trgovini mobilnih aplikacij App Hub.

Potrebna orodja in knjižnice:

- Windows Phone SDK.

4.3.5 WebOS

Zahteve:

- osebni računalnik, ki ima nameščenega enega od operacijskih sistemov: Windows, Mac OS X ali Linux.

Potrebna orodja in knjižnice:

- Oracle Virtual Box,
- webOS SDK,
- cygwin (samo na okoljih Windows).

4.3.6 Symbian

Zahteve:

- osebni računalnik, ki ima nameščenega enega od operacijskih sistemov: Windows, Mac OS X ali Linux.

Potrebna orodja in knjižnice:

- cygwin (samo na okoljih Windows).

4.4 Dostopnost sistemskih virov mobilnih naprav

Pri razvoju mobilnih aplikacij je treba upoštevati, da se upravljanje mobilnih naprav deloma razlikuje od upravljanja osebnih in prenosnih računalnikov. Navadno nimajo tipkovnice, imajo pa zaslon, občutljiv na dotik. Operacijski sistemi različno podpirajo kretnje, značilne za zaslone, občutljive na dotik (glej tabelo 2.1). Pri razvoju se je zato priporočljivo držati predvsem tistih, ki so po platformah večinsko podprti. Pod besedno zvezo sistemski viri naprav imamo v mislih tiste komponente in možnosti, ki olajšujejo uporabo oziroma bogatijo funkcionalnost naprave. Primer komponente je enota GPS, ki omogoča globalno določanje lege in je njeno funkcionalnost mogoče uporabiti v aplikaciji. Podoben primer je dostop in uporaba stikov, kjer so shranjeni podatki o osebah, kakršni so ime, priimek, naslov, telefonska številka itd. V nadaljevanju so naštet sistemski viri, ki so podprti v PhoneGap:

- *Merilnik pospeška (angl. accelerometer)*; uporaba senzorja zaznavanja gibanja oziroma obračanja mobilne naprave.
- *Kamera (angl. camera)*; uporaba kamere omogoča fotografiranje oziroma snemanje video vsebin.
- *Kompas (angl. compass)*; uporaba kompasa omogoča določanje strani neba.
- *Stiki (angl. contacts)*; dostop do stikov oseb omogoča branje, pisanje in spreminjanje podatkov, kakršni so: imena, priimki, naslovi, telefonske številke, naslovi elektronske pošte itd.
- *Datotečni sistem (angl. file)*; dostop do datotečnega sistema naprave omogoča branje, pisanje in spreminjanje vsebine datotek.
- *Geografska lokacija (angl. geolocation)*; dostop do enote GPS omogoča določanje globalne geografske lege.
- *Večpredstavnost (angl. multimedia)*; uporaba omogoča predvajanje zvočnih in video vsebin.
- *Omrežja (angl. network)*; uporaba omogoča dostop do podatkov o omrežjih (WiFi, UMTS, GPRS itd.).
- *Obvestilo (opozorilo) (angl. notification(alert))*; uporaba omogoča prikazovanje grafičnega opozorila.
- *Obvestilo (zvok) (angl. notification(sound))*; uporaba omogoča sprožitev zvočnega opozorila.
- *Obvestilo (vibracija) (angl. notification(vibration))*; uporaba omogoča opozarjanje z vibriranjem.
- *Podatkovno skladišče (angl. storage)*; omogoča kreiranje in uporabo lokalne podatkovne baze na mobilni napravi.

Ogrodje PhoneGap ne podpira vseh sistemskih virov naprav na vseh operacijskih sistemih, kar je razvidno iz slike 4.2.

V nadaljevanju bomo po posameznem viru mobilne naprave na kratko spoznali, katere metode nam ponuja ogrodje PhoneGap, pri čemer se ne bomo spuščali v podrobnosti in na katerih operacijskih sistemih je vir mogoče uporabiti [4].

	 iOS iPhone / iPhone 3 G	 iOS iPhone 3GS in novejši	 Android	 OS 4.6 IN 4.7	 OS 5.x	 OS 6.0 +	 WebOS	 DP7	 Symbian	 Bada
MERILNIK POSPEŠKA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
KAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
KOMPAS	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓
STIKI	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
DATOTEČNI SISTEM	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗
GEOGRAFSKA LOKACIJA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VEČPREDSTAVNOST	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
OMREŽJA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OBVESTILO (OPOZORILO)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OBVESTILO (ZVOK)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OBVESTILO (VIBRACIJA)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PODATKOVNO SKLADIŠČE	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗

Slika 4.2: Dostopnost sistemskih virov naprav v ogrodju PhoneGap po mobilnih platformah [35].

4.4.1 Merilnik pospeška

Gibanje naprave v prostoru (v treh smereh) merimo z objektom `accelerometer`.

Metoda `accelerometer.getCurrentAcceleration`

Metoda vrne premik relativno na orientacijo naprave po vseh treh koordinatah x, y in z. Merilec pospeška zazna spremembo (razliko) med začetnim in končnim položajem naprave. Podprta je na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji) in iOS.

Metoda `accelerometer.watchAcceleration`

Metoda frekvenčno vrača spremembo položaja naprave v intervalu, ki ga posredujemo v milisekundah preko parametra `frequency`. Podprta je na platformah

Android, BlackBerry WebWorks (OS 5.0 ali višji) in iOS.

Metoda `accelerometer.clearWatch`

Metoda ustavi spremljanje gibanja naprave po referenčni enolični identifikaciji `watchID`. Podprta je na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji) in iOS.

4.4.2 Kamera

Današnjih mobilnih naprav (predvsem pametnih telefonov) si ne predstavljamo brez fotoaparata oziroma kamere. Objekt `camera` omogoča dostop do privzete programske opreme za uporabo kamere na napravi.

Metoda `camera.getPicture`

Metoda uporabi privzeto programsko opremo na napravi za posnetek fotografije oziroma prikaže fotografijo iz ustrezne mape (albuma) na mobilni napravi. Vrnjena vrednost je base64², kodiran tipa niz (angl. string) ali kot naslov (URI) slikovne datoteke. Kamero uporabimo za fotografiranje z vrednostjo lastnosti `Camera.sourceType = Camera.PictureSourceType.CAMERA` (privzeta vrednost). V primeru `Camera.sourceType = Camera.PictureSourceType.PHOTOLIBRARY` ali `Camera.PictureSourceType.SAVEDPHOTOALBUM` se prikaže pogovorno okno, na katerem je mogoče izbrati fotografijo za prikaz. Rezultat je lahko v kodirani obliki base64 ali URI, ki predstavlja lokacijo na napravi, kar izbiramo z ustrezno vrednostjo parametra `cameraOptions`. Metoda je podprta na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji), iOS in Windows Phone 7 (Mango).

4.4.3 Avdiovizualne vsebine

Objekt `capture` omogoča dostop do zvočnih, slikovnih in video vsebin na mobilni napravi.

Metoda `capture.captureAudio`

Metoda asinhrono kliče programsko opremo za snemanje zvočnih vsebin na mobilni napravi. Snemanje se konča bodisi z uporabnikovo prekinitvijo snemanja bodisi s številom posnetkov. To dosežemo s parametrom `limit` v objektu `CaptureAudioOptions`. Privzeta vrednost parametra je ena (1), kar pomeni, da

²Vrsta kodiranja, ki binarni zapis konvertira v bitni zapis ASCII 7

se operacija snemanja zaključi po prvem posnetku. Ob prenehanju snemanja se sproži `CaptureCB`, ki vrne polje³ (angl. `array`) objektov `MediaFile`. Metoda je podprta na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji), iOS in Windows Phone 7 (Mango).

Metoda `capture.captureImage`

Metoda je namenjena zajemanju fotografij. Asinhrona operacija preko programske opreme za upravljanje kamere zajame in vrne fotografijo. Omogoča enkratni ali zaporedni zajem več fotografij. Operacija se zaključi bodisi z izhodom iz aplikacije bodisi z doseženim številom fotografij, kar nam omogoča parameter `limit` kot lastnost objekta `CaptureImageOptions`. Privzeta vrednost parametra je ena (1), kar pomeni, da se po zajemu prve fotografije operacija zaključi. Po končanem snemanju se proži `CaptureCB`, ki vrne polje (angl. `array`) objektov `MediaFile`. Metoda je podprta na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji), iOS in Windows Phone 7 (Mango).

Metoda `capture.captureVideo`

Namen metode je v zajemanju video vsebin. Asinhrona operacija preko programa za upravljanje kamere začne snemanje video vsebin. Omogoča zajem več posnetkov v seji. Snemanje se konča bodisi s prekinitvijo bodisi z doseženim številom posnetkov. To dosežemo s parametrom `limit` kot lastnostjo objekta `CaptureVideoOptions`. Privzeta vrednost parametra je ena (1), kar pomeni, da se po zajemu enega posnetka operacija zaključi. Po končani operaciji se proži `CaptureCB`, ki vrne polje (angl. `array`) objektov `MediaFile`. Metoda je podprta na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji), iOS in Windows Phone 7 (Mango).

Metoda `MediaFile.getFormatData`

Metoda je namenjena zajemanju metapodatkov objekta `MediaFileData`. Podprta je na platformah Android, BlackBerry WebWorks (OS 5.0 ali višji), iOS in Windows Phone 7 (Mango).

4.4.4 Kompas

Objekt `compass` uporabimo za zaznavanje smeri oziroma usmeritve naprave.

³Urejena zbirka vrednosti atributa, katerih položaj je določen z enim ali več indeksi.

Metoda `compass.getCurrentHeading`

Zajame smer oziroma usmeritev naprave. Vrnjena vrednost je v stopinjah od 0 do 359,99. Podatki se vrnejo preko objekta `CompassHeading`. Podprta je na platformah Android, iPhone in Windows Phone 7 (Mango) pod pogojem, da ima naprava vgrajen kompas.

Metoda `compass.watchHeading`

V nekem časovnem intervalu frekvenčno meri smer oziroma usmeritev naprave. Vrednosti so v stopinjah od 0 do 359,99. Časovni interval podajamo preko parametra `frequency`, ki je lastnost objekta `compassOptions`. Povratna vrednost je `watchID`, ki predstavlja referenco na časovni interval in ga uporabljamo za prekinitev spremljanja usmeritve naprave v metodi `compass.clearWatch`. Metoda je podprta na platformah Android, iPhone in Windows Phone 7 (Mango) pod pogojem, da ima naprava vgrajen kompas.

Metoda `compass.clearWatch`

Metoda na osnovi parametra `watchID` ustavi zajemanje smeri oziroma usmeritve naprave. Podprta je na platformah Android, iPhone in Windows Phone 7 (Mango) pod pogojem, da ima naprava vgrajen kompas.

Metoda `compass.watchHeadingFilter`

Vrača smer oziroma usmeritev v stopinjah v trenutku, če se le-ta spremeni vsaj za neko določeno število stopinj. Izmerjena vrednost je v stopinjah med 0 in 359,99 stopinj. Najmanjše število stopinj, ki sprožijo operacijo, posredujemo preko parametra `filter`, ki je lastnost objekta `compassOptions`. Vrnjena vrednost je `watchID` kot referenca na opazovani interval, ki ga uporabljamo za prekinitev spremljanja usmeritve naprave v metodi `compass.clearWatchFilter`. Metoda je podprta na platformi iPhone.

Metoda `compass.clearWatchFilter`

Na osnovi parametra `watchID` ustavi zajemanje smeri oziroma usmeritve naprave, ki smo ga začeli z metodo `compass.clearWatchFilter`. Podprta je na platformi iPhone.

4.4.5 Omrežja

Z objektom `connection` pridobimo informacije o mobilnih in brezžičnih omrežjih. Dostopen je preko vmesnika `navigator.network`.

