

**SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET**

IVAN PODVOREC

**MOBILNE NATIVNE APLIKACIJE
BAZIRANE NA WEB STANDARDIMA**

DIPLOMSKI RAD

Zagreb, 2013.



Sveučilište u Zagrebu
Grafički fakultet

IVAN PODVOREC

MOBILNE NATIVNE APLIKACIJE BAZIRANE NA WEB STANDARDIMA

DIPLOMSKI RAD

Mentor:
prof. dr. sc., Klaudio Pap

Student:
Ivan Podvorec

Zagreb, 2013.

*Zahvaljujem se mentoru prof. dr. sc.,
Klaudiju Papu na stručnim savjetima
prilikom izrade ovog diplomskog rada.*

SAŽETAK

Pametni mobilni uređaji (smartphone i tablet) su sve zastupljeniji te postaju naša svakodnevica što dovodi do sve veće potražnje za mobilnim aplikacijama. Kako bi se doprlo do što većeg broja potencijalnih klijenata i kako bi se ti isti klijenti zadržali potrebno je, uz danas neizostavne web stranice (ili portale), imati i mobilnu aplikaciju dostupnu za što veći broj mobilnih platformi. Razvoj aplikacije za više platformi je prilično zahtjevan jer svaka platforma koristi drugačiji programski jezik te je potreban velik broj razvijatelja. Postoji i alternativno rješenje za razvoj „nativnih“ mobilnih aplikacija. One nisu izrađene programiranjem u nativnom programskom jeziku, ali se instaliraju na isti način kao i standardne nativne aplikacije. Tako je moguće razvijati multiplatformsku mobilnu aplikaciju baziranu na web tehnologijama uz pomoć HTML, CSS i Javascript-a te korištenjem besplatnih servisa.

Za praktični dio rada izraditi će se jedna mobilna aplikacija zasnovana na web tehnologijama za Android operacijski sustav te uz minimalne preinake za ostale mobilne platforme (iOS, Symbian, Blackberry, WebOS i Windows Phone). U radu će se istražiti prednosti, nedostaci, problemi i moguća ograničenja mobilnih aplikacija zasnovanih na web standardima u odnosu na nativne aplikacije. Također će istraživati i problematika izrade dizajna aplikacije neovisnog o dimenzijama ekrana kojeg posjeduje uređaj.

Ključne riječi: pametni mobilni uređaji, aplikacije, web tehnologija, HTML, CSS

ABSTRACT

Smart mobile devices (smartphones and tablets) are becoming more and more frequent, and part of our everyday lives, which leads to a growing demand for mobile applications. In order to reach the largest possible number of potential customers and to keep those same customers we need, along with today inevitable website (or portals), to have a mobile application available on as many mobile platform as possible. Developing applications for multiple platforms is quite challenging because each platform uses a different programming language and needs a large number of developers. There is also an alternative solution for the development of "native" mobile applications. They are not made by programming in native programming language, but it is installed in the same manner as conventional native applications. Thus it is possible to develop multi-platform mobile application based on web technology using HTML, CSS and JavaScript, and some free services.

For the practical part of the thesis will be made one mobile application based on Web technology for the Android operating system, and with minimal changes it will be adjusted for for other mobile platforms (iOS, Symbian, BlackBerry, WebOS and Windows Phone). In this thesis will explore the advantages, disadvantages, problems and possible limitations of mobile applications based on web standards compared to a native applications. It will also explore the issue of adapting the design of the application, regardless of the dimensions of the screen that has the device.

Keywords: smart mobile devices, applications, Web technologies, HTML, CSS

SADRŽAJ

1. UVOD.....	1
2. MOBILNI OPERACIJSKI SUSTAVI I PLATFORME.....	2
2.1. Android operacijski sustav	2
2.1.1. Razvoj Android aplikacija	2
2.2. iOS – Operacijski sustav za iPhone, iPod i iPad.....	4
2.2.1. Razvoj iOS aplikacija	4
2.3. Windows Phone	5
2.4. BlackBerry	7
2.5. Usporedba mobilnih operacijskih sustava.....	7
3. WEB STANDARDI – HTML, CSS, JAVASCRIPT	9
3.1. HTML	9
3.1.1. HTML 5	9
3.2. Cascading Style Sheets – CSS	10
3.2.1. CSS 3	13
3.3. JavaScript	14
3.3.1. JSON.....	15
4. RAZVOJNE OKOLINE ZA RAZVOJ „NATIVNIH“ MOBILNIH APLIKACIJA TEMELJENIH NA WEB STANDARDIMA	16
4.1. jQuery Mobile	17
4.2. PhoneGap	18
4.2.1. PhoneGap nativni plugin-ovi.....	20
4.2.2. PhoneGap build.....	21
4.3. Prednosti i nedostaci hibridnih mobilnih aplikacija.....	21
5. PRAKTIČNI RAZVOJ APLIKACIJE.....	25
5.1. Grafičko sučelje aplikacije.....	26
5.2. Opis programskog dijela aplikacije.....	32
5.2.1. PhoneGap Build konfiguracija aplikacije, priprema za izradu nativnih instalacija	43
5.2.2. PhoneGap Build izrada nativnih instalacija.....	45
6. ZAKLJUČAK.....	47
7. LITERATURA	48
POPIS PRILOGA	49
POPIS SLIKA.....	50
POPIS TABLICA	51
POPIS IZVORNIH KODOVA.....	52
POPIS OZNAKA I KRATICA.....	53

SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET
Getaldićeva 2
Zagreb, 18. 2. 2013.

Temeljem podnijetog zahtjeva za prijavu teme diplomskog rada izdaje se

R J E Š E N J E

kojim se studentu/ici Ivanu Podvorecu, JMBAG 0246012807, sukladno čl. 5. st. 5. Pravilnika o izradi i obrani diplomskog rada od 13. 2. 2012. godine, odobrava izrada diplomskog rada, pod naslovom: Mobilne nativne aplikacije bazirane na web standardima, pod mentorstvom prof. dr. sc. Klaudija Papa.

Sukladno čl. 9. st. 1. Pravilnika o izradi i obrani diplomskog rada od 13. 2. 2012. godine, Povjerenstvo za nastavu, završne i diplomske ispite predložilo je ispitno Povjerenstvo, koje je potvrđeno odlukom Fakultetskog vijeća od 28. 1. 2013. godine kako slijedi:

1. prof. dr. sc. Koren Antun, predsjednik/ica
2. prof. dr. sc. Pap Klaudio, mentor/ica
3. prof. dr. sc. Žiljak Vilko, član/ica

Dekanica

prof. dr. sc. Diana Milčić

1. UVOD

Prvo će se istražiti mogućnosti razvojnih okruženja za izradu „nativnih“ mobilnih aplikacija baziranih na web tehnologijama i standardima. Zatim će se odabrati najpogodnije razvojno okruženje za izradu „nativne“ mobilne aplikacije bazirane na web standardima. Treba istražiti moguća ograničenja u razvoju i pristupu nativnim funkcijama u takvim aplikacijama, a da se ne izgubi ciljana kvaliteta i prenosivost na druge operativne sustave. U eksperimentalnom djelu rada treba testirati moguća proširenja aplikacija uz korištenje nativnih plugin-ova. Izradom takve aplikacije moći će se istražiti njene daljnje mogućnosti i moguća ograničenja.

Cilj istraživanja je izraditi funkcionalnu "nativnu" mobilnu aplikaciju za Android operacijski sustav zasnovanu na web standardima (HTML, CSS i Javascript), s minimalnim korištenjem nativnog programskog jezika, koja može uz minimalne promjene biti prilagođena za druge mobilne platforme. Treba dokazati da će se moći dobiti nativna funkcionalnost uz korištenje nativnih plugin-ova unutar web tehnologija i standarda. Istovremeno takvu aplikaciju se mora moći prenijeti na drugi operacijski sustav za druge uređaje uz minimalne preinake. Trebalo bi biti moguće bez posebnih znanja nativnih programskih jezika dobiti „nativnu“ mobilnu aplikaciju za više platformi.

2. MOBILNI OPERACIJSKI SUSTAVI I PLATFORME

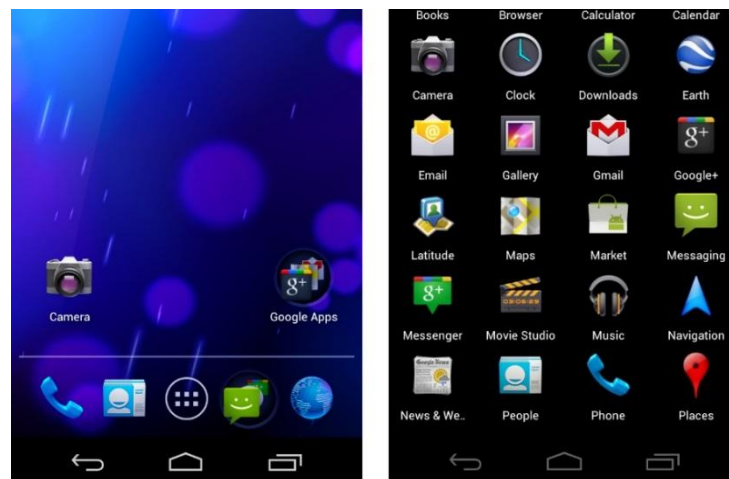
2.1. Android operacijski sustav

Android operacijski sustav objavljen je 2007. godine od strane Google.¹ Prva komercijalna verzija Androida objavljena je 2008. godine² te je u kratkom vremenskom periodu postigao veliki uspjeh.