Lastnost `connection.type`

Omogoča kontrolo, ki pove, ali smo povezani v mobilno ali brezžično omrežje. Vrne status in tip vzpostavljene povezave. Lista vrednosti, ki jih lahko zavzame lastnost `connection.type`, so:

- `Connection.UNKNOWN` // *nepoznana*,
- `Connection.ETHERNET` // *ethernet omrežje*,
- `Connection.WIFI` // *brezžično omrežje*,
- `Connection.CELL_2G` // *mobilno omrežje 2. generacije*,
- `Connection.CELL_3G` // *mobilno omrežje 3. generacije*,
- `Connection.CELL_4G` // *mobilno omrežje 4. generacije*,
- `Connection.NONE` // *povezava ni vzpostavljena v nobeno od omrežij*.

Lastnost je mogoče uporabiti na platformah iOS, Android, BlackBerry WebWorks (OS 5.0 ali višji) in Windows Phone 7 (Mango).

4.4.6 Stiki

Dostop, urejanje in dodajanje stikov upravljamo preko objekta `contacts`.

Metoda `contacts.create`

Sinhrona metoda vrne nov objekt `Contact`, katerega lastnost so podatki o novem stiku, ki ga želimo dodati v podatkovno bazo stikov na mobilni napravi. Objekt ni obstojen (angl. *persistent*) in je za trajno hrambo potrebno izvesti še metodo `Contact.save`. Operacija je podprta na platformah iOS, Android in BlackBerry WebWorks (OS 5.0 ali višji).

Metoda `contacts.find`

Asinhrona metoda izvede poizvedbo po podatkovni bazi stikov na napravi in vrne polje (angl. array) objektov tipa `Contact`, katerega lastnosti so podatki o konkretni osebi. Pri uporabi metode je treba specificirati, po katerih lastnostih se bo poizvedba izvedla preko parametra `contactFields`. Vrnjeni objekti tipa `Contact` bodo vsebovali samo tiste lastnosti, ki so določene s parametrom `contactFields`. Vkolikor parameter ne vsebuje iskanih lastnosti, bodo vrnjeni objekti `Contact` vsebovali le lastnost `id`. Nasprotno pa vrednost parametra `[\"*\"]` vrne objekte tipa `Contact` z vsemi lastnostmi. Uporabimo lahko tudi lastnost `contactFindOptions.filter` kot iskalni pogoj poizvedbe. Podprta je na platformah iOS, Android in BlackBerry WebWorks (OS 5.0 ali višji).

4.4.7 Lastnosti naprave

Dostop do informacij o programski in strojni opremi naprave zagotavlja objekt `device`.

Lastnost `device.name`

Vsebuje vrednost modela naprave. Določen je med proizvodnjo naprave in se lahko razlikuje med različicami istega modela. Lastnost je omogočena na platformah Android, BlackBerry, BlackBerry WebWorks (OS 5.0 ali višji), iPhone in Windows Phone 7 (Mango).

Lastnost `device.phonegap`

Vrednost lastnosti je različica izvajalnega okolja PhoneGap na napravi. Lastnost je omogočena na platformah Android, BlackBerry, BlackBerry WebWorks (OS 5.0 ali višji), iPhone in Windows Phone 7 (Mango).

Lastnost `device.platform`

Vrednost lastnosti je enaka imenu operacijskega sistema na napravi. Lastnost je omogočena na platformah Android, BlackBerry, BlackBerry WebWorks (OS 5.0 ali višji), iPhone in Windows Phone 7 (Mango).

Lastnost `device.uuid`

Vsebuje vrednost univerzalnega enoličnega identifikatorja, ki ga določi proizvajalec naprave. Lastnost je omogočena na platformah Android, BlackBerry,

BlackBerry WebWorks (OS 5.0 ali višji), iPhone in Windows Phone 7 (Mango).

Lastnost `device.version`

Vrednost lastnosti je različica operacijskega sistema naprave. Lastnost je omogočena na platformah Android 2.1 ali višji, BlackBerry, BlackBerry WebWorks (OS 5.0 ali višji), iPhone in Windows Phone 7 (Mango).

4.4.8 Datotečni sistem

PhoneGap nudi objekte za kreiranje, branje in pisanje vsebine in imen datotek ter map na datotečnem sistemu. V ta namen so nam na voljo sledeči objekti:

- *DirectoryEntry* - predstavlja mapo na datotečnem sistemu (po specifikaciji W3C),
- *DirectoryReader* - vsebuje listo datotek in map znotraj mape,
- *File* - vsebuje attribute datoteke,
- *FileEntry* - predstavlja datoteko na datotečnem sistemu,
- *FileError* - uporablja se pri napakah v zvezi z metodami *File* API,
- *FileReader* - omogoča branje datoteke,
- *FileSystem* - predstavlja datotečni sistem,
- *FileTransfer* - omogoča prenos datoteke s ali na strežnik,
- *FileTransferError* - uporablja se v primeru napake prenosa s ali na strežnik,
- *FileUploadOptions* - uporablja se v kombinaciji z objektom *FileTransfer*, pri čemer specificira dodatne parametre za prenos na strežnik (angl. upload),
- *FileUploadResult* - v primeru uspešnega zaključka prenosa (*FileTransfer*),
- *FileWriter* - omogoča pisanje v datoteko,
- *Flags* - vsebuje parametre za objekt *DirectoryEntry* oziroma metode `getFile` in `getDirectory`,

- *LocalFileSystem* - dostop do korenske mape (angl. root) datotečnega sistema,
- *Metadata* - zagotavlja informacije o stanju datoteke ali mape.

Datotečni sistem in razredi so podprti na operacijskih sistemih Android, BlackBerry WebWorks (OS 5.0 ali višji), iOS in Windows Phone 7 (Mango).

4.4.9 Geografska lokacija

Dostop do GPS-enote ponuja objekt `geolocation`. Geografska širina in dolžina sta podatka, ki ju lahko uporabimo na osnovi koordinat GPS (angl. Global Positioning System). Približek lokacije lahko pridobimo v omrežju na osnovi IP-naslova, RFID, WiFi in Bluetooth MAC-naslova ali enoličnega identifikatorja (ID) celice GSM/CDMA. Nikakršnega zagotovila s strani knjižnic aplikacijskega programskega vmesnika (API) nimamo, da vrača prave vrednosti. API je osnovan na specifikaciji W3C.

Metoda `geolocation.getCurrentPosition`

Asinhrona metoda vrne trenutno lokacijo naprave v objektu tipa `Position`. Podprta je na platformah Android, BlackBerry (OS 4.6), BlackBerry WebWorks (OS 5.0 ali višji), iPhone, Windows Phone 7 (Mango).

Metoda `geolocation.watchPosition`

Asinhrona metoda začne spremljanje geografske lokacije. Vrnjeno vrednost `watchID` uporabimo v metodi `geolocation.clearWatch` za zaustavitev spremljanja geografske lege. Ob spremembi lokacije v objektu tipa `Position` vrne novo vrednost. Podprta je na platformah Android, BlackBerry (OS 4.6), BlackBerry WebWorks (OS 5.0 ali višji), iPhone, Windows Phone 7 (Mango).

Metoda `geolocation.clearWatch`

Ustavi spremljanje geografske lege na osnovi vrednosti `watchID`. Podprta je na platformah Android, BlackBerry (OS 4.6), BlackBerry WebWorks (OS 5.0 ali višji), iPhone, Windows Phone 7 (Mango).

4.4.10 Večpredstavnost

Objekt `Media` uporabimo za zapis oziroma predvajanje zvočnih datotek na napravi. Trenutna implementacija večpredstavnosti ni po specifikaciji W3C, vendar bo v bližnji prihodnosti. Na voljo so naslednje metode:

- `media.getCurrentPosition` - vrne trenutno lokacijo predvajane zvočne datoteke,
- `media.getDuration` - vrne dolžino zvočne datoteke v sekundah,
- `media.play` - začne oziroma nadaljuje s predvajanjem zvočnega zapisa,
- `media.pause` - začasno ustavi predvajanje; premor,
- `media.release` - sprosti sistemske vire, uporabljene pri predvajanju posnetka,
- `media.seekTo` - premik na določeno lokacijo (v milisekundah) zvočne datoteke,
- `media.startRecord` - sproži snemanje zvočnega posnetka,
- `media.stopRecord` - zaključi snemanje zvočnega posnetka,
- `media.stop` - zaustavi predvajanje zvočnega posnetka.

Metode so podprte na platformah Android, iOS in Windows 7 (Mango).

4.4.11 Opozorila

Grafična, zvočna in opozorila z vibriranjem omogoča objekt tipa `notification`.

Metoda `notification.alert`

Prikaže okno z opozorilom oziroma izbiro nadaljnje aktivnosti. Prikazano okno je na večini platform sistemsko, na drugih je prikazano kot JavaScript opozorilo `alert` spletnega brskalnika.

Metoda `notification.confirm`

Prikaže okno za potrditev aktivnosti, ko želimo, da uporabnik ponovno potrdi, da želi izvesti neko akcijo. Privzeta gumba sta "Potrdi" in "Prekini".

Metoda `notification.beep`

Metoda izvede zvočni signal (pisk). Vhodni parameter določa, kolikokrat naj se pisk ponovi.

Metoda `notification.vibrate`

Sproži tresenje naprave, konkretnije proženje enote, ki omogoča tresenje. Vhodni parameter določa časovno dolžino tresenja v milisekundah.

Metode so podprte na operacijskih sistemih Android, BlackBerry (OS 4.6), BlackBerry WebWorks (OS 5.0 ali višji), iPhone, Windows Phone 7 (Mango).

4.4.12 Podatkovno skladišče

Dostop in upravljanje lokalne hrambe podatkov omogoča programski vmesnik, osnovan na specifikacijah W3C Web SQL *Database in W3C Web *Storage. Nekatere naprave (in pripadajoči operacijski sistemi) že podpirajo specifikaciji, zato ogrodje PhoneGap uporabi privzeto implementacijo (in ne lastne). Na napravah brez podpore lokalni hrambi je treba zagotoviti implementacijo PhoneGap , kompatibilno z W3C-specifikacijo.

Metoda `openDatabase`

Z metodo `window.openDatabase` kreiramo novo podatkovno skladišče SQL Lite. Vrnjena vrednost je objekt tipa `Database`, s katerim imamo možnost upravljanja podatkov. Podprta je na platformah Android, BlackBerry WebWorks (OS 6.0 ali višji) in iPhone.

4.5 Življenjski cikel aplikacije PhoneGap

Aplikacijo PhoneGap sestavljata dva dela: izvajalno okolje PhoneGap, ki je v osnovi spletni brskalnik, in uporabniška aplikacija, razvita s spletnimi tehnologijami. Zagon aplikacije povzroči zagon ogrodja in šele nato aplikacije, ki teče znotraj izvajalnega stroja. V življenjskem ciklu ponuja naslednje metode [2]:

- *deviceready* - eden najpomembnejših dogodkov, ki se sproži, ko je domorodni del aplikacije PhoneGap v celoti naložen,
- *pause* - sproži se, ko domorodna aplikacija PhoneGap aktivno aplikacijo preda v izvajanje v ozadju,

- *resume* - sproži se, ko domorodni del aplikacije PhoneGap aktivira aplikacijo, ki se je izvajala v ozadju,
- *online* - sproži se ob spremembi stanja povezave v medmrežje,
- *offline* - sproži se podobno kakor *online* ob spremembi stanja povezave v medmrežje,
- *backbutton* - uporabimo, če želimo spremeniti funkcionalnost ob pritisku na gumb “Nazaj” (angl. back),
- *batterycritical* - sproži se, ko aplikacija PhoneGap zazna kritični nivo napolnjenosti baterije (odvisno od naprave),
- *batterylow* - sproži se, ko PhoneGap aplikacija zazna nizek nivo napolnjenosti baterije (odvisno od naprave),
- *batterystatus* - sproži se ob spremembi napolnjenosti baterije vsaj za 1 odstotek,
- *menubutton* - uporabimo, vkolikor želimo spremeniti funkcionalnost ob pritisku na gumb “Meni” (angl. menu),
- *searchbutton* - uporabimo, če želimo spremeniti funkcionalnost ob pritisku na gumb “Išči” (angl. search),
- *startcallbutton* - uporabimo, vkolikor želimo spremeniti funkcionalnost ob pritisku na gumb “Klič” (angl. call),
- *endcallbutton* - uporabimo, vkolikor želimo spremeniti funkcionalnost ob pritisku na gumb “Končaj klic” (angl. end call),
- *volumedownbutton* - uporabimo, vkolikor želimo spremeniti funkcionalnost ob pritisku na gumb “Zmanjšaj jakost zvoka” (angl. volume down),
- *volumeupbutton* - uporabimo, vkolikor želimo spremeniti funkcionalnost ob pritisku na gumb “Povečaj jakost zvoka” (angl. volume up).

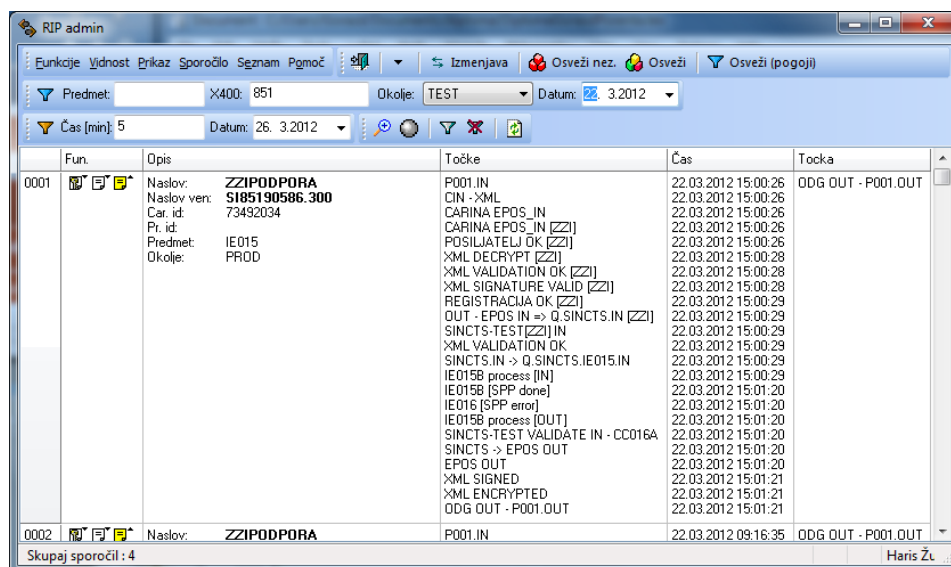
Na operacijskem sistemu Android so podprti vsi dogodki razen *startcallbutton*, *endcallbutton*, *volumedownbutton* in *volumeupbutton*. Na platformi iOS so podprti vsi dogodki, razen tistih, ki se nanašajo na gumbe, kar je posledica specifične naprave iOS, ki navadno nimajo tipk. Na BlackBerry WebWorks (OS 5.0 ali višji) so podprti vsi dogodki razen *searchbutton*. Drugi operacijski sistemi na tem področju nimajo podpore.

Poglavje 5

Razvoj aplikacije z ogrodjem PhoneGap

5.1 Aplikacija RipAdmin

V podjetju ZZI, d. o. o., razvijamo aplikacije za znanega naročnika. Ena izmed razvitih rešitev je tudi elektronska izmenjava. Gre za kombinacijo asinhrono hrbtenice, osnovane na sistemu čakalnih vrst, in sinhrono obdelave sporočil.



Slika 5.1: Aplikacija RipAdmin v okolju Microsoft Windows 7.

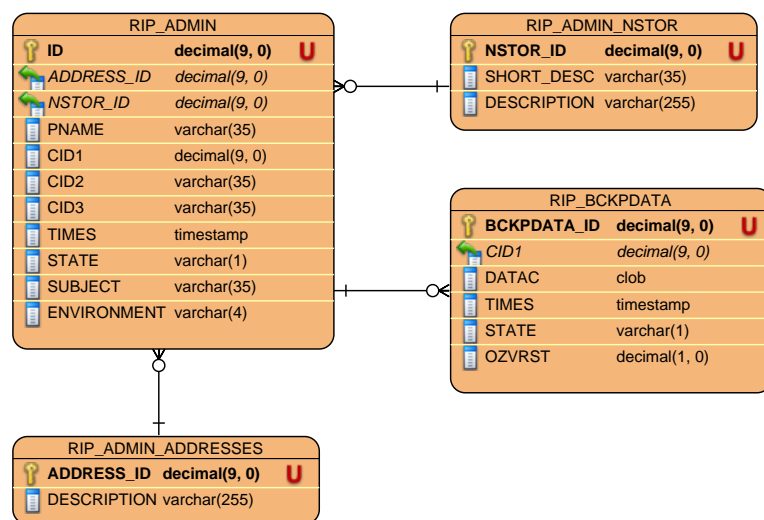
Za nadzorovanje sistema smo razvili aplikacijo RipAdmin (administracija računalniške

Metode, ki implementirajo primere uporabe, so naslednje:

- allPointsX400 (Procesi, ki ustrezajo iskalnim pogojem),
- allPointsBTW (Nezaključeni procesi),
- allPoints (Procesi, izvajani v zadnjih N sekundah),
- endPoint (Ročno zaključevanje procesa),
- getDok (Zajem sporočila),
- login (Avtorizacija),
- settings (Vnos nastavitvev).

5.4 Podatkovna baza

Vse aktivnosti v elektronski izmenjavi zapisujejo kontrolne točke izvajanega procesa v sistem čakalnih vrst. Procesiranje podatkov iz čakalnih vrst se izvaja na aplikacijskem strežniku, kjer se zapišejo v ustrezne tabele podatkovne baze. Podrobni podatkovni model je predstavljen na sliki 5.4.



Slika 5.4: Podrobni podatkovni model aplikacije RipAdmin.

5.5 Uporabniki aplikacije

Uporabniki se ob uporabi spletne storitve prijavijo z imenom in geslom, ki sta shranjena v podatkovni bazi. Glede na svoja pooblastila imajo možnost spremljanja delovanja določenega dela elektronske izmenjave. Administracija uporabniških imen, gesel in pooblastil se uporablja redko, zato uporabniški vmesnik za tovrstne aktivnosti ni razvit. Podatke o uporabnikih neposredno v podatkovni bazi vzdržuje skrbnik podatkov v njej.

5.6 Razvoj mobilne aplikacije RipAdmin

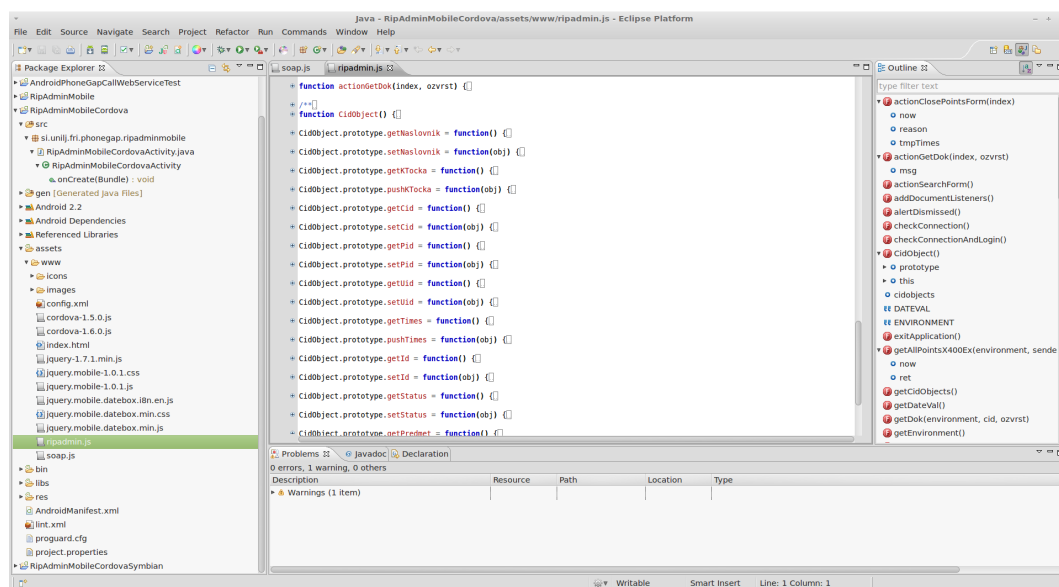
Zagotavljanje neprestanega delovanja elektronske izmenjave in nadzor od dežurne osebe zahtevata, da ima v vsakem trenutku dostop do pregleda stanja sistema. Doslej je veljalo, da ima v vsakem trenutku dostop do računalnika z vzpostavljeno povezavo s svetovnim medmrežjem. To je pomenilo, da je moral biti dežurni v neposredni bližini doma. S tehničnim napredkom pametnih telefonov (in mobilnih naprav) z zasloni znatnih ločljivosti se je ideja o razvoju mobilne različice aplikacije ponujala sama od sebe. Zaslone večjih ločljivosti, ki jih vsebujejo naprave, ponujajo okolje, primerno tudi bolj zahtevnim aplikativnim rešitvam. Zaposleni si model mobilne naprave izbiramo sami, tako da je stanje na področju raznovrstnosti mobilnih operacijskih sistemov podobno kakor v svetovnem merilu. Razvoj raznovrstnih aplikacij, ki bi tekle na posameznih platformah, bi poleg vzpostavljanja različnih okolij zahteval tudi učenje novih programskih jezikov.

Razvoj programske opreme se je v zadnjih letih usmeril v rešitve, dostopne preko spleta. Izvajalno okolje sodobnih aplikacij je aplikativni strežnik in portal. Razvijalci so v zadnjih letih energijo vlagali v znanje o spletnih tehnologijah, zato jih v podjetniškem okolju ni težko najti. Razvojni oddelki informacijskih podjetij vlagajo znatna sredstva v kadre s področja programskega jezika Java, skriptnega jezika JavaScript, označevalnega jezika za oblikovanje spletnih strani (HTML) in oblikovanja spletnih strani s stilskimi predlogami (CSS). Vse to je bil povod, da smo se odločili za razvoj mobilne aplikacije s paketom PhoneGap.

5.6.1 Razvojno okolje

Vzpostavili smo okolje za mobilni operacijski sistem Android različice 2.3.6. Za mobilno platformo je bilo treba namestiti razvojno okolje Eclipse IDE 3.4 (slika 5.5), Android SDK, ADT-vtičnik (angl. plugin) in knjižnice PhoneGap.

Namestitev smo opravili na operacijskem sistemu Mint Linux 12 (64-bitna različica).



Slika 5.5: Razvojno orodje Eclipse IDE 3.7.0.

5.6.2 jQuery Mobile

Pri razvoju aplikacije smo uporabili odprtokodno JavaScript knjižnico jQuery Mobile. Paket je mobilna različica projekta jQuery in vsebuje funkcije, ki poenostavijo razvoj spletnih strani in vključujejo izvajanje kode JavaScript na strani odjemalca. Na tak način smo poenostavili razvoj uporabniškega vmesnika. Knjižnica vsebuje razna orodja in funkcije za upravljanje aplikacije na zaslonu, občutljivem na dotik.