Slika 2.1 Android korisničko sučelje, verzija Androida 4.0.x

<http://developer.android.com/about/versions/android-4.0-highlights.html>



2.1.1. Razvoj Android aplikacija

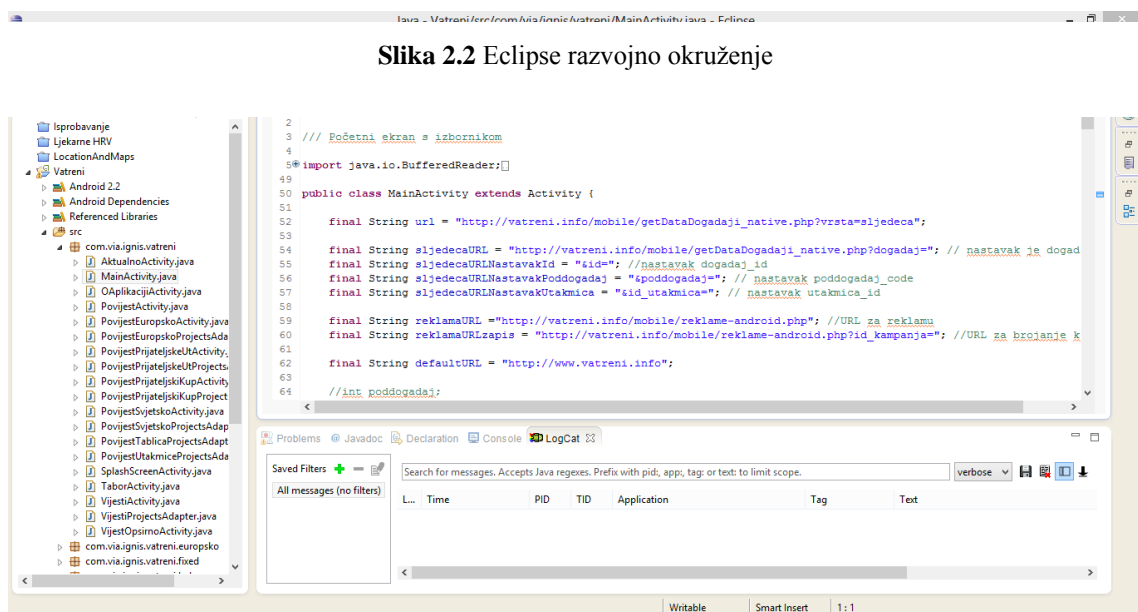
Android aplikacije razvijaju najčešće se razvijaju u Java programskom jeziku korištenjem Android SDK (eng. *Software Development Kit*, set razvojnih programa za razvoj aplikacija za određenu platformu). Također postoji i mogućnost razvoja u C/C++ programskom jeziku, ali tada se umjesto Android SDK koristi Android NDK (eng. *Native Development Kit*, set razvojnih programa za razvoj u izvornom kodu). Razvoj aplikacija u C/C++ omogućuje brže izvršavanje aplikacija, ali sam razvoj je puno složeniji nego razvoj isključivo u Java programskom jeziku. Za aplikacije koje

¹ *** http://www.openhandsetalliance.com/press_110507.html – Open Handset Alliance, studeni 2007.

² *** <http://android-developers.blogspot.com/2008/09/announcing-android-10-sdk-release-1.html> – rujan 2008.

zahtijevaju veliko opterećenje procesora razvoj u C/C++ programskom jeziku bolji je odabir nego razvoj u Java programskom jeziku.³

Razvoj Android aplikacija je moguć na svim operacijskim sustavima koji imaju mogućnost pokretanja Java aplikacija (Windows, Mac, Linux). Razvojno okruženje (eng. *development environment*) u kojem se razvijaju Android aplikacije je Eclipse (**Slika 2.2**) uz korištenje Android razvojnih alata (ADT – eng. *Android Development Tool*) i Android SDK.



Slika 2.2 Eclipse razvojno okruženje

Korištenjem Android razvojnih alata (ADT) u Eclipse-u omogućuje izradu aplikacija, otkrivanje pogrešaka (eng. *Dalvik Debug Monitor Server - DDMS*), i testiranje aplikacija uz pomoć Android emulatora koji je sastavni dio Android SDK.

Aplikacije se objavljuju na Google Play Store-u, a da bi se mogle objaviti potrebno je uplatiti jednokratnu naknadu od 25\$ (otprilike 150kn). Trenutno se može preuzeti preko 700 000 aplikacija.⁴

³ *** <http://developer.android.com/tools/sdk/ndk/index.html> - Android

⁴ *** <http://www.businessweek.com/news/2012-10-29/google-says-700-000-applications-available-for-android-devices> - Listopad 2012.

2.2. iOS – Operacijski sustav za iPhone, iPod i iPad

Prva verzija pojavila se 2007. godine pod nazivom iPhone OS te je ostvarila veliki uspjeh u kratkom periodu. Godine 2010. Apple odlučuje promijeniti ime iPhone OS u iOS. Verzija iOS 4.x donosi značajne promjene s čak 1500 novih API-ja (eng. *application programming interface*, sučelje za programiranje aplikacija) za izradu aplikacija te omogućuje po prvi put višezadaćnost (eng. *multitasking*).⁵



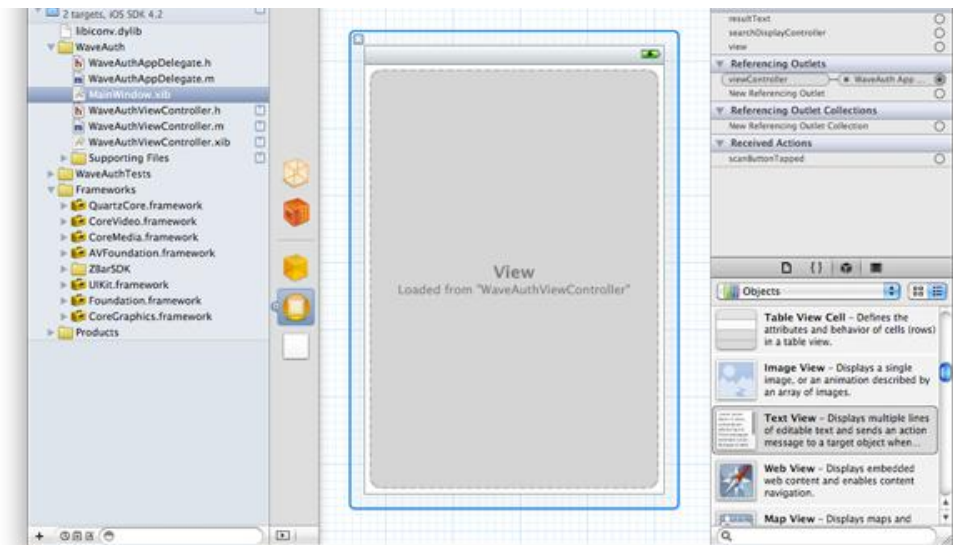
2.2.1. Razvoj iOS aplikacija

Za razvoj iOS aplikacija koristi se programski jezik Objective C. To je standardni programski jezik za Macintosh baziran na C programskom jeziku. Razvojno okruženje za razvoj iOS aplikacija je Xcode (**Slika 2.4**). Također potreban nam je iOS SDK koji omogućuje pristup svim API-ima operacijskog sustava, sadrži alate za otkrivanje pogrešaka te iOS simulator.

⁵ *** <http://www.webcitation.org/5oqYO6thc> – Apple, srpanj 2010.

Slika 2.4 Xcode 4 sučelje

<http://upload.wikimedia.org/wikipedia/en/8/8c/Xcode4.png>



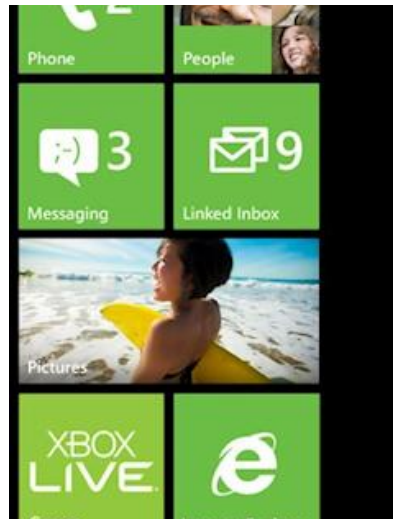
Xcode je besplatan razvojni alat namijenjen isključivo za izvršavanje na Macintosh platformi. Da bi se izrađena aplikacija mogla testirati na pravom uređaju potrebna je licenca *iOS Developer program* koja se plaća na godišnjoj razini od 99\$ (otprilike 600kn). Licenca *iOS Developer program* također omogućava objavu i distribuciju aplikacija na App Store-u.

2.3. Windows Phone

Windows Phone je najmlađi operacijski sustav za mobilne uređaje koji je objavljen 2010. kao Microsoftov odgovor na Google-ov Android te Apple-ov iOS.

Slika 2.5 Windows Phone 7 početni ekran

http://www.hindustantimes.com/Images/HTEditImages/Images/1_WP7_start_screen.png



Microsoft se odlučio na drukčiji pristup prilikom razvoja korisničkog sučelja te ga je preuzeo sa svojeg multimedijalnog reproduktora Zune. Samo korisničko sučelje je bitno drugačije od konkurentskih operacijskih sustava. Za razvoj aplikacija se mogu koristiti programski jezici iz skupine .NET programskih jezika.

Za razvoj aplikacija za ovaj mobilni operativni sustav potreban je Windows Phone SDK koji sadrži sve programe i alate koji su potrebni za razvoj aplikacije. Windows Phone SDK sadrži Microsoft Visual Studio Express for Windows Phone kao razvojno okruženje, Windows Phone emulator za testiranje aplikacija. Da bi se mogle razvijati aplikacije za najnoviji Windows Phone 8 potrebno je imati računalo s instaliranim Windows 8 64-bitnim operacijskim sustavom.⁶ Ukoliko želimo objaviti aplikaciju i omogućit njeno preuzimanje na Windows Phone mobilne uređaje potrebno je registrirati Microsoft Dev Center račun. Naknada za otvaranje Dev Center računa iznosi 99\$ (otprilike 600 kuna).⁷

⁶ *** [http://msdn.microsoft.com/library/windowsphone/develop/ff402529\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windowsphone/develop/ff402529(v=vs.105).aspx) – Microsoft, Studeni 2012

⁷ *** [http://msdn.microsoft.com/library/windowsphone/help/jj206719\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windowsphone/help/jj206719(v=vs.105).aspx) – Microsoft, Studeni 2012

2.4. BlackBerry

Prva verzija ovog mobilnog operacijskog sustava objavljena je 1999. godine od strane Research In Motion (RIM). Godine 2010. RIM je objavio BlackBerry Tablet OS namjenjem tabletu BlackBerry PlayBook⁸. S pojavom iOS-a i Androida RIM-ov BlackBerry OS je ušao u probleme i vrlo brzo je gubio svoj tržišni udio. Početkom 2013. godine RIM planira izdati BlackBerry 10 OS koji bi trebao povećati njihov tržišni udio i ući u borbu s iOS-om te Androidom.