5.6.3 Uporaba spletnih storitev

V prvi fazi razvoja smo razvili programsko kodo v objektnem skriptnem jeziku JavaScript za dostop do obstoječih spletnih storitev (angl. web services). Opis metod spletne storitve je podan v WSDL-datoteki in vsebuje specifikacije vhodnih parametrov in tipa podatkov, ki jih vrne. Uporabili smo objekt, ki vsebuje attribute, potrebne za komunikacijo, in je predstavljen v primeru kode 5.1:

Primer kode 5.1: Razred SoapWebService v skriptnem jeziku JavaScript za klice spletnih storitev.

```
function SoapWebService(url){
    this.url = url;
    this.result = "";
    this.httpResponseHeader = "";
    this.methodName = "";
    this.cookie = "";
    this.xmlDoc = null;
}
```

Razred med drugim vsebuje metodo `SoapWebService.prototype.call = function (methodName, async, methodParams)`, preko katere pošljemo SOAP-klic do aplikacijskega strežnika. Parameter `methodName` vsebuje ime SOAP-metode, parameter `async` pove, ali gre za asinhron ali sinhron klic, in parameter tipa polje (angl. array) `methodParams` vsebuje seznam parov imen in vrednosti parametrov. Komunikacija se začne s prijavo na spletno storitev z metodo `ripAdminLogin`, katere definicija je v primeru kode 5.2.

Primer kode 5.2: Metoda `ripAdminLogin`.

```
function ripAdminLogin( username, password) {
    wws = new SpletnaStoritevSOAP(getUrl());
    wws.call("login", false,
            new Array("userName", username, "pass", password));
}
```

Pri klicu metode `ripAdminLogin` iz HTTP-glave preberemo in shranimo HTTP-piskotek (angl. HTTP Cookie) v objekt `SOAPWebService`. Ohranja nam sejo na aplikacijskem strežniku. V nadaljevanju so predstavljene še druge metode spletnega strežnika.

unhandledPoints

Metoda nam vrne nezaključene procese elektronske izmenjave. Bodisi so sporočila še vedno v neki točki elektronske izmenjave bodisi je prišlo v sistemu do tehnične težave in je sporočilo končalo v ustreznem obnovitvenem procesu. V prvem primeru po navadi ni potreben nikakršen poseg, je pa informacija zanimiva, ker nam pove, kje v sistemu prihaja do zastojev. V slednjem primeru je težava na sistemskem nivoju (primer: nedostopna podatkovna baza) oziroma je sporočilo strukturno tako neustrezno, da iz vsebine ni mogoče identificirati

pošiljatelja. Tovrstne napake sporočamo podpori uporabnikom. Primer kode 5.3 nam prikazuje vsebino metode.

Primer kode 5.3: Metoda ripAdminUnhandledPoints.

```
function ripAdminUnhandledPoints() {
  wws.call("unhandledPoints", false,
    new Array("compressed", "uncompressed"));
  var error = wws.getResponseException();
  if (error != null && error != '')
    throw error;
  return wws.getResponseValue();
}
```

ripAdminAllPointsX400Ex

Metoda ripAdminAllPointsX400Ex vrne seznam procesov, ki ustrezajo iskalnemu pogoju. Primer kode 5.4 prikazuje vsebino metode.

Primer kode 5.4: Metoda ripAdminAllPointsX400Ex.

```
function ripAdminAllPointsX400Ex(environment, sender, subject,
  dateVal) {
  wws.call("allPointsX400Ex", false,
    new Array("okolje", environment,
      "x400", sender,
      "predmet", subject,
      "day", dateVal,
      "compressed", "uncompressed"));
  var error = wws.getResponseException();
  if (error != null && error != '')
    throw error;
  return wws.getResponseValue();
}
```

Vhodni parametri, ki določajo prikazan seznam, so:

- `environment`; določa okolje procesiranja sporočila: razvojno okolje (Development), uvajalno okolje (Test) in produkcijsko okolje (Production),
- `sender`; določa naslov pošiljatelja,
- `subject`; določa tip sporočila,
- `dateVal`; določa datum procesiranja sporočila,

- `compressed`; določa ali naj bo rezultat kodiran v BASE64¹.

Seznam procesiranja sporočil nam omogoča spremljanje procesiranja sporočil konkretnega pošiljatelja ali tipa. V njem lahko razberemo čas, ko je bilo sporočilo v obdelavi.

ripAdminEndPoint

Nadzorni sistem v primeru tehnične ali strukturne napake sporočila preusmerja v ustrezne čakalne vrste. Nadaljevanje procesa opravi administrator sistema in ne nadzornik. Vseh procesov se ne da nadaljevati, saj je mogoče, da je sporočilo preveč popačeno. Nadzorna aplikacija mora dovoljevati ročno zaključevanje procesa. Prav to je funkcionalnost metode `ripAdminEndPoint`, katere vsebina je prikazana na primeru kode 5.5. Parametra sta enolični identifikator procesa `cid` in kratek opis napake v dolžini do 35 znakov `comment`.

Primer kode 5.5: Metoda `ripAdminEndPoint`.

```
function ripAdminEndPoint(cid, comment) {
    wws.call("endPoint", false,
            new Array("cid", cid, "comment", comment));
}
```

ripAdminGetDok

Vsebino sporočila pridobimo z metodo `ripAdminGetDok`. Vhodni parametri so `environment` (pove, katero okolje nas zanima - razvoj, test ali produkcija), `cid` (enolični identifikator procesa) in `ozvrst` (pove, ali želimo vhodno ali izhodno sporočilo: vrednost "2" ali vrednost "1"). Vsebina je prikazana na primeru kode 5.6.

Primer kode 5.6: Metoda `ripAdminGetDok`.

```
function ripAdminGetDok(ozvrst, cid, ozvrst) {
    wws.call("getDok", false,
            new Array("environment", ozvrst, "cid",
                    ozvrst, "ozvrst",));
}
```

¹Vrsta kodiranja, ki binarni zapis konvertira v bitni zapis ASCII 7.

5.6.4 Razvoj programske logike in uporabniškega vmesnika

Programska koda mora na eni strani zagotoviti dostop do spletnih storitev in njihovo uporabo, na drugi pa kontrolo prehodov med stranmi. S prvo smo se deloma srečali že v prejšnjem poglavju. Spletne storitve so bile razvite kmalu po pojavu protokola SOAP, ki še ni omogočal izmenjave objektov, zato se podatki med strežnikom in odjemalcem prenašajo kot niz znakov (angl. string). Razvili smo metodo, ki razčleni (angl. parse) vrnjeno vrednost spletne storitve in preslika vrednosti v polje (angl. array) objektov tipa `CProcess`, ki je predstavljen na sliki 5.6. Objekti se po klicanju spletnih storitev hranijo v pomnilniku naprave, a zaradi majhnega števila ne predstavljajo problemom pri kapacitetah današnjih mobilnih naprav.

CProcess
-id : long
-cid : long
-pid : string
-uid : string
-naslovník : string
-predmet : string
-okolje : string
-nstor : int
-status : string
-ktocka : string []
-times : date[]
+getCid()
+setCid(cid) : void
+getPid()
+setPid(pid) : void
+getUid()
+setUid(uid) : void
+getNaslovník()
+setNaslovník(naslovník) : void
+getId()
+setId(id) : void
+getStatus()
+setStatus(status) : void
+getPredmet()
+setPredmet(predmet) : void
+getOkolje()
+setOkolje(okolje) : void
+getNstor()
+setNstor(nstor) : void
+getKtocka()
+setKtocka(ktocka) : void
+getTimer()
+setTimer(timer) : void

Slika 5.6: Razred `CProcess` vsebuje attribute procesa.

Uporabniški vmesnik je zasnovan na osnovi knjižnic (JavaScript) in pripadajočih stilnih predlog spletnih strani (CSS) odprtokodnega projekta jQueryMobile. Čeprav tako jQueryMobile kakor PhoneGap omogočata prehod med več HTML-spletnimi stranmi, smo zaradi poenostavitve uporabili samo datoteko `index.html`. Knjižnice jQueryMobile omogočajo celo več spletnih strani znotraj ene HTML-datoteke, vendar se je pri testiranju na mobilnih napravah s starejšimi operacijskimi sistemi pokazalo, da taka rešitev v našem primeru ni primerna. Programska logika, ki nadzoruje prikazovanje strani, dinamično polni vsebino v razdelek `div data-role='content'`, prikazan na primeru kode 5.7.

Primer kode 5.7: Telo spletne strani `index.html`.

```
<body>
  <!-- Stran -->
  <div data-role="page">
    <!-- Glava -->
    <div data-role="header" id="header1" data-position="fixed">
      <span class="ui-btn-left">
      </span>
      <h1>Rip Admin Mobile</h1>
      <nav data-role='navbar'>
        <ul>
          <li><a data-iconpos="notext" data-icon="home" href="
            javascript:renderMain()" data-role="button"></a></li>
          <li><a data-iconpos="notext" data-icon="search" href="
            javascript:renderSearchForm()" data-role="button"
            ></a></li>
          <li><a data-iconpos="notext" data-icon="gear" href="
            javascript:renderLogin()" data-role="button"></a></li>
          <li><a data-iconpos="notext" data-icon="delete" href="
            javascript:exitApplication()" data-role="button">
            </a></li>
        </ul>
      </nav>
    </div><!-- konec - Glava -->

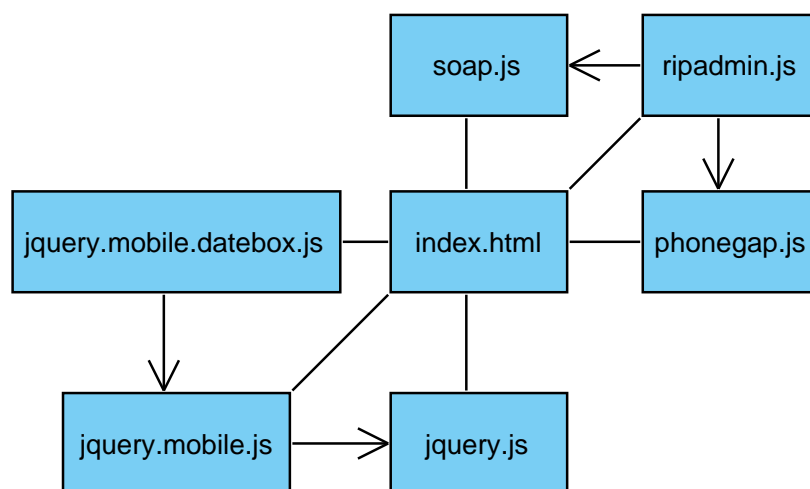
    <!-- Vsebina -->
    <div data-role="content" id="vsebina">
      <center><p>Please Wait....</p></center>
    </div><!-- konec - Vsebina -->

    <!-- Noga -->
    <div data-role="footer" id="footer1" data-position="fixed">
```

```
<h1>PhoneGap</h1>
</div><!-- konec - Noga -->

</div><!-- konec - Stran -->
</body>
```

Tak način onemogoča anomalije prikazovanja spletne strani na napravah, na katerih ogrodje PhoneGap ni ali pa je le deloma podprto, je pa nanje vseeno mogoča namestitvev aplikacije. Vzroku popačenj uporabniškega vmesnika se namreč ni mogoče izogniti, saj na nekaterih operacijskih sistemih PhoneGap ni mogoč dostop do informacij o sami napravi in platformi. Arhitektura aplikacije sledi modelu Model-Pogled-Kontroler (angl. Model-View-Controller), kjer model predstavlja programsko logiko aplikacije v JavaScript datotekah `ripadmin.js` in `soap.js`, pogled predstavlja vsebina telesa (značka `<body>`) HTML-datoteke `index.html` in stilna predloga `jquery.mobile.css` ter kontroler skriptni del HTML-datoteke `index.html`. Shematski prikaz vključenosti in odvisnosti HTML-datoteke in skriptnih datotek je prikazan na sliki 5.7.