Aplikacije za BlackBerry OS razvijaju se u programskom jeziku Java. Kako bi ih počeli razvijati potreban nam je BlackBerry Java SDK u koji je uključen BlackBerry plugin za Eclipse razvojno okruženje.⁹ Također aplikacije se mogu razvijati u Adobe AIR-u, HTML5 te u nativnom okruženju koristeći C/C++ programski jezik. BlackBerry je izdao i alat za jednostavno migriranje i prilagodbu aplikacije napravljene za Android operacijski sustav u aplikaciju za BlackBerry OS. Da bi izrađenu aplikaciju mogli objaviti na BlackBerry App World servisu i omogućit ju za preuzimanje na uređaje potreban je besplatan BlackBerry App World Vendor račun. Preuzimanje novih i nadogradnja postojećih aplikacija na BlackBerry uređaju je moguće s BlackBerry App World servisa.

2.5. Usporedba mobilnih operacijskih sustava

Razvoj Android i BlackBerry aplikacija moguć je na svim operacijskim sustavima. S druge strane razvoj aplikacija za iOS je moguć isključivo na Apple platformi i operacijskom sustavu, a za Windows Phone je isključivo moguć na Microsoft platformi i operacijskom sustavu. Za razvoj aplikacija za gore navedene mobilne operacijske sustave potrebno je poznavanje tri različita programska jezika (Java, Objective-C, .NET programski jezik).

Problem razvijateljima aplikacija za Android operacijski sustav stvara brzo izdavanje novih verzija sustava (nekoliko godišnje) pa su potrebne česte prilagodbe već izdanih aplikacija novim verzijama operacijskog sustava. Također veliki problem stvara

⁸ *** <http://www.marketwire.com/press-release/RIM-Unveils-The-BlackBerry-PlayBook-NASDAQ-RIMM-1325727.htm> - Rujan 2010.

⁹ *** <https://developer.blackberry.com/java/download/eclipse/> - BlackBerry

i veliki broj različitih uređaja sa različitim specifikacijama hardvera (veličina i rezolucija ekrana, vrsta i brzina procesora, različiti senzori).

Sve te razlike, ograničenja u korištenju određenih operacijskih sustava na kojima se može razvijati aplikacija uvelike povećava cijenu i vrijeme razvoja jer je potrebno raditi minimalno tri „različite“ aplikacije kako bi ista bila dostupna na što većem broju uređaja. BlackBerry je izdao alat za transformaciju Android aplikacija u aplikacije za BlackBerry OS pa je potrebno razviti samo tri aplikacije.

Postoji alternativni način razvoja aplikacija za mobilne operacijske sustave korištenjem HTML (eng. *Hyper Text Markup Language*), CSS (eng. *Cascading Style Sheets*) i JavaScript tehnologija. U kasnijem dijelu radu ću ovaj način pobliže objasniti.

Objava Android aplikacije na Play Store-u ne prolazi testiranje već je ista dostupna u kratkom periodu od *upload-a* iste na Google Play Store. S druge strane Apple, Microsoft i RIM svaku aplikaciju prijavljenu za objavu detaljno testiraju i garantiraju da će raditi na svim njihovim uređajima za koje se traži odobrenje za objavu te aplikacije neće biti dostupne sve dok njihovi uvjeti nisu ispunjeni.

3. WEB STANDARDI – HTML, CSS, JAVASCRIPT

3.1. HTML

HTML (eng. *Hyper Text Markup Language*) je glavni jezik za objavu i prikaz sadržaja na internetu iza kojeg stoji World Wide Web konzorcij.¹⁰ Prvi javno dostupan opis HTML jezika pojavio se 1991. Taj opis je objavio Tim Barners-Lee i sadržavao je 20 elemenata od kojih se u HTML4 verziji koristi 13 elemenata. HTML4 objavljen je 1997., ali finalna verzija standarda je objavljena 1999. godine kao HTML 4.01.

```
<html>
  <head>
    <title>Primjer HTML bez CSS</title>
  </head>
  <body>
    <h1>Naslov</h1>
    <p>HTML paragraf</p>
  </body>
</html>
```

Izvorni kod 3.1 Jednostavan HTML dokument

Primjer jednostavnog HTML dokumenta (**Izvorni kod 3.1**):

3.1.1. HTML 5

Godine 2004. započeo je razvoj HTML 5. HTML 5 je još uvijek u razvoju i konačna standardizacija se očekuje 2014. godine.¹¹ HTML 5 uvodi sučelje za programiranje aplikacija (API) što omogućuje izradu kompleksnih web aplikacija. Također HTML 5 uvodi nove elemente (`<section>`, `<article>`, `<header>` i `<nav>`) kako bi se poboljšala semantička čitljivost samih HTML dokumenata tražilicama. Uvode se sintaktički elementi `<video>`, `<audio>`, `<canvas>`. Velika većina mogućnosti HTML 5 je izrađena uzimajući u obzir da mogu biti pokrenute ne uređajima slabijih hardverskih specifikacija (na pametnim telefonima te tabletima).

¹⁰ *** <http://www.w3.org/html/> - World Wide Web Consortium (W3C)

¹¹ *** <http://www.w3.org/2011/02/htmlwg-pr.html> - W3C

Tvrtka Strategy Analytics je 2011. objavila istraživanje u kojem predviđa da će u 2013. godine broj mobilnih uređaja koji podržavaju HTML 5 narasti do čak jedne milijarde.¹² Osim pametnih telefona i tablet uređaja također će i neki takozvanih običnih mobilnih telefona imati podršku za HTML5. S obzirom na te podatke HTML 5 postaje brzorastuća tehnologija koja je prisutna u svakom uglu od pametnih telefona pa do automobila i sl.

3.2. Cascading Style Sheets – CSS

CSS (eng. *Cascading Style Sheets*) je mehanizam za dodavanje stila i oblikovanja (font, boja, razmak, dimenzije) web dokumentima iza kojeg stoji World Wide Web konzorcij (W3C).¹³ Prva objava CSS 1 bila je 1996. godine.

Velika prednost CSS-a je taj da je sadržaj odvojen od oblikovanja. Postoje tri načina uključivanja CSS-a u HTML dokument: linijski, unutrašnji i vanjski. Najveći prioritet izvršavanja ima linijski CSS (**Izvorni kod 3.2**) koji se koristi u HTML dokumentu dodavanjem *style* atributa određenoj oznaci unutar HTML-a.

```
<p style="background: blue; color: white;">Primjer linijskog CSS-a</p>
```

Izvorni kod 3.2 Primjer linijskog CSS-a

Prilikom korištenja vanjskog CSS dokumenta sadržaj je u potpunosti odvojen od oblikovanja te je na taj način jednostavnija kasnija manipulacija u slučaju potrebe za nekim promjenama u dizajnu i oblikovanju web stranice.

Treći način primjene CSS-a je takozvani unutrašnji. Unutar *<head>* oznake HTML dokumenta potrebno je dodati *<style>* oznaku sa atributom koji HTML prevoditelju označuje da je riječ o CSS-u. Unutar *<style>* oznake zatim se piše CSS kod kako je prikazano na sljedećem primjeru (**Izvorni kod 3.3**, pogledati stranicu 11).

¹² *** <http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=5145> – Strategy Analytics, Prosinac 2012.

¹³ *** <http://www.w3.org/Style/CSS/> - W3C

```
<head>
  <style type="text/css">
    body { background-color: red; }
  </style>
</head>
```

```
body { background-color: #AAAAAA; }
h1 {
  color: orange;
  text-align: center;
}
p {
  font-family: Arial;
  font-size: 16px;
}
```

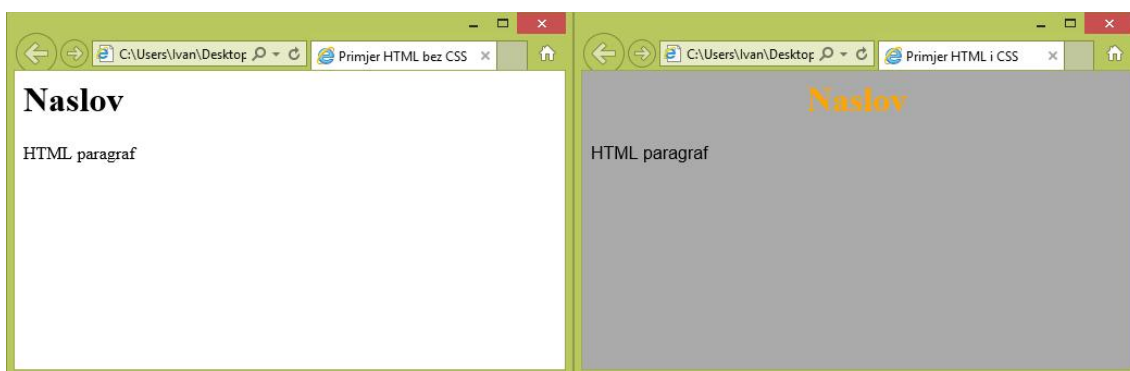
Izvorni kod 3.4 Jednostavna CSS dokument za oblikovanje HTML-a

Vanjski CSS dokument se može uključiti u HTML dokument dodavanjem jedne linije koda unutar *<head>* oznake (eng. *tag*) kako je prikazano na sljedećem izvornom kodu:

Izvorni kod 3.5 HTML dokument s uključenim vanjskim CSS dokumentom

```
<html>
<head>
  <title>Primjer HTML i CSS</title>
  <link rel="stylesheet" type="text/css" href="stil.css" />
</head>
<body>
  <h1>Naslov</h1>
  <p>HTML paragraf</p>
</body>
```

Slika 3.1 Usporedba prikaza HTML-a. S lijeve strane bez primjene CSS-a, a s desne strane uz primjenu CSS-a



Na slici je prikazana usporedba prikaza običnog HTML-a (**Izvorni kod 3.1**) u web pregledniku bez primjene CSS-a te uz primjenu CSS-a (**Izvorni kod 3.4**).