Slika 5.7: Shema vključenosti in odvisnosti datotek HTML in JavaScript.

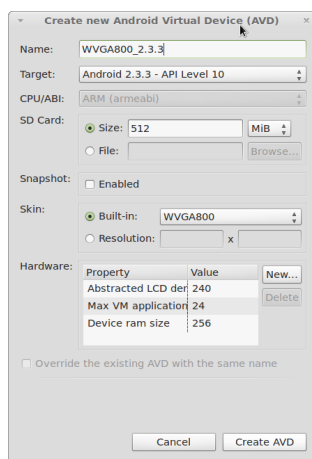
5.6.5 Prevajanje in testiranje aplikacije

Prevajanje (angl. compile) za operacijski sistem Android smo izvajali v razvojnem orodju Eclipse IDE. Testiranje in razhroščevanje (angl. debug) za isto platformo je potekala v posnemovalniku, ki je bil nameščen kot vtičnik (angl. plugin) v orodje Eclipse IDE (slika 5.8). Konfiguracijo posnemovalnika



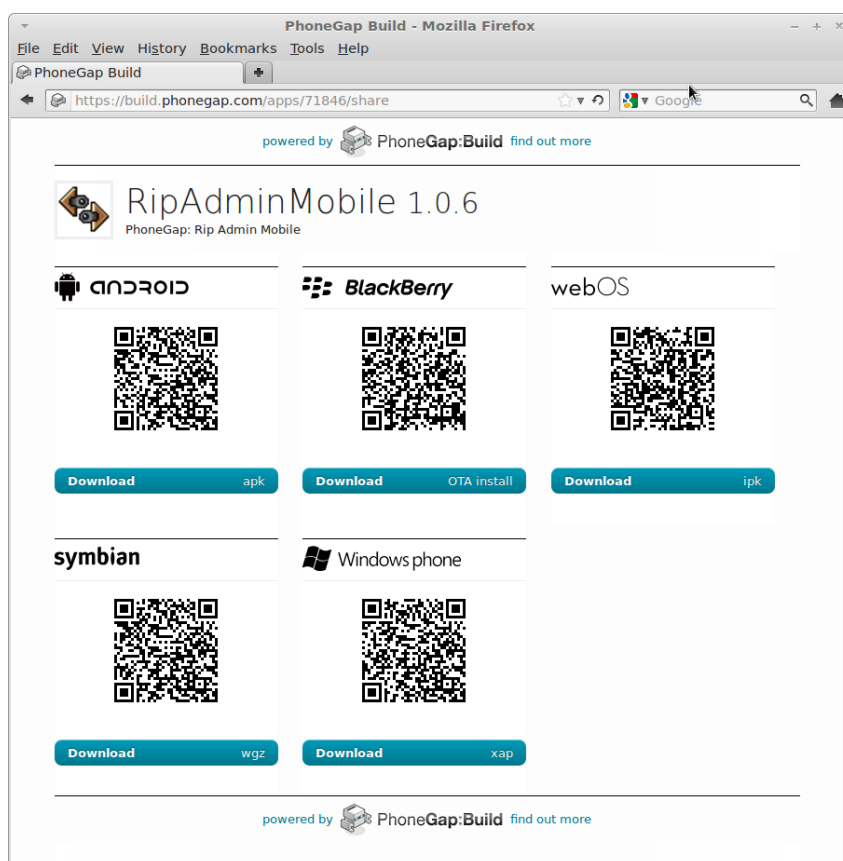
Slika 5.8: Posnemovalnik mobilne naprave z operacijskim sistemom Android.

opravimo v orodju Eclipse IDE. Preko upravitelja navideznih naprav Android (angl. Android Virtual Device Manager) izberemo velikost zaslona, različico Android SDK in spominski prostor SD-kartice (slika 5.9).



Slika 5.9: Konfiguracija posnemovalnika Android SDK.

Za druge mobilne operacijske sisteme smo uporabili storitev v oblaku PhoneGap:Build, ki ponuja prevajanje izvorne kode aplikacije v domorodno kodo za operacijske sisteme Android, BlackBerry, webOS, Symbian in Windows Phone. Storitev omogoča ustvarjanje novih aplikacij in posodabljanje starih. Lahko so javnega značaja, kar pomeni, da so dostopne vsem, ki obišejo spletno stran, ali pa zasebne. Brezplačna različica storitve dovoljuje eno zasebno aplikacijo na uporabnika in neomejeno število javnih. Na spletni strani je mogoče ustvariti skupino uporabnikov (razvijalcev), ki jim omogočimo sodelovanje pri razvoju aplikacije. Naša aplikacija je javno dostopna na spletni strani, ki je predstavljena na sliki 5.10.



Slika 5.10: Javno dostopna aplikacija RipAdmin mobile na spletni strani PhoneGap:Build.

Storitev omogoča tudi spreminjanje parametrov aplikacije, kot so različica aplikacije, različica uporabljenih knjižnic PhoneGap (trenutna različica 1.5),

ikono aplikacije, minimalno različico mobilnega operacijskega sistema, na katerem aplikacija deluje, uporabljene komponente mobilnih naprav in podobno. Enak učinek ima uporaba datoteke `config.xml`, prikazane na primeru kode 5.8.