3.2.1. CSS 3

Prva skica CSS 3 standarda pojavila se još 1999. godine. Za razliku od CSS 2, koji je veliki skup definicija za različite značajke, CSS 3 je podijeljen u više odvojenih cjelina nazvanih moduli. Svaki od tih modula definira nove ili proširuje značajke definirane u CSS 2 standardu. Trenutno postoji preko 50-ak modula koji se razvijaju od World Wide Web konzorcija i radne grupe zadužene za razvoj CSS-a. Većina modula još nema svoju stabilnu verziju, ali se sve više upotrebljava na webu.

Stabilni moduli koje preporučuje W3C¹³:

- Media Queries – modul koji je zadužen za učitavanje određenog djela CSS-a ovisno o određenim parametrima kao što su rezolucija ekrana, tip uređaja i sl.
- Namespaces
- Selectors Level 3 – ovaj modul je zadužen za označavanje na koji element unutar HTML dokumenta će biti primijenjen određeni dio CSS-a

- Color – proširuje značajke iz CSS 2 verzije

Postoje i moduli za 2D i 3D transformacije, animacije, tekstualne efekte, govor i drugi. Neki od tih modula su u statusu CR (eng. *candidate recommendation*, kandidat za finalnu verziju) te se provode testiranja prije finalne verzije koja će imati preporuku World Wide Web konzorcija za korištenje.

3.3. JavaScript

Predstavljen je od strane Netscape Communications Corporation 1995. godine kao otvoreni, multiplatformski objektni skriptni jezik¹⁴, a standardiziran je 1996. godine. JavaScript je rođen iz potrebe koordiniranja HTML web stranice s ugrađenim sadržajima kao što su Java applete-i, ali danas se koristi i za obogaćivanje izgleda, dizajna te upotpunjavanje korisničkog doživljaja web stranica. Osim korištenja za web programiranje, JavaScript se koristi i u PDF preglednicima, dodacima na radnoj površini (eng. *widget*). JavaScript skriptni programski jezik također podržava i objektno orijentirano programiranje.

Primjer jednostavnog JavaScript koda (**Izvorni kod 3.6**):

Izvorni kod 3.6 Jednostavni JavaScript program

```
function hw() {  
  alert (Hello world!);  
}
```

Prikazani dio JavaScript koda (**Izvorni kod 3.6**) prilikom pozivanja funkcije *hw()* vraća poruku upozorenja (eng. *alert* - upozorenje) Hello World!

Iako je bio vrlo brzo prihvaćen neki profesionalni programeri nisu ga prihvatili sve do pojave Ajax-a (eng. *Asynchronous JavaScript and XML*) koji je omogućio slanje

¹⁴ Scott Duffy (2003). *How to do everything with JavaScript*, McGraw-Hill/Osborne, ISBN 0-07-222887-3

i primanje podataka sa web poslužitelja (eng. *server*) asinkronim načinom (u pozadini) bez utjecaja na prikaz i ponašanje same web stranice.

3.3.1. JSON

JSON (eng. *JavaScript Object Notation*) je jednostavan format za razmjenu podataka lako čitljiv ljudima te računalima. JSON je tekstualni format potpuno neovisan o programskom jeziku te je kao takav idealan za razmjenu podataka. Građen je na dvije strukture:¹⁵

- Kolekcija parova imena i pripadajućih vrijednosti (**Izvorni kod 3.7**)
- Poredana lista vrijednost (polje) (**Izvorni kod 3.8**)

```
{ "Ime": "Ivan" ,  
  "Prezime": "Ivić" ,  
  "godina" : 1993 }
```

Izvorni kod 3.7 Primjer JSON objekta

```
{ "popis": [  
  { "Ime": "Ivan" , "Prezime": "Ivić" , "godina" : 1993 } ,  
  { "Ime": "Ana" , "Prezime": "Anić" , "godina" : 2000 }  
]}
```

Izvorni kod 3.8 Primjer JSON polja

¹⁵ *** <http://www.json.org/>

4. RAZVOJNE OKOLINE ZA RAZVOJ „NATIVNIH“ MOBILNIH APLIKACIJA TEMELJENIH NA WEB STANDARDIMA

Razvoj aplikacije za više platformi je prilično zahtjevan jer svaka platforma koristi drugačiji programski jezik te je potreban velik broj razvijatelja. Sukladno tome povećava se i cijena izrade same aplikacije. Postoji i alternativno rješenje za razvoj „nativnih“ mobilnih aplikacija pomoću web tehnologija i besplatnih web servisa. Ovako razvijene aplikacije se još nazivaju i hibridne mobilne aplikacije.

Hibridna mobilna aplikacija je ona aplikacija koja radi unutar nativnog kontejnera te iskorištava mogućnosti ugrađenog web engine-a za prikaz lokalnog HTML-a i lokalno procesuiranje JavaScript-a. Takvim aplikacijama su dostupne funkcije uređaja poput akcelerometra, kamere i lokalne pohrane koje nisu dostupne u mobilnim web aplikacijama. Hibridne aplikacije koriste tzv. *Web view* komponentu (kod Androida je to *WebView*, kod iOS-a je to *UIWebView*) kako bi prikazale HTML i procesuirale JavaScript preko cijelog ekrana.¹⁶

Neki od poznatijih razvojnih okolina za razvoj multiplatformskih aplikacija temeljenih na web tehnologijama:

- PhoneGap
- Adobe AIR
- IBM Worklight
- Oracle ADF Mobile Framework

Navedene razvojne okoline, osim Adobe AIR-a, koriste PhoneGap za pakiranje aplikacija u nativne instalacijske pakete uz neke svoje dodatke, dok Adobe AIR koristi vlastitu razvojnu okolinu. Ove razvojne okoline nam omogućuju izradu mobilnih aplikacija baziranih na web tehnologijama te njihovo pakiranje u nativne instalacijske pakete.

Također postoje različiti razvojne okoline za razvoj mobilnih korisničkih sučelja koji olakšavaju razvoj mobilnih aplikacija baziranih na web tehnologijama. Svaka od tih

¹⁶ *** <http://icenum.com/community/blog/icenum-team-blog/2012/06/14/what-is-a-hybrid-mobile-app->
- Doug Seven, Lipanj 2012.

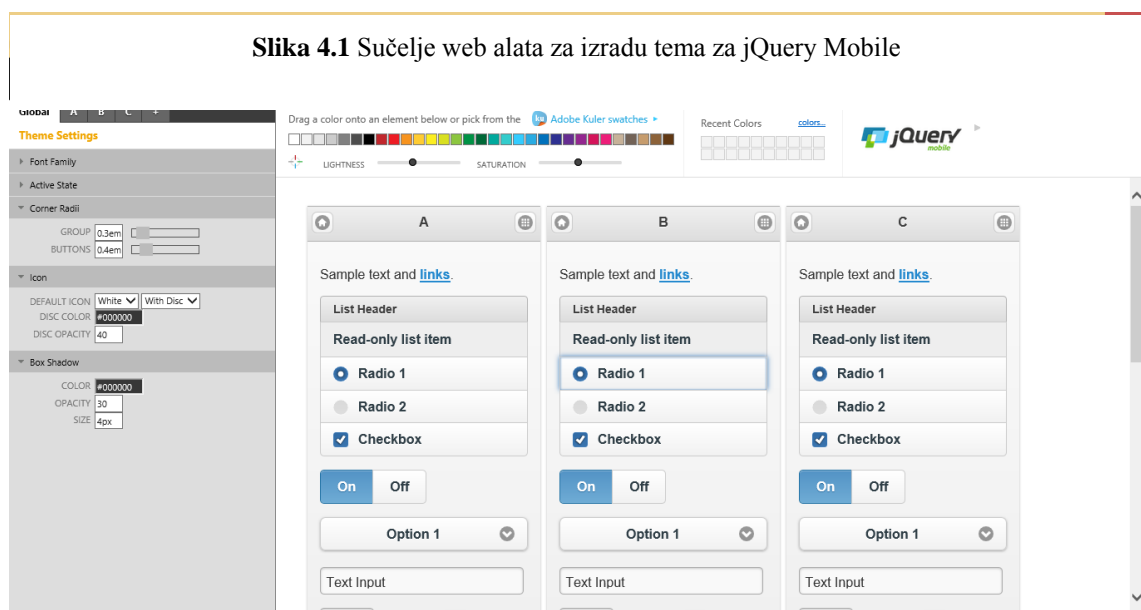
razvojnih okolina ima implementirane različite mogućnosti (fiksirana zaglavlja i podnožja, različite liste, teme i sl) koje nije potrebno nanovo programirati već ih se može pozvati preko specifičnih API-ja što uvelike ubrzava i olakšava razvoj hibridnih aplikacija. Neke od poznatijih razvojnih okolina razvoj mobilnih korisničkih sučelja su:

- jQueryMobile,
- Sencha Touch,
- Wink Toolkit, iUI

4.1. jQuery Mobile

Optimizirana UI (eng. user interface, korisničko sučelje) razvojna okolina za uređaje s zaslonima osjetljivim na dodir. Razvojna okolina je jedinstvenog korisničkog sučelja na svim mobilni platformama. jQuery Mobile je baziran na jQuery i jQuery UI razvojnim okolinama koji su već duže vrijeme jedni od najkorištenijih za razvoj desktop web stranica i web aplikacija.

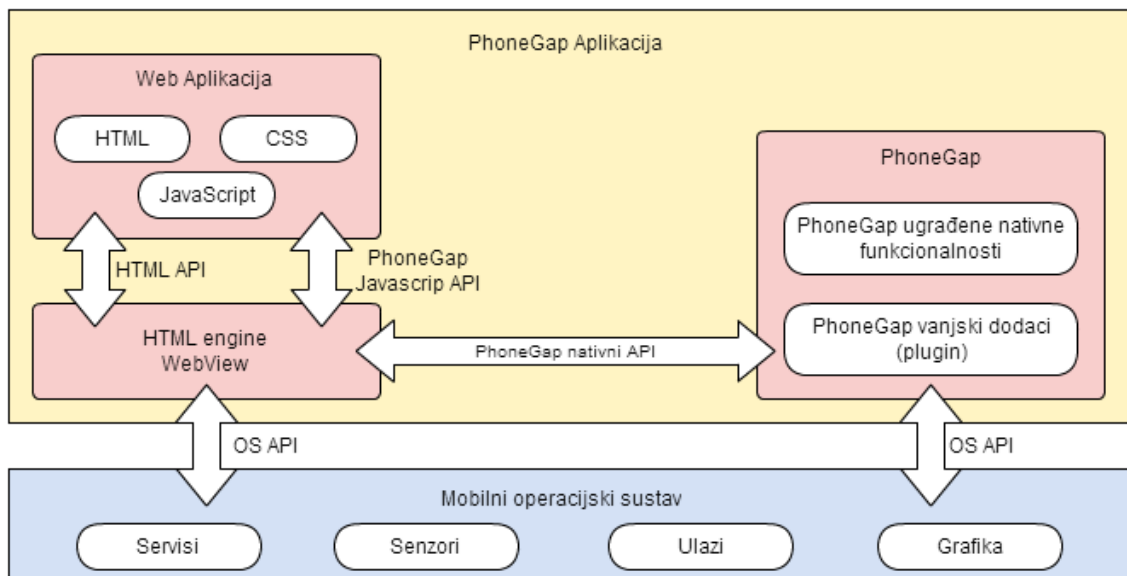
Odlika jQuery mobile razvojne okoline je dobra podrška te konstantno izdavanje novih verzija s novim mogućnostima te poboljšanjima već postojećih, jednostavna izrada dizajna pomoću web alata Theme Roller za jQuery mobile (**Slika 4.1**).



Sve mogućnosti ove razvojne okoline su u potpunosti funkcionalne na Apple iOS 3.2, Android 2.1, Windows Phone 7, Blackberry 6 te novijim verzijama tih operacijskih sustava.

4.2. PhoneGap

PhoneGap je razvojna okolina otvorenog koda za izradu multiplatformskih hibridnih mobilnih aplikacija uz pomoć HTML, CSS i JavaScript-a. PhoneGap je JavaScript apstrakcijski sloj iznad nativnih API-ja koji omogućava pristup određenim mogućnostima uređaja koje nisu dostupne u standardnim mobilnim web aplikacijama (kamera, akcelerometar i sl.). PhoneGap omogućuje komunikaciju između nativnih funkcionalnosti te izrađene mobilne aplikacije bazirane na web standardima.



Slika 4.2 Prikaz PhoneGap arhitekture aplikacija

Na slici (**Slika 4.2**) je prikazana arhitektura PhoneGap hibridne mobilne aplikacije te komunikacija između dijelova aplikacije. PhoneGap razvojno okruženje se sastoji od dva dijela:

- od nativnog dijela koda koji je zadužen za komunikaciju sa mobilnim operacijskim sustavom na kojem je instalirana aplikacija i
- od JavaScript koda koji je zadužen za komunikaciju između izrađene mobilne web aplikacije te PhoneGap nativnog dijela koda.

Tablica 4.1 Podržane nativne mogućnosti PhoneGap razvojne okoline

	Android	iPhone / iPhone 3G	iPhone 3GS i noviji	Blackberry OS 5.0 i noviji	Windows Phone 7 i noviji
Akcelerometar	+	+	+	+	+
Kamera	+	+	+	+	+
Kompas	+	-	+	-	+
Kontakti	+	+	+	+	+
Datotečni sustav	+	+	+	+	+
Geolokacija	+	+	+	+	+
Media	+	+	+	-	+
Mreže	+	+	+	+	+
Obavijesti – upozorenje / zvuk / vibracija	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+
Pohrana	+	+	+	+	+

U tablici (**Tablica 4.1**) je prikazana podrška nativnih funkcionalnosti određenih mobilnih uređaja te pripadajućih mobilnih operacijskih sustava koje omogućuje PhoneGap razvojna okolina s ugrađenim nativnim dodacima.

Pozivanje nativne funkcionalnosti

Sve nativne funkcionalnosti ugrađene u PhoneGap pozivaju se pomoću definiranih API-ja kroz JavaScript kod. Primjer pozivanje nativne funkcionalnosti snimanja fotografije te prikaza snimljene fotografije unutar aplikacije je prikazana na

izvornom kodu (**Izvorni kod 4.1**). Snimljena fotografija se učitava unutar HTML elementa ``.

Izvorni kod 4.1 Isječak JavaScript koda za snimanje i prikaz slike

```
function onSuccess(imageURI) {
    var image = document.getElementById('myImage');
    image.src = imageURI;
}
function onFail(message) {
    alert('Pogreška: ' + message);
}
```

4.2.1. PhoneGap nativni plugin-ovi

Osim ugrađenih funkcionalnosti (**Tablica 4.1**) postoji mogućnost izrade nativnih pluginova. PhoneGap nativni plugin omogućuju izradu vlastitih prilagođenih klasa i odgovarajućih JavaScript sučelja za korištenje u PhoneGap aplikacijama. Na ovaj način se može izraditi plugin za bilo koju nativnu funkcionalnost koja nije podržana od strane PhoneGap-a.

Za izradu PhoneGap plugin-a je potrebno sljedeće:

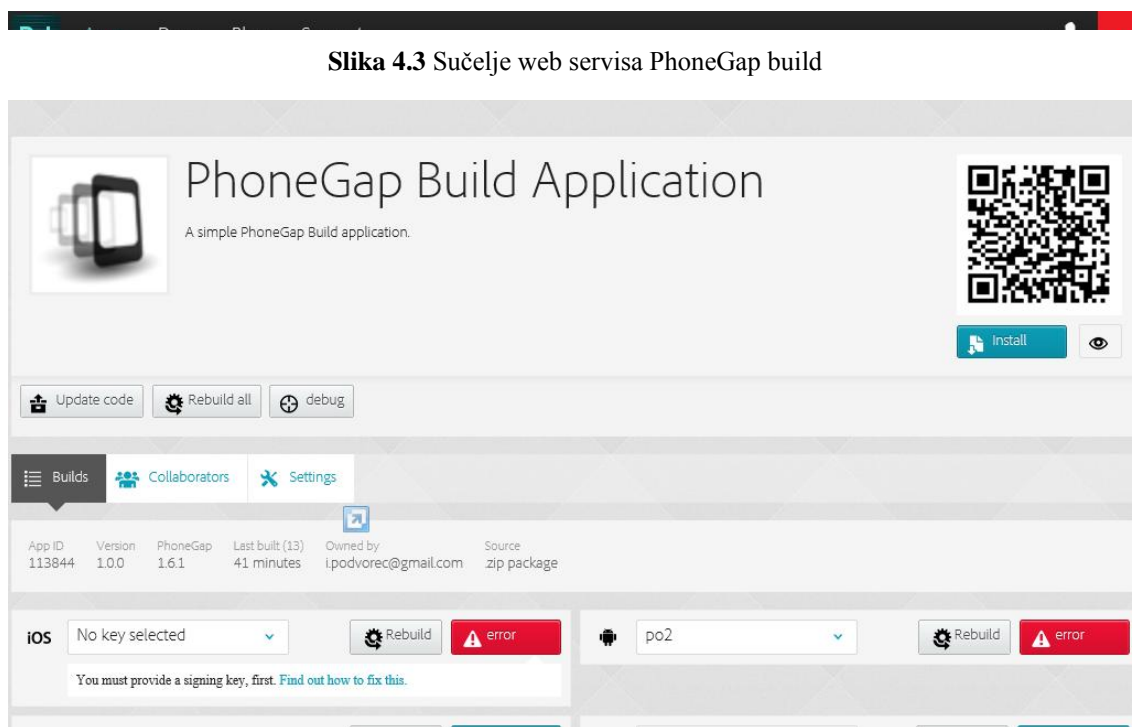
- Napisati JavaScript sučelje koje će koristiti PhoneGap API za komunikaciju s nativnim kodom.
- Registrirati plugin s nativnim projektom, tj. JavaScript sučelje s nativnim kodom
- Napisati nativni kod kojem će PhoneGap prosljeđivati primljene naredbe

Nativne plugin-ove je potrebno programirati na nativnom jeziku specifičnom za platformu, odnosno ukoliko se želi izraditi aplikacija za tri različite platforme potrebno je izraditi tri plugin-a. Također postoje već izrađeni većinom besplatni plugin-ovi koje je moguće pronaći na internetu te ih preuzeti i uključiti u aplikacije.

4.2.2. PhoneGap build

PhoneGap build je web servis za za pakiranje izrađenih mobilnih web aplikacija u nativne instalacijske pakete koji se mogu instalirati na odgovarajuće mobilne platforme. PhoneGap build trenutno podržava pakiranje aplikacija u nativne instalacijske pakete za Apple iOS, Android, Windows Phone, BlackBerry, Symbian te WebOS.

Da bi se mogao izraditi nativni instalacijski paket za određeni mobilni operacijski sustav potrebno je izrađenu mobilnu web aplikaciju zapakirati u `.zip` datoteku te je uploadati na web servis (<http://build.phonegap.com>). Ta `.zip` datoteka smije sadržavati samo HTML, CSS, JavaScript datoteke te resurse (slike korištene u aplikaciji, PDF dokumente i sl), ali ne smije sadržavati nikakav nativni kod. Za iOS, Android i BlackBerry potrebno je imati ključeve za potpisivanje aplikacije da bi se aplikacije mogle zapakirati (**Slika 4.3**).



4.3. Prednosti i nedostaci hibridnih mobilnih aplikacija

Za razliku od mobilnih web aplikacija kod kojih su sve datoteke spremljene na web poslužitelju, hibridne aplikacije imaju sve datoteke (HTML, CSS, JavaScript i

slike) spremljene lokalno te nije potrebna veza na Internet. S korisničke strane hibridna mobilna aplikacija se ne razlikuje od „normalne“ aplikacije jer se ponaša na isti način kao i „normalna“ aplikacija.