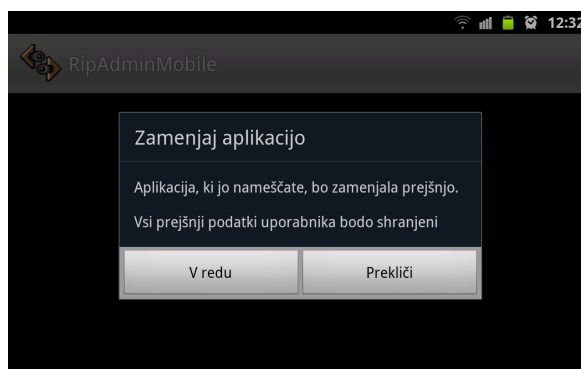
Primer kode 5.8: Vsebina datoteke `config.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns      = "http://www.w3.org/ns/widgets"
  xmlns:gap      = "http://phonegap.com/ns/1.0"
  id              = "si.unilj.fri.gp.phonegap.ripadminmobile"
  versionCode    = "8"
  version        = "1.0.6">
  <name>RipAdminMobile</name>
  <description>PhoneGap: Rip Admin Mobile</description>
  <author email="gorazd.porenta@siol.net">Gorazd Porenta</
    author>
  <gap:platforms>
    <gap:platform name="android" minVersion="3.0" />
    <gap:platform name="webos" />
    <gap:platform name="symbian.wrt" />
    <gap:platform name="blackberry" project="widgets"/>
  </gap:platforms>
  <!-- if a single icon is specified then Build will use it for
    all platforms -->
  <icon src="icons/icon.png" />
  <!-- the default attribute can also be used -->
  <icon src="icons/icon.png" default="true" />
  <!-- specifies the icon to be displayed when hovered over (on
    BlackBerry) -->
  <icon src="icons/icon-hover.png" hover="true" />
  <!-- iOS requires specific dimensions for icons; classic
    iPhone screens, Retina
    displays, and iPad screens respectively -->
  <icon src="icons/iphone-57.png" width="57" height="57"/>
  <icon src="icons/iphone-72.png" width="72" height="72"/>
  <icon src="icons/iphone-114.png" width="114" height="114"/>
  <!-- Android requires specific dimensions; ldpi, mdpi, and
    hdpi respectively -->
  <icon src="icons/android-36.png" width="36" height="36"/>
  <icon src="icons/android-48.png" width="48" height="48"/>
  <icon src="icons/android-72.png" width="72" height="72"/>
  <preference name="fullscreen" value="true" />
  <preference name="target-device" value="handset" />
  <preference name="phonegap-version" value="1.5.0" />
  <preference name="permissions" value="none"/>
  <feature name="http://api.phonegap.com/1.0/file"/>
  <feature name="http://api.phonegap.com/1.0/network"/>
```

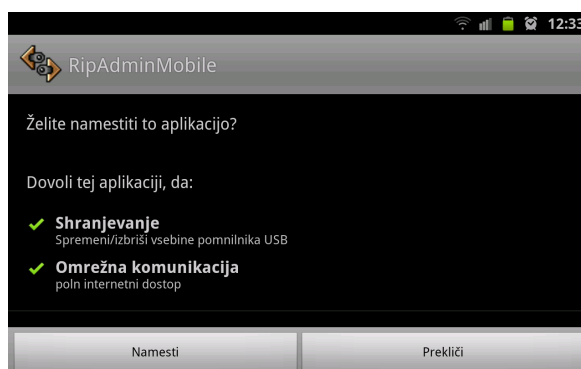
```
<feature name="http://api.phonegap.com/1.0/notification"/>  
<!-- sample preference specification -->  
</widget>
```

5.6.6 Namestitev aplikacije

Najenostavnejši način namestitve aplikacije je neposredno s spletne strani PhoneGap:Build. Vkolikor je na mobilni napravi že nameščena ena od različic aplikacij, je nadgradnjo treba potrditi ob ustreznem vprašanju sistema, kakor je prikazano na sliki 5.11.



Slika 5.11: Pogovorno okno na platformi Android ob zaznavi sistema, da je aplikacija že nameščena.



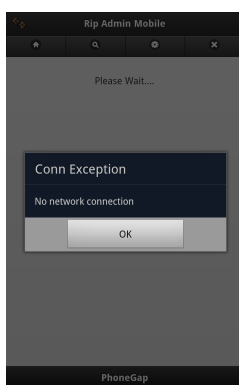
Slika 5.12: Pogovorno okno na platformi Android za potrditev dostopa aplikacije do sistemskih virov naprave.

Ob namestitvi moramo aplikaciji dovoliti dostop do lokalne hrambe podatkov in datotečnega sistema ter do omrežnih komunikacij, kar prikazuje slika 5.12.

Aplikacija se namesti v pomnilnik mobilne naprave. Slaba stran aplikacij PhoneGap je nezmožnost prenosa namestitve na SD spominsko kartico, s katero si po navadi povečamo pomnilniške kapacitete mobilne naprave.

5.6.7 Uporaba aplikacije

Aplikacija za delovanje potrebuje dostop do mobilnega ali brezžičnega omrežja. Po zagonu aplikacije se v ta namen preveri dostopnost omrežij. V primeru, da mobilna naprava ni vključena v nobeno izmed njih, se pojavi opozorilo (slika 5.13).



Slika 5.13: Opozorilo aplikacije, ko ni vzpostavljena povezava v komunikacijsko omrežje.

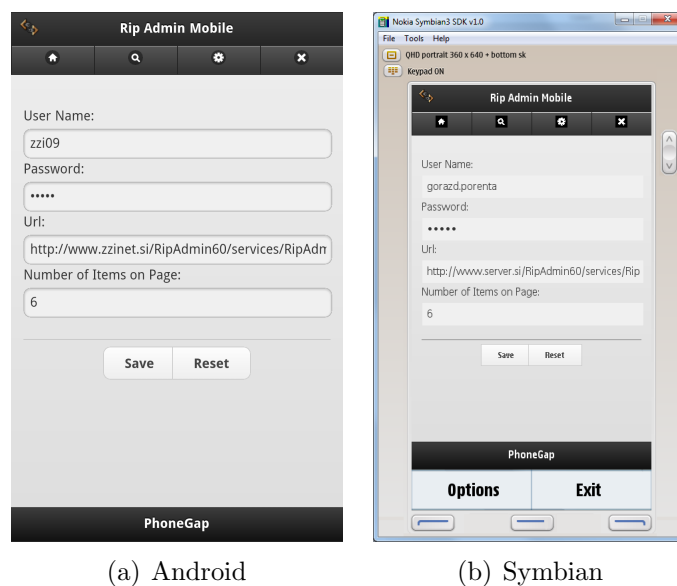
Ena izmed pomanjkljivosti ogrodja PhoneGap je v tem, da nima možnosti sodobnih mobilnih aplikacij, ki v primeru, da uporabnik potrdi povezovanje v mobilno omrežje, samodejno vzpostavi potrebno povezavo.

Nastavitve

Po prvem zagonu aplikacije oziroma ob pritisku na ikono “Nastavitve” v glavi aplikacije se pojavi zaslonska maska (na sliki 5.14) preko katere vnesemo parametre delovanja aplikacije:

- uporabniško ime (User),
- geslo (Password),

- spletni naslov storitve (Url),
- število zapisov na stran (Number of Items on Page).



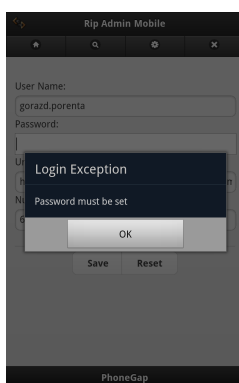
(a) Android

(b) Symbian

Slika 5.14: Nastavitve aplikacije.

Število zapisov na prikazano stran smo dodali po testiranju na mobilnih napravah s povprečno zmogljivim procesorjem, saj se je izkazalo, da prikazovanje strani z nekaj deset zapisi takim napravam povzroča precejšnje breme. Pri analizi je bilo ugotovljeno, da bistveni del problema predstavljajo knjižnice jQueryMobile. V literaturi [2] zasledimo, da se uporaba jQueryMobile odsvetuje zaradi porabe sistemskih (pomnilniških in procesorskih) kapacitet.

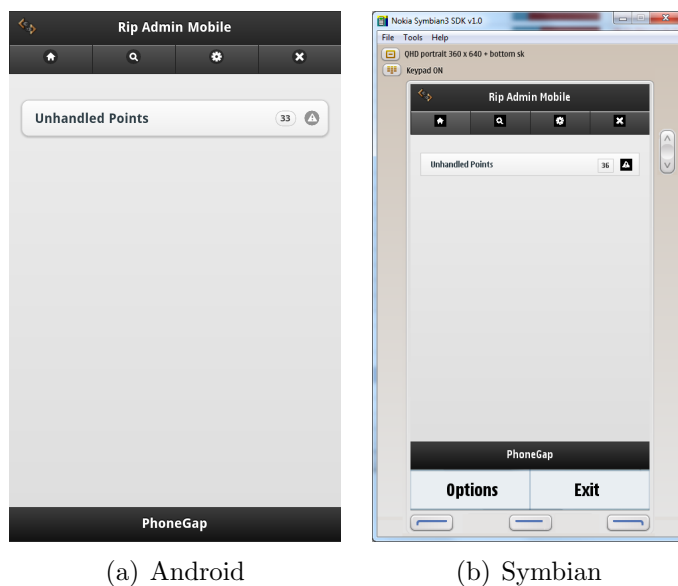
Po potrditvi (s pritiskom na gumb "Save") se vrednosti polj preverijo. Prav tako se preveri tudi dostopnost do mobilnega oziroma brezžičnega omrežja, nakar se izvede prijava v aplikacijo na aplikacijskem strežniku. Vkolikor polja niso pravilno izpolnjena, bodisi z napačnimi vrednostmi bodisi so vrednosti polj prazne, aplikacija prikaže opozorilo z napako, kakor je prikazano na sliki 5.15. Podatki se ob uspešno izvedeni prijavi v spletno storitev shranijo v podatkovno skladišče mobilne naprave. Tako jih ob ponovni uporabi aplikacije ni treba na novo vpisovati. Gumb "Reset" povrne vrednosti iz podatkovnega skladišča.



Slika 5.15: Opozorilo aplikacije pri praznem vnosnem polju za geslo.

Začetna stran

Začetna stran na sliki 5.16 prikazuje število nezaključenih procesov v sistemu elektronske izmenjave.



(a) Android

(b) Symbian

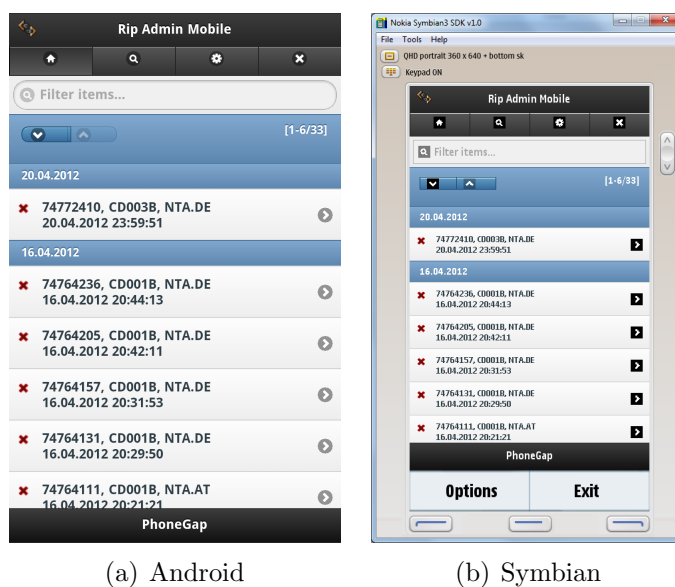
Slika 5.16: Začetna stran aplikacije.

Vkolikor so podatki za prijavo na spletno storitev že v podatkovnem skladišču naprave, je to prva stran po zagonu aplikacije. Pri zagonu se preveri dostopnost mobilnega oziroma brezžičnega omrežja in izvede prijavo v spletno storitev. Na osnovni zaslon se vrnemo s pritiskom na ikono "Domov" kjerkoli v aplikaciji.

Ob pritisku na gumb “Unhandled points” se prikaže seznam nezaključenih procesov.

Seznam nezaključenih procesov

Seznam nezaključenih procesov na sliki 5.17 prikazuje listo zapisov glede na datum začetka procesa.



Slika 5.17: Lista nezaključenih procesov.

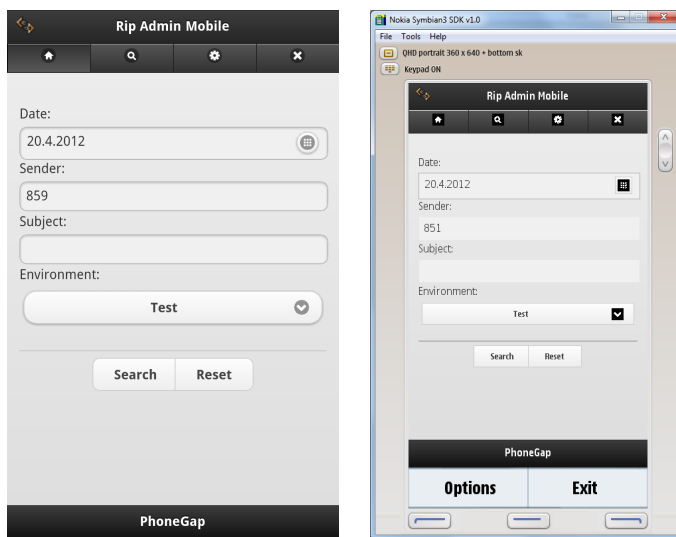
Lista je sortirana po kriteriju od najnovejših do najstarejših nezaključenih procesov. Vsebuje tudi polje za hitro iskanje “Filter items...”, s katerim je mogoče omejiti prikazani seznam. Polje pride do izraza predvsem pri več kakor desetih zapisih na stran, saj so v nasprotnem primeru na zaslonu vidni vsi zadetki (odvisno od velikosti zaslona). Za systemskega inženirja, ki spremlja elektronsko izmenjavo, vsebuje seznam vse pomembne podatke:

- enolični identifikator elektronske izmenjave (CID),
- tip sporočila,
- naslovnika sporočila in
- datum prispetja sporočila v sistem elektronske izmenjave.

V zgornjem desnem kotu, pod glavo aplikacije, je podatek (število), ki pove, kje na seznamu se nahajamo in koliko zapisov je v celotnem seznamu. Po zapisih prehajamo bodisi s pritiskom na gumb pod poljem za hitro iskanje bodisi s potegom prsta po zaslonu. Ob pritisku na proces se prikaže zaslon z podrobnostmi procesa.

Iskanje

Zaslon na sliki 5.18 je dosegljiv preko ikone "Iskanje". Omogoča vnos vrednosti pogoja, po katerem bodo procesi filtrirani.



(a) Android

(b) Symbian

Slika 5.18: Iskalni pogoji aplikacije.

Pogoje sestavljajo naslednja polja:

- datum (Date),
- naslovník (Sender),
- predmet (Subject) in
- okolje (Environment).

Vrednost datumskega polja mora biti nastavljena. V pomoč nam je prikaz koledarja s pritiskom na ustrezno ikono. Obvezno je tudi polje okolje, kjer izbiramo v listi vrednosti med:

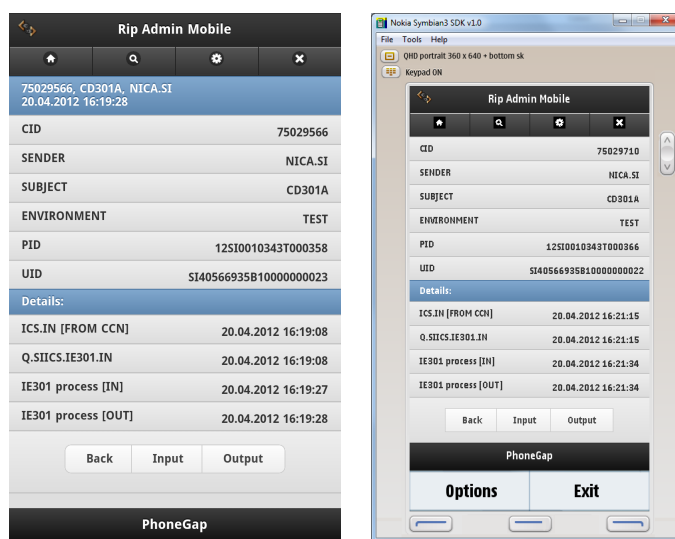
- produkcijskim okoljem (Production),
- testnim okoljem (Test) in
- razvojnim okoljem (Development).

Polji naslovnik in predmet sta opcijski s poljubnimi alfanumeričnimi vrednostmi. Po potrditvi iskalnih kriterijev se vrednosti shranijo v podatkovno skladišče mobilne naprave. Ob pritisku na tipko “Reset” pa vrednosti napolnimo z vrednostmi iz podatkovnega skladišča.

Prikazan seznam je podoben seznamu nezaključenih procesov. Edina razlika je v zeleno obarvani ikoni pred zapisom zaključenega procesa.

Podrobnosti procesa

Na sliki 5.19 prikazani podatki o samem procesu in o operacijah, ki so bile izvedene med procesom.



(a) Android

(b) Symbian

Slika 5.19: Podrobnosti procesa.

Seznam podatkov, ki so na voljo sistemskemu inženirju, vsebuje:

- enolični identifikator elektronske izmenjave (CID),
- naslovnika (Sender),

- predmet (Subject),
- okolje (Environment),
- enolični identifikator posrednika (PID),
- enolični identifikator pošiljatelja (UID) ter
- seznam imen in časov operacij, izvedenih v elektronski izmenjavi.

Seznam gumbov in pomen na podrobnem prikazu podatkov o procesu vsebuje:

- nazaj (Back) - vrnitev na seznam procesov,
- vhodni dokument (Input) - prikaz izvirnega dokumenta,
- izhodno dokument (Output) - prikaz ciljnega dokumenta,
- zaključí (Close) - zaključí nedokončan proces.

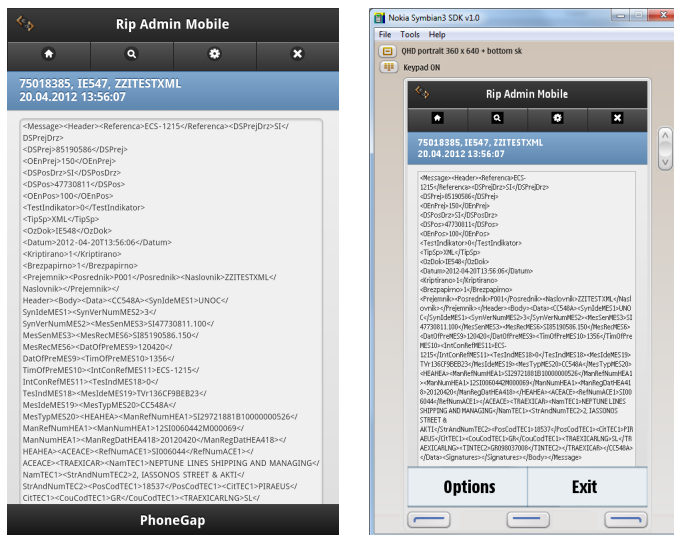
Možnost zaključka je samo na podrobnem prikazu nezaključenega procesa. Vrnitev na seznam procesov lahko dosežemo tudi s potegom prsta po zaslonu v desno stran.

Prikaz dokumenta

Prikaz dokumenta na sliki 5.20 dosežemo preko zaslona s podrobnimi informacijami o procesu s pritiskom na gumba "Input" oziroma "Output". Prvi prikaže vhodni dokument, kakršen je v sistem elektronske izmenjave prispel, drugi pa dokument, kakršen je sistem elektronske izmenjave zapustil. V večini primerov se med seboj razlikujeta, saj se dokumenti v samem procesiranju spreminjajo, in sicer:

- s procesiranjem sporočila, ki tvori neko izhodno sporočilo,
- s transformacijami iz enega formata v drugega (npr. XML - EDIFACT²),
- z dešifriranjem sporočil ali
- z elektronskim podpisovanjem in šifriranjem sporočil.

²Standard Organizacije združenih narodov za elektronsko izmenjavo.



(a) Android

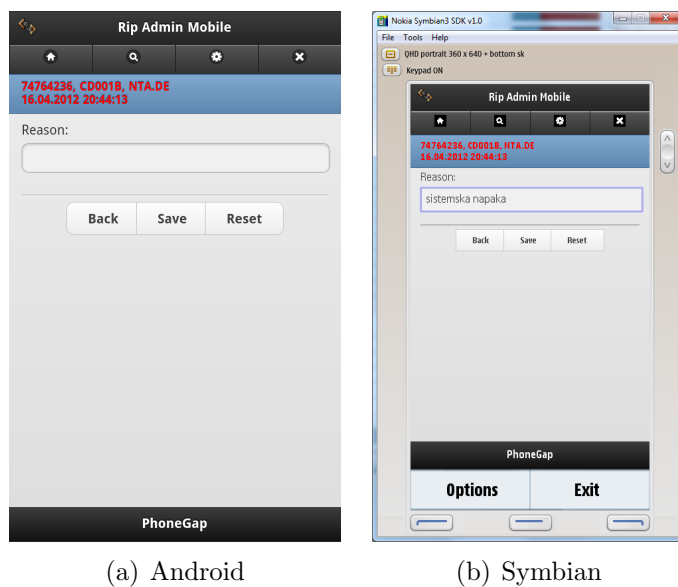
(b) Symbian

Slika 5.20: Pregled sporočila.

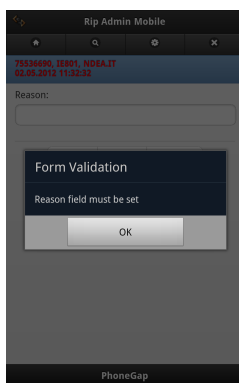
V primeru, ko gre za identičen vhodni in izhodni dokument in je procesiranje krajše, obstaja možnost, da je v podatkovni bazi zapisan samo eden izmed njiju. V takem primeru se v polju, ki sicer prikazuje vsebino, prikaže informacija, da dokumenta ni bilo mogoče najti. Na predhodno zaslonsko masko o podrobnostih procesa se vrnemo s pritiskom na gumb “Back” ali s potegom prsta po zaslonu v desno.

Zaključek procesa

Slika 5.21 prikazuje zaslonsko masko ob pritisku na gumb “Close” na zaslonu o podrobnostih nezaključenega dokumenta. Sistemski inženir analizira težavo, zaradi katere se je procesiranje zaustavilo in je tudi s ponovitvijo procesiranja ni mogoče odpraviti. Po zaključku aktivnosti svoja spoznanja sporoči odgovornim osebam. Proces preko zaslona tudi ročno zaključi na način, da v polje (Reason) vpiše razlog za zastoj. Gre za obvezno in neprazno polje, zato v primeru napačnega vnosa aplikacija javi opozorilo (slika 5.22). Na zaslon z vsebino dokumenta se vrnemo s pritiskom na gumb “Back” ali s potegom prsta v desno smer oziroma s potrditvijo razloga nezaključenega procesa.



Slika 5.21: Zaslonska maska za zaključevanje procesa.

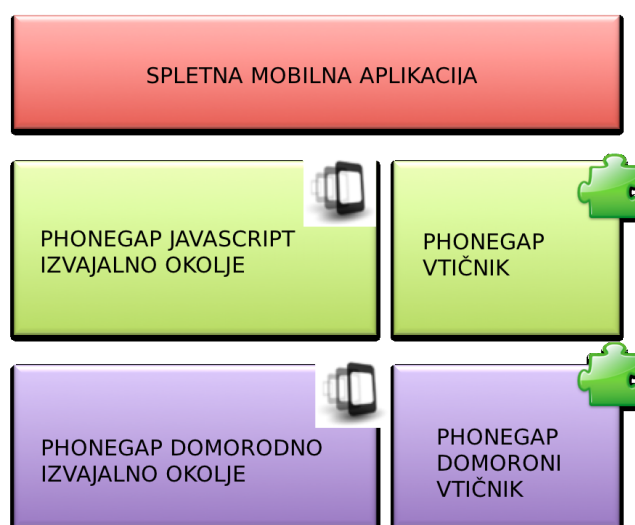


Slika 5.22: Opozorilo aplikacije pri praznem polju razloga za zaključek procesa.

Poglavje 6

Prihodnost PhoneGap

Tehnični razvoj naprav, nadgradnje operacijskih sistemov in razvoj telekomunikacijskih omrežij sprožajo korenite spremembe tudi na področju razvojnih tehnologij. Jezik HTML je v letu 2011 doživel novo različico, ki prinaša nove možnosti pri dinamičnem prikazovanju spletnih vsebin ter podporo novim oblikam interakcije (zasloni, občutljivi na dotik).



Slika 6.1: Arhitektura PhoneGap aplikacij.

Tekom razvoja mobilne aplikacije smo uporabili sorazmerno majhen delež funkcionalnosti, ki jih ponuja ogrodje PhoneGap. Večji del drugih smo preizkusili s primeri aplikacij, ki so na voljo na spletni strani. Spletne razvojne tehnologije so na eni strani dobro poznane, na drugi strani pa se še razvijajo, kar pomeni

nadaljnje približevanje željam tako razvijalcev kot uporabnikov aplikacij. Projekt PhoneGap je pravkar prešel v odprtokodno skupnost Apache pod imenom Cordova. Pod novim imenom je razvojni cikel še krajši. Novo različico ogrodja tako lahko pričakujemo vsak mesec (tabela 6.1). Napreduje tudi storitev v oblaku PhoneGap:Build s podporo vse več operacijskim sistemom. Tako je bila v zadnji posodobitvi dodana prevedba v domorodno kodo za platformo Windows Phone 7.

Datum izdaje	Različica
2.5.2012	1.7.0
24.4.2012	1.7.0rc1
18.4.2012	1.6.1
11.4.2012	1.6.0
29.3.2012	1.6.0rc1
3.3.2012	1.5.0
27.2.2012	1.5.0rc1
1.2.2012	1.4.1
31.1.2012	1.4.0
20.1.2012	1.4.0rc1
17.12.2011	1.3.0

Tabela 6.1: Datumi izdaje različic knjižnic PhoneGap.

Prihodnje različice obljublajo implementacijo standardov W3C v programskih vmesnikih PhoneGap API, kjer še niso podprta. Različica 2.x naj bi podpirala:

- šifriranje (angl. crypto),
- dopolnjeno in dokumentirano nastavitveno datoteko `config.xml`,
- protokol WebSockets,
- Web notifications API¹,
- HTML zajem večpredstavnosti (media capture²),

¹<http://dev.w3.org/2006/webapi/WebNotifications/publish/Notifications.html>

²<http://www.w3.org/TR/2011/WD-html-media-capture-20110414/>

- programski vmesnik koledarja (angl. calendar API³),
- programski vmesnik sporočilnih sistemov (angl. Messaging API⁴),
- programski vmesnik menijev (angl. Menu API), ki bo omogočal izgradnjo menijev na vseh platformah,
- avtomatski zagon phonegap.js v WebView,
- prenehanje podpore `accelerometer`,
- prenehanje podpore `media` (uporaba HTML5 `audio`),
- indeksirano podatkovno bazo⁵,
- podporo večjezičnosti (angl. internationalization),
- prevajanje iz ukazne vrstice (angl. command-line build),
- `Device.exitApp()` kot izhod iz aplikacije, neodvisen od naprave,
- omogočanje paketov vtičnikov,
- arhitekturo, temelječa na vtičnikih - PhoneGap brez vgrajenih programskih vmesnikov (API),
- SMS-sporočanje,
- domorodne kontrole,
- domoroden dostop do stikov,
- analitična orodja za kontrolo prostih sistemskih virov in
- izboljšano omrežno komunikacijo.

Povedali smo že, da je aplikacija PhoneGap iz dveh delov:

- PhoneGap JavaScript API, ki omogoča dostop do komponent naprave iz skriptnega jezika JavaScript in
- domorodne kode, odvisne od operacijskega sistema, ki se proži iz PhoneGap JavaScript API.

³<http://dev.w3.org/2009/dap/calendar/>

⁴<http://www.w3.org/TR/messaging-api/>

⁵<http://www.w3.org/TR/IndexedDB/>

Naveden način ni najbolj primeren za:

- obsežne aplikacije,
- aplikacije, izvajane v ozadju in
- aplikacije s kompleksno programsko logiko.

Tovrstne aplikacije imamo možnost razviti z razširitvijo ogrodja PhoneGap z lastnimi vtičniki, katerih arhitektura je vidna na sliki 6.1. Prav vtičniki pa so arhitektura, na kateri se bo v prihodnje razvijalo ogrodje PhoneGap. Osnovna knjižnica ne bo vsebovala programskih vmesnikov API. Po potrebi bo vsakega izmed njih mogoče dodati kot vtičnik in tako omogočiti aplikaciji dostop do komponent naprave. Aplikacije bodo na ta način manjše in preprostejše.

Poglavje 7

Sklepne ugotovitve

V diplomskem delu smo razvili mobilno aplikacijo s spletnimi tehnologijami in ogrodjem PhoneGap, ki nam je omogočil prevajanje izvorne kode v domorodno za več mobilnih operacijskih sistemov. Testiranje smo opravili na pametnem telefonu z operacijskim sistemom Android različice 2.3.6, posnemovalniku operacijskega sistema Nokia Symbian[^]3 in na pametnem telefonu na platformi Symbian Anna.

Po letih razvoja aplikacij za svetovni splet dobro poznamo jezik HTML, objektni skriptni jezik JavaScript in stilne predloge CSS. Prav tako nam je poznano razvojno okolje Eclipse IDE, in sicer kot komercialna različica podjetja IBM Rational Application Developer. V Eclipse IDE smo namestili vtičnika Android SDK in Aptana Studio, ki je poenostavil razvoj kode v skriptnem jeziku JavaScript, saj omogoča avtomatsko zaključevanje ukazov in preverjanje veljavnosti napisane kode, kar pri osnovnem orodju ni možno.

Prototipno smo se lotili razvoja dostopa in uporabe obstoječih spletnih storitev. Programsko logiko v skriptnem jeziku JavaScript je bilo treba dopolniti s kodo za ohranjanje HTTP-seje. Na strani odjemalca je sejo po uspešni prijavi treba ohraniti (preko HTTP-piškotka).

Po navodilih na spletni strani skupnosti PhoneGap smo se najprej lotili manjših primerov aplikacij, ki so pokrivalo po en tip dostopa do sistemskih virov naprave. Delovanje smo preverili v posnemovalniku, nato pa še na pametnem telefonu z identičnim operacijskim sistemom. Za uporabniški vmesnik smo uporabili odprtokodno rešitev jQueryMobile, ki pa se je izkazala za ozko grlo. Na osnovi prototipov dostopa do spletne storitve in primerov aplikacij v PhoneGap smo se odločili za razvoj mobilne različice aplikacije RipAdmin, ki se trenutno uporablja v podjetju ZZI, d. o. o., za spremljanje delovanja elektronske izmenjave Carinskega informacijskega sistema Republike Slovenije.