Tablica 4.2 Usporedba mogućnosti web, hibridnih i nativnih mobilnih aplikacija

	W e b a p l i k a c i j a	H i b r i d n a a p l i k a c i j a	N a t i v n a a p l i k a c i j a
Cij en a r a z v i j a n j a	N i s k a	N i s k a	V i s o k a
Vr i j e m e r a z v i j a n j a	K r a t k o	K r a t k o	D u g o
Pr e n o s i v o s t n a d r u g e p l a t f o r m e	V r l o v e l i k a	V r l o v e l i k a	M a l a / n i k a k v a
Br z i n a r a d a	S r e d n j a d o b	S r e d n j a d o b	V r l o b r z a

		rz a	
Nativne funkcionalnosti	Nema	Sve	Sve
Dostupnost aplikacije u duća ni ma aplikacija	Nije dostupna	Dostupna	Dostupna

Iz tablice (**Tablica 4.2**) se mogu pročitati ograničenja u mogućnostima različitih vrsta mobilnih aplikacija.

Prednosti

Hibridne aplikacije omogućuju višestruko iskorištenje koda kroz različite mobilne platforme. To znači da će izrađena aplikacija raditi isto na svim mobilnim platformama koje podržava razvojna okolina (PhoneGap). Ta ista razvojna okolina se brine o specifičnim razlikama između mobilnih platforma. S JavaScript-om je većina razvijatelja već upoznata te se može reći da je jednostavnije razviti hibridnu mobilnu aplikaciju nego krenuti u učenje drugih programskih jezika za razvoj nativnih aplikacija.¹⁶ Samo korisničko sučelje te ispravak pogrešaka u kodu moguće je obaviti u standardnim desktop web preglednicima s određenim besplatnim alatima od kojih su neki ugrađeni u same preglednike.

Nedostaci

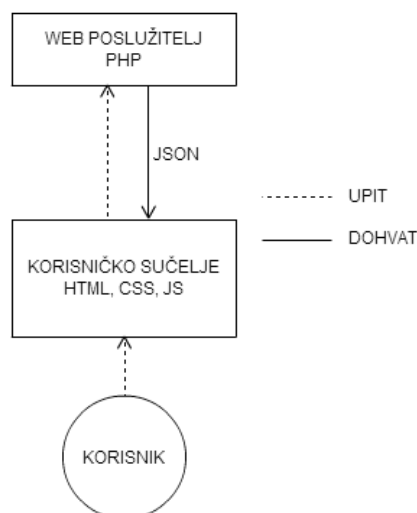
Jedan od glavnih nedostataka hibridnih aplikacija je osjetljivost korisničkog sučelja odnosno njegova brzina što je posljedica dodatnih slojeva apstrakcije (*web-to-native* apstrakcijski sloj). Drugi nedostatak je ograničena podrška za native funkcionalnosti što ovisi o korištenoj razvojnoj okolini. Ako nam je potrebna neka određena nativna funkcionalnost koja nije podržana od strane same razvojne okoline potrebno je programirati nativni plugin.

5. PRAKTIČNI RAZVOJ APLIKACIJE

Praktični dio ovoga rada je izrada hibridne multi platformne aplikacije za Android operacijski sustav bazirane na web standardima koja može uz minimalne promjene biti prilagođena za druge mobilne platforme (iOS, Blackberry, Windows Phone). Za razvoj aplikacije korištene su web tehnologije HTML, CSS te JavaScript i besplatni web servis PhoneGap Build za izradu nativnih instalacija.

PhoneGap razvojno okruženja za razvoj hibridnih mobilnih aplikacije je odabrano zbog jednostavnosti korištenja te velike online zajednice korisnika. Isto tako PhoneGap omogućuje jednostavnu izradu nativnih instalacija hibridnih mobilnih aplikacija web servisom PhoneGap Build (<http://build.phonegap.com>). Za izradu jednostavnih aplikacija nije potrebno znanje iz nativnih programskih jezika. S druge strane ukoliko su nam potrebne neka nativne funkcionalnosti postoji ugrađene osnovne nativne funkcionalnosti.

Komunikacijski protokol se bazira na više programskih jezika i procedura kako bi se postigla zadovoljavajuća brzina te postiglo maksimalno korisničko iskustvo. Ako se odlučimo za pokretanje bilo koje navedene funkcije aplikacije pokreće se niz programskih procedura u komunikacijskom protokolu koji je prikazan sljedećom slikom:



Slika 5.1 Shema rada aplikacije

5.1. Grafičko sučelje aplikacije

Grafičko sučelje aplikacije je bazirano na jQuery Mobile okruženju za razvoj korisničkih sučelja namijenjeno pametnim uređajima (pametni telefoni, tableti) uz minimalne preinake.

Slika 5.2 Prikaz ikona aplikacije

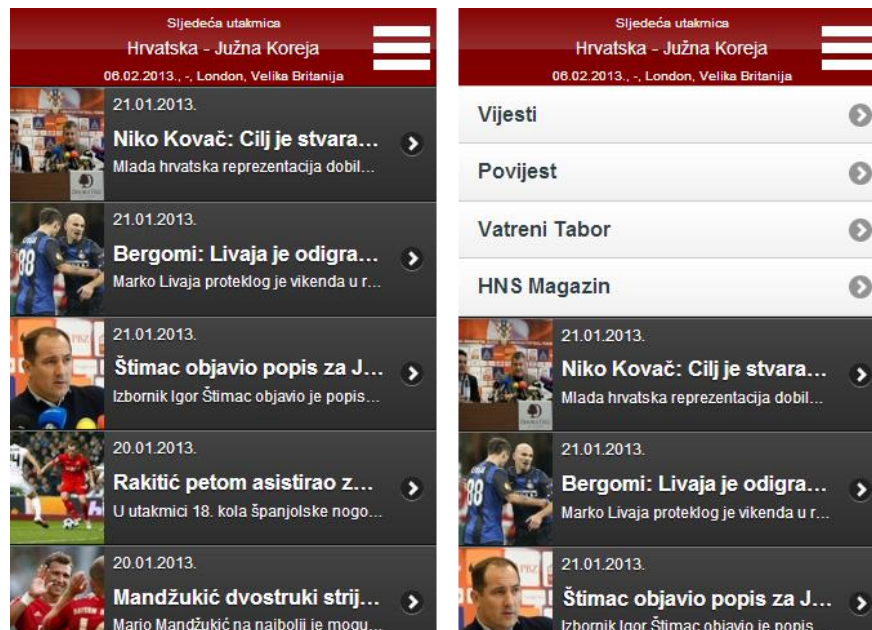


Dodirom (klikom) na ikonu aplikacije (**Slika 5.2**) u sučelju operacijskog sustava otvara se početni ekran učitavanja aplikacije (eng. *splashscreen*). (**Slika 5.3**)

Slika 5.3 Ekran učitavanja aplikacije



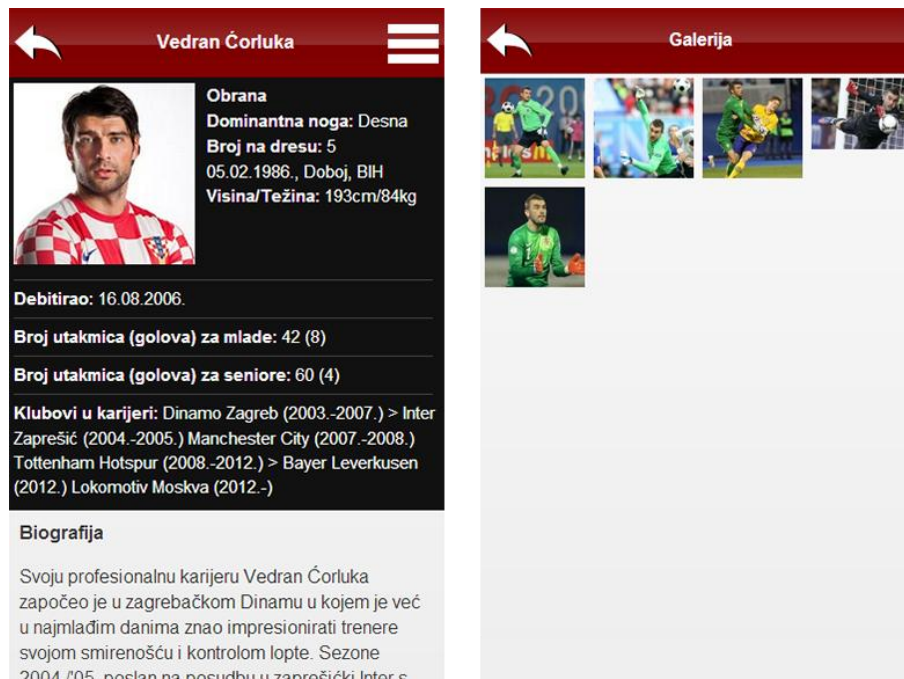
Nakon učitavanja aplikacije otvara se početni ekrana s vijestima. U zaglavlju grafičkog sučelja aplikacije se nalazi informacija o sljedećoj utakmici. S desne strane



zaglavlja se nalazi gumb za otvaranje izbornika. (Slika 5.4)

Slika 5.4 Prikaz ekrana uređaju u portretnom položaju. Slika desno bez otvorenog izbornika, slika lijevo s otvorenim izbornikom

Na lijevoj strani zaglavlja na svim stranicama (osim početne te galerije) se nalazi



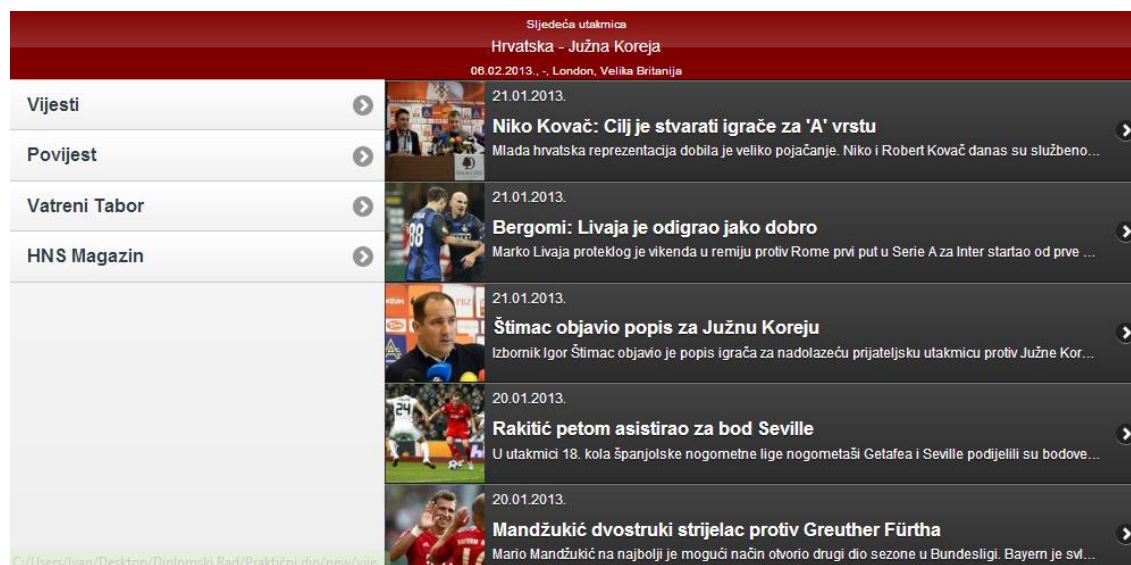
Slika 5.5 Prikaz zaglavlja i galerije gumb za povratak na prethodnu stranica. (**Slika 5.5**).