Kljub zmogljivemu procesorju pametnega telefona je aplikacija že v zgodnjih fazah testiranja delovala neodzivno. Analiza problema je pokazala na prikaz uporabniškega vmesnika, izdelanega s knjižnicami jQueryMobile. Za normalno delovanje aplikacije je bilo treba omejiti število zapisov na strani. Ta podatek je postal tudi del nastavitvev in se hrani v podatkovnem skladišču mobilne naprave, v kolikor na platformi ta možnost obstaja. Prevajanje za operacijski sistem Android smo izvajali v razvojnem okolju. Prevajanje izvirne kode za druge platforme smo opravili preko storitve v oblaku PhoneGap:Build.

V aplikacijo se prijavimo z uporabniškim imenom in geslom, ki sta identična kakor za prijavo v aplikacijo, namenjeno osebnim računalnikom. Podatka in naslov spletne storitve se hranita v podatkovnem skladišču mobilne naprave. Pred klicem spletne storitve preverimo, ali ima naprava vzpostavljeno povezavo z brezžičnim ali mobilnim omrežjem. Prenos na operacijski sistem Symbian je terjal nekaj prilagoditev. Na tej platformi nista podprti možnosti preveriti, ali smo priklopljeni v omrežje, in pisanje oziroma branje iz/v podatkovnega skladišča mobilne naprave. Na napravah z OS Symbian je zato ob vsakem zagonu aplikacije treba vpisati uporabniško ime, geslo in naslov spletne storitve. Naslov spletne storitve nima privzete vrednosti zaradi javnega značaja aplikacije na storitvi v oblaku PhoneGap:Build.

Nadaljnje nadgradnje aplikacije bodo zahtevale tudi nadgradnjo obstoječe spletne storitve, in sicer s kontaktnimi števkami in naslovi elektronske pošte odgovornih za posamično področje elektronske izmenjave. Smiselno bi bilo vključiti tudi pošiljanje SMS-sporočil, kar pa bo v ogrodje PhoneGap vključeno v prihodnosti.

Prednosti razvoja v ogrodju PhoneGap:

- izraba obstoječega znanja in izkušenj spletnih tehnologij; ob poznavanju HTML, CSS in JavaScript je treba spoznati le specifikacijo programskih vmesnikov (API) ogrodja PhoneGap,
- odlično premošča razliko med spletnimi aplikacijami in komponentami mobilnih naprav; na enostaven način je mogoč dostop do kamere, stikov itd.,
- lahek dostop do spletnih storitev z uporabo dodatnih knjižnic.

Pri razvoju v ogrodju moramo biti pazljivi:

- zgolj zaradi ogrodja PhoneGap ni nobenega jamstva, da bo delujoča aplikacija na enem operacijskem sistemu (na primer Android) brezpogojno delovala tudi na drugih,

- če želimo izvorno kodo sami prevajati v domorodno, moramo imeti za vsako platformo vzpostavljena razvojna orodja,
- aplikacije z visoko stopnjo grafičnih elementov in animacij bodo terjale svoj davek pri procesorski obremenitvi naprave.

Kljub vsem omejitvam, ki jih je treba imeti v mislih že pred samim razvojnim ciklom, je ogrodje za večino poslovnih aplikacij primerno okolje.

Slike

4.1	Logotipi mobilnih operacijskih sistemov, podprtih v ogrodju PhoneGap.	26
4.2	Dostopnost sistemskih virov naprav v ogrodju PhoneGap po mobilnih platformah.	31
5.1	Aplikacija RipAdmin v okolju Microsoft Windows 7.	42
5.2	Arhitektura aplikacije.	43
5.3	Primeri uporabe aplikacije RipAdmin.	44
5.4	Podrobni podatkovni model aplikacije RipAdmin.	45
5.5	Razvojno orodje Eclipse IDE 3.7.0.	47
5.6	Razred CProcess vsebuje attribute procesa.	51
5.7	Shema vključenosti in odvisnosti datotek HTML in JavaScript.	53
5.8	Posnemovalnik mobilne naprave z operacijskim sistemom Android.	54
5.9	Konfiguracija posnemovalnika Android SDK.	54
5.10	Javno dostopna aplikacija RipAdmin mobile na spletni strani PhoneGap:Build.	55
5.11	Pogovorno okno na platformi Android ob zaznavi sistema, da je aplikacija že nameščena.	57
5.12	Pogovorno okno na platformi Android za potrditev dostopa aplikacije do sistemskih virov naprave.	57
5.13	Opozorilo aplikacije, ko ni vzpostavljena povezava v komunikacijsko omrežje.	58
5.14	Nastavitve aplikacije.	59
5.15	Opozorilo aplikacije pri praznem vnosnem polju za geslo.	60
5.16	Začetna stran aplikacije.	60
5.17	Lista nezaključenih procesov.	61
5.18	Iskalni pogoji aplikacije.	62
5.19	Podrobnosti procesa.	63
5.20	Pregled sporočila.	65

5.21	Zaslonska maska za zaključevanje procesa.	66
5.22	Opozorilo aplikacije pri praznem polju razloga za zaključek procesa.	66
6.1	Arhitektura PhoneGap aplikacij.	67

Tabele

1.1	Svetovna prodaja pametnih telefonov po operacijskem sistemu v 4. četrtnem letju 2011.	4
2.1	Primeri gibov na zaslonih, občutljivih na dotik.	7
3.1	Različice iOS in podprte mobilne naprave.	11
3.2	Različice platforme Android, kodna imena, API nivo in delež dostopov.	13
3.3	Različice BlackBerry OS in podprte naprave BlackBerry.	14
3.4	Različice Symbian OS, leto izdaje in na novo podprte tehnologije.	18
6.1	Datumi izdaje različic knjižnic PhoneGap.	68

Literatura

- [1] T. Myer, *Beginning PhoneGap*, John Wiley & Sons, Inc., Indianapolis, Indiana, 2012.
- [2] A. Lunny, *PhoneGap*, Packt Publishing Ltd., Birmingham, UK, 2011.
- [3] (2012) About PhoneGap. Dostopno na:
<http://phonegap.com/about/>
- [4] (2012) PhoneGap API Reference. Dostopno na:
<http://docs.phonegap.com/en/1.5.0/index.html>
- [5] (2012) Mobile device. Dostopno na:
<http://dictionary.sensagent.com/mobile+device/en-en/>
- [6] (2012) Mobile Application (Mobile App). Dostopno na:
<http://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [7] (2012) HTML. Dostopno na:
<http://sl.wikipedia.org/wiki/HTML>
- [8] (2012) CSS. Dostopno na:
<http://sl.wikipedia.org/wiki/CSS>
- [9] (2012) JavaScript. Dostopno na:
<http://sl.wikipedia.org/wiki/Javascript>
- [10] (2012) Web service. Dostopno na:
http://en.wikipedia.org/wiki/Web_service
- [11] (2012) iOS. Dostopno na:
<http://mojapple.net/wiki/iOS>

- [12] (2012) iOS App Programming Guide. Dostopno na:
<https://developer.apple.com/library/ios/documentation/iphone/conceptual/iphoneosprogrammingguide/iphoneappprogrammingguide.pdf>
- [13] (2012) Android overview. Dostopno na:
http://www.openhandsetalliance.com/android_overview.html
- [14] (2012) About the Eclipse Foundation. Dostopno na:
<http://www.eclipse.org/org/>
- [15] (2012) Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth. Dostopno na:
<http://www.gartner.com/it/page.jsp?id=1924314>
- [16] (2012) Android Platform Versions. Dostopno na:
<http://developer.android.com/resources/dashboard/platform-versions.html>
- [17] (2012) BlackBerry Developer. Dostopno na:
<https://bdsc.webapps.blackberry.com/devzone/>
- [18] (2012) Windows Phone Development. Dostopno na:
[http://msdn.microsoft.com/en-us/library/ff402535\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402535(v=vs.92).aspx)
- [19] (2012) HP WebOS Developer Center. Dostopno na:
<https://developer.palm.com/>
- [20] (2012) Nokia Developer. Dostopno na:
<http://www.developer.nokia.com/>
- [21] (2012) History of Symbian Operating System, Now Plane to Put it into Dead Pool. Dostopno na: <http://infodunya.com/history-of-symbian-operating-system-now-plane-to-put-it-into-dead-pool/>
- [22] (2012) Bada. Dostopno na:
<http://www.bada.com/>
- [23] (2012) 6 Multi-Touch Gesture Icon Sets. Dostopno na:
<http://favbulous.com/post/695/6-multi-touch-gesture-icon-sets>

-
- [24] (2012) 10 of Best Cross-Platform Mobile Development Tools. Dostopno na: <http://www.toolsjournal.com/item/157-10-of-best-cross-platform-mobile-development-tools>
- [25] (2012) Rhomobile: mobilize your entrprise apps. Dostopno na: <http://www.rhomobile.com/>
- [26] (2012) appMobi{!}. Dostopno na: <http://www.appmobi.com/>
- [27] (2012) apccelerator. Dostopno na: <http://www.appcelerator.com/>
- [28] (2012) dragonRAD overview. Dostopno na: <http://www.dragonrad.com/overview>
- [29] (2012) Sencha Touch 2.0. Dostopno na: <http://www.sencha.com/>
- [30] (2012) UnityMOBILE. Dostopno na: <http://www.unitymobile.com/>
- [31] (2012) MoSync. Dostopno na: <http://www.mosync.com/>
- [32] (2012) Qt - Cross-Platform application and UI framework. Dostopno na: <http://qt.nokia.com/>
- [33] (2012) The Real Power of Mobile Computing. Dostopno na: http://www.superwaba.com.br/en/tc_overview.asp
- [34] (2012) Grafična predstavitev podprtih platform. Dostopno na: <http://phonegap.com/start>
- [35] (2012) Grafična predstavitev podprtih podprtih funkcionalnost po mobilnih platformah. Dostopno na: <http://phonegap.com/about/features/>