Dodirom (klikom) gumba galerija unutar biografije igrača otvara se galerija sličica koje se klikom otvaraju na novoj stranici preko cijelog ekrana. (**Slika 5.6**) Tada se slike mogu izmjenjivati potezima klizanja (eng. *swipe*) preko ekrana.



Slika 5.6 Prikaz galerije preko cijelog ekrana

Ukoliko se uređaj na kojem je aplikacija instalirana zarotira iz portretnog položaja u pejzažni položaj grafičko sučelje aplikacije se prilagođava na taj način da je izbornik tada otvoren u lijevoj strani ekrana te više nema gumba za otvaranje i zatvaranje izbornika u zaglavlju. (**Slika 5.7**) Osim zakretanja ekrana u pejzažni položaj



da bi se promijenilo grafičko sučelje aplikacije treba biti zadovoljena i rezolucija ekrana.

Slika 5.7 Prikaz sučelja aplikacije u pejzažnom položaju

Sučelje aplikacije je prilagodljivo rezoluciji te orijentaciji ekrana uređaja na kojima se aplikacija koristi. Isto je izvedeno pomoću CSS3 modula medijski upita (eng. *media queries*).

CSS 3 medijski upiti omogućuju korištenje različitih CSS stilova na temelju karakteristika uređaja, orijentaciji i rezoluciji ekrana. Aplikacija koristi 4 medijska upita (**Izvorni kod 5.1**):

1. upit - odnosi se na sve rezolucije prikaznog polja (eng. *viewport*) manjeg od ili jednakog 480px (eng. *pixel* – točka) te za sve orijentacije uređaja (portretnu i pejzažnu). Prikazno polje je dio ekrana na kojem je renderirana stranica.

2. upit se sastoji od tri dijela od kojih svaki mora biti zadovoljen da bi se primijenio CSS stil koji se nalazi unutar tog medijskog upita. Odnosi se na sve rezolucije prikaznog polja od 481px do 960px te pejzažnu orijentaciju uređaja.
3. upit se također sastoji od tri dijela koji trebaju biti zadovoljeni da bi se primijenio CSS stil koji se nalazi unutar tog medijskog upita. Ovaj upit se odnosi na sve rezolucije prikaznog polja od 481px do 960px te za portretnu orijentaciju uređaja.
4. upit se odnosi se na sve rezolucije prikaznog polja koje je minimalne širine 961px i šire te na portretnu i pejzažnu orijentaciju uređaja.

Izvorni kod 5.1 CSS3 medijski upiti za prilagodbu izgleda sučelja aplikacije

```
.izbornik { display: none; padding: 0; margin-bottom: 50px; }
.sadrzaj { position: absolute; margin-top: 0px; text-align: center; left: 0; right: 15px; }
.sadrzaj-center { left: 15px; }
.tabor img { max-width: 42%; }
.tabor_pozadina_info_img { position: relative; height: 165px; }
}

@media screen and (min-width: 481px) and (max-width: 960px) and (orientation:landscape) {
    .izbornik { position: fixed; width: 180px; display: block !important; }
    .sadrzaj { position: absolute; left: 225px; right: 15px; max-width: 100%; text-align: center; }
    .header-button { display: none !important; }
    .tabor img { max-width: 42%; }
    .tabor_pozadina_info_img { position: relative; height: 165px; }
}

@media screen and (min-width: 481px) and (max-width: 960px) and (orientation:portrait) {
    .izbornik { position: fixed; width: 180px; }
    .sadrzaj { position: absolute; left: 225px; right: 15px; max-width: 100%; text-align: center; }
    .header-button { display: none !important; }
    .tabor img { max-width: 42%; }
    .tabor_pozadina_info_img { position: relative; height: 165px; }
}

@media screen and (min-width: 961px){
    .izbornik { position: fixed; width: 300px; padding: 0; margin: 0; display: block !important; }
    .sadrzaj { position: absolute; left: 345px; right: 10px; max-width: 100%; text-align: center; }
    .header-button { display: none !important; }
}
}
```

5.2. Opis programskog dijela aplikacije

Za programski dio aplikacije je zadužen u JavaScriptu uz jQuery mobile programsko razvojno okruženje. JavaScript se koristi za dohvat te prikaz podataka unutar grafičkog sučelja aplikacije te za neke dijelove funkcioniranja grafičkog sučelja. Prije bilo kakvog korištenja JavaScript koda treba provjeriti pripremljenost samog uređaja na kojem je instalirana aplikacija, tj. da li se PhoneGap u potpunosti učitao. U suprotnom bi moglo doći do sporijeg funkcioniranja ili čak do nepokretanja same aplikacije. Kompletan JavaScript kod se zatim može koristiti unutar funkcije *onDeviceReady* (**Izvorni kod 5.2**).

```
document.addEventListener("deviceready", onDeviceReady, false);
function onDeviceReady() {
    // Programski dio aplikacije
}
```

Izvorni kod 5.2 Isječak koda za provjeru pripremljenosti uređaja

Programski dio aplikacije se može podijeliti u dvije cjeline čijom sinergijom se dobiva potpuno funkcionalna aplikacija:

- HTML dio
- JavaScript dio koji se može podijeliti na:
 - Dohvat podataka s udaljenog poslužitelja
 - Generiranje HTML dokumenta s dohvaćenim podacima
 - Galerija slika

Sadržaj aplikacije je strukturiran unutar HTML dokumenta. Struktura HTML dokumenta prikazana je na sljedećem izvornom kodu:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Diplomski</title>
<link rel="stylesheet" href="js/jquery.mobile-1.2.0.css" />
<script src="js/jquery-1.8.3.min.js"></script>
<script src="js/jquery.mobile-1.2.0.min.js"></script>
<script src="phonegap.js"></script>
<script src="js/vijesti.js"></script>
<script src="js/dohvat.js"></script>
<link rel="stylesheet" href="css/stil.css" />
</head>
<body>
<div data-role="page" id="vijesti">
  <div data-role="header" data-position="fixed" class="header
header-naslovna" data-tap-toggle="false" data-theme="s">
    <a href="#" class="show_menu"></a>
    <div class="header-txt-slj"><script>sljedeca();</script></div>
    <a href="#" class="show_menu header-button" data-icon="menu-my"
data-role="button" data-iconpos="notext">Menu</a>
  </div>
  <div data-role="content" class="content">
    <div class="izbornik">
      <ul data-role="listview">
        <li><a href="#vijesti" class="izbornik-
vijest">Vijesti</a></li>
        <li><a href="#povijest" class="izbornik-
povijest">Povijest</a></li>
        <li><a href="#tabor" class="izbornik-tabor">Watreni
Tabor</a></li>
        <li><a href="#hnsmag" class="izbornik-hnsmag">HNS
Magazin</a></li>
      </ul>
    </div>
  </div>
</div>
```


Izvorni kod 5.3 Struktura HTML dokumenta

```
<ul data-role="listview" class="svevijesti" data-theme="a">
  <li></li>
</ul>
</div>
</div>
</div>
</body>
</html>
```

Na prethodnom izvornom kodu (**Izvorni kod 5.3**) je prikazana struktura HTML dokumenta koji sadržava osnovne elemente koje zahtijeva jQuery mobile razvojno okruženje. U *head* elementu potrebno je pozvati JavaScript skripte te CSS stilske datoteke neophodne za ispravno funkcioniranje jQuery mobile razvojnog okruženja. Osnovni *div* element mora biti označen atributom *data-role="page"* jer u suprotnom neće biti dostupne sve funkcionalnosti jQuery mobile razvojnog okruženja. Unutar tih elemenata mogu se koristiti bilo koji HTML elementi, liste, tablice, paragrafi i td. Slična struktura se koristi na svim stranicama aplikacije dok se sadržaj koji dohvaćamo sa udaljenog web poslužitelja unutar HTML-a učitava pomoću JavaScript-a.

Izvorni kod 5.4 Isječak koda zadužen za prikaz i sakrivanje izbornika

```
$('.show_menu').click(function() {
$(".izbornik").slideToggle(); });
$('.izbornik-vijest').click(function() {
$(".izbornik").slideToggle(); });
$('.izbornik-povijest').click(function() {
$(".izbornik").slideToggle(); });
$('.izbornik-tabor').click(function() {
$(".izbornik").slideToggle(); });
$('.izbornik-hnsmag').click(function() {
$(".izbornik").slideToggle();});
});
```

Funkcija za prikaz i sakrivanje izbornika je učitana prilikom učitavanja aplikacije, uvijek je u pripravnim stanju jer očekuje dodir označenog dijela u grafičkom sučelju aplikacije preko strukture HTML dokumenta te se tek onda izvršava prikaz ukoliko je izbornik sakriven, odnosno sakrivanje izbornika ukoliko je prikazan. **(Izvorni kod 5.4)**

Dohvat podataka

Za prijenos podataka s web poslužitelja koristi se JSON format zapisa podataka. Na sljedećem izvornom kodu je prikazan primjer generiranog JSON dokumenta od strane web poslužitelja **(Izvorni kod 5.5)**. Za generiranje JSON-a zadužena je PHP

```
[{
  id_vijest: "647",
  naslov: "Radošević u Napoliju!",
  kratki_text: "Igrač Hajduka i odnedavni reprezentativac Josip Radošević u ovim trenucima sređuje zadnje formalnosti prije potpisa pe...",
  text: "Igrač Hajduka i odnedavni reprezentativac Josip Radošević u ovim trenucima sređuje zadnje formalnosti prije potpisa petogodišnjeg ugovora s talijanskim prvoligašem Napolijem. Pregovori ... a tek kada sezona završi preseliti u grad podno Vezuva.",
  poveznica: false,
  thumb: "uploads/thumb-radosevic_vatreni13590433506.jpg",
  slika: "http://vatreni.info/backend/uploads/13590433497-radosevic_vatreni.jpg",
  video_url: false,
  datum_unosa: "24.01.2013.",
  autor: "Igor Igor"
},
...]
```

Izvorni kod 5.5 Primjer JSON podataka koje isporučuje web poslužitelj

skripta.

U programskom dijelu aplikacije dohvaćaju se JSON podaci koje je generirao web poslužitelj uz pomoć funkcije te se prikazuju unutar grafičkog sučelja aplikacije **(Izvorni kod 5.6)**.

```

var url = "http://vatreni.info/mobile/getData_native.php?broj=";
var urlVijest =
"http://vatreni.info/mobile/getData_native.php?id_vijest=";
var img_prefix = "vatreni.info/backend/";
var broj_vijesti = 10;

$.getJSON(url+broj_vijesti,
function (data) {
    $.each(data, function (i, item) {
        var thumb_url = item.thumb;
        if (thumb_url.charAt(0) == "/") thumb_url =
thumb_url.substr(1);
        if (thumb_url.charAt(thumb_url.length - 1) == "/")
thumb_url = thumb_url.substr(0, thumb_url.length - 1);
        $('sveVijesti').append('<li><a
href="vijest.html?id='+item.id_vijest+'" data-ajax="false"
id='+item.id_vijest+'><p>'+item.datum_unosa+'</p><h
1>' + item.naslov + '</h1><p>'+ item.kratki_text + '</p></a></li>');
    });

    $(".sveVijesti").append('<li class="jos"><a id="10" href="#"
onClick="josVijesti(this.id); obrisiLink();">Učitaj još
vijesti</a></li>');

    $('sveVijesti').listview('refresh');
});

function josVijesti(clicked) {
var e1 = document.getElementById(clicked).id;
e1 = parseInt(e1) + 10;
$.getJSON(url + e1,
function (data) {
    $('sveVijesti').html(' ');
    $.each(data, function (i, item) {
        var thumb_url = item.thumb;
        if (thumb_url.charAt(0) == "/") thumb_url =
thumb_url.substr(1);
        if (thumb_url.charAt(thumb_url.length - 1) == "/")
thumb_url = thumb_url.substr(0, thumb_url.length - 1);
        $('sveVijesti').append('<li><a

```

Izvorni kod 5.6 Isječak koda za dohvat podataka te prikaz istih unutar liste u grafičkom sučelju

```
id='+item.id_vijest+'><p>'+item.datum_unosa+'</p><h  
1>' + item.naslov + '</h1><p>'+ item.kratki_text + '</p></a></li>');  
  
    });  
    $('svevijesti').append('<li class="jos"><a href="#" id='+el+'  
onClick="josvijesti(this.id); obrisiLink();">Učitaj još  
vijesti</a></li></ul>');  
    $('svevijesti').listview('refresh');  
});  
}
```

U prvom dohvat preuzima se 10 najnovijih vijesti te se učitavaju u listu zajedno sa pripadajućim identifikacijskim brojevima (id). Ti identifikacijski brojevi služe za raspoznavanje određene vijesti kad se želi prikazati opširnija vijest. Nakon očitanih

```
function dohvatJedneVijesti() {
    $('.page').live('pageshow', function(event) {
        var id = getUrlVars()["id"];
        $.getJSON(urlVijest+id, prikaziVijest);
    });
}

function prikaziVijest(data) {
    $.each(data, function (i, item) {
        $('.header-txt').append(item.naslov);
        $(".vijest").append('<div class="vijest_slika_pozadina"></div><p
class="vijest_objavljeno">Objavljeno: '+item.datum_unosa+' </p><hr/><p
class="vijest_tekst"> '+ item.text +'</p>'); });
        $('.page').show().trigger('updateLayout');
    }
}

function getUrlVars() {
```

zadnjih deset vijesti u listu se učitava i gumb za dohvat sljedećih 10 vijesti.

Izvorni kod 5.7 Isječak koda za dohvat i prikaz opširnije vijesti

```
window.location.href.split(window.location.href.indexOf('?') + 1).split('&');
    for(var i = 0; i < hashes.length; i++)
    {
        hash = hashes[i].split('=');
        vars.push(hash[0]);
        vars[hash[0]] = hash[1];
    }
    return vars;
}
```

Za dohvat cijele vijesti prvo je potrebno prilikom odabira vijesti proslijediti identifikacijski broj vijesti, a zatim istu proslijeđenu varijablu preuzeti kako bi se mogla dohvatiti željena vijest. Nakon što se izdvoji proslijeđena varijabla pokreće se funkcija za dohvat podataka. Nakon dohvata i preuzimanje željene vijesti ista vijest se prikazuje unutar grafičkog sučelja aplikacije. (**Izvorni kod 5.7**). Slična procedura se izvršava prilikom dohvata podataka za ostale dijelove aplikacije.

Galerija slika

Prvo je potrebno dohvatiti podatke o slikama, odnosno web lokacije umanjenih verzija slika te slika pune rezolucije (**Izvorni kod 5.8**). Nakon dohvata podataka generira se dio HTML dokumenta u kojeg se umeću dohvaćene web lokacije kako bi se slike mogle prikazati unutar grafičkog sučelja aplikacije.

```
function dohvatslika () {
    var id = getUrlVars()["id"];
    var vrsta = getUrlVars()["vrsta"];

    $.getJSON('http://vatreni.info/mobile/getDataTabor_native.php?odabir='+vrsta+'&id='+id+'&galerija=slike',
    function (data) {
```

Izvorni kod 5.8 Primjer dohvat podataka za galeriju, generiranje HTML-a i pozivanje funkcije galerije

```
href="http://www.vatreni.info/backend/'+item.slika+'"></a></li>');  
    });  
    $('#galerija_grid').show().trigger('updateLayout');  
    $('#galerija_grid').slideGallery(); //poziv galerije  
    });  
});
```

Nakon uspješno generiranog HTML dijela galerije potrebno je pozvati funkciju galerije (**Izvorni kod 5.9**). Funkcija galerije omogućuje prikaz slika pune rezolucije preko cijelog ekrana mobilnog uređaja te jednostavnu navigaciju kroz slike učitane u galeriju. Prvo je potrebno iz generiranog HTML dokumenta dohvatiti web lokacije slika pune rezolucije. Zatim se generira HTML dio koda unutar kojeg se učitava slika pune rezolucije te se omogućuje navigacija kroz galeriju slika uz pomoć klizanja prstom po ekranu. Dohvaćene slike se učitavaju u memoriju kako bi se što brže prikazale u galeriji.

```
(function ($) {  
    $.fn.slideGallery = function () {  
        var load_img = 'js/images/ajax-loader.gif';  
        var i = $("<img />").attr('src',load_img).load() //  
preloadanje ajax-loader.gif  
        var g = $(this);  
        $(this).children().each(function (index, element) {  
            var $t = $(this).find('a');  
            $t.attr('data-href',  
$t.attr('href')).removeAttr('href');  
            }).click(function (e) {  
                $(g).children('[selector]').removeAttr('selector');  
                //uklanjanje markera s prethodno prikazane slike  
                $(this).attr('selector', 'yes'); //markiranje kliknute  
slike (thumb-a)  
                var velika_slika = $(this).find('a').attr('data-href');
```



```

// url velika slika
    if (velika_slika !== '') {
        if (!$('#slidegallery').length) {
            $('body').append('<div data-role="page" id="slidegallery" data-
            theme="c" data-title=""><div data-role="content"
            id="slidecontent"><div id="slikaslide"></div><div
            id="imageslide"></div></div></div>');
            $.mobile.initializePage();
            $('#slidegallery').live('pagehide', function ()
            {
                $(this).remove()
            }); //brisanje stranice za otvaranje velike
            slike nakon izlaza (pritisak na back gumb)
            $('#imageslide').click(function (e) {
                }).swipeleft(function () { //prethodna slika
                    if
            ($$(g).children('[selector]').prev().length) {
                $(g).children('[selector]').prev().click()
            } else {
                $(g).children(':last-child').click()
            }
            }).swiperight(function () { //sljedeca slika
                if
            ($$(g).children('[selector]').next().length) {
                $(g).children('[selector]').next().click()
            } else {
                $(g).children(':first-child').click()
            }
            }) }
            $('#slikaslide').fadeOut(400, function () {
                $('#slikaslide').html('') // učitavanje slike
                $("<img />").attr('src',
            velika_slika).load(function (e) {
                $('#slikaslide').css({'background-image':
            'url(' + velika_slika + ')'}).html('').fadeIn(400)
            });
        });
    });

```

Izvorni kod 5.9 Dio koda zadužen za funkcioniranje galerije slika

```
    }}}})  
})(jQuery);
```

5.2.1. PhoneGap Build konfiguracija aplikacije, priprema za izradu nativnih instalacija

Kako bi se mogli podesiti meta podaci aplikacije za PhoneGap Build web servis potrebno je izraditi konfiguracijsku XML (*config.xml*) datoteku sa željenim podacima. Unutar konfiguracijske datoteke navedeno je ime aplikacije, opis, verzija te jedinstveno identifikacijsko ime aplikacije (id).

```
<?xml version="1.0" encoding="UTF-8" ?>  
  <widget xmlns = "http://www.w3.org/ns/widgets"  
    xmlns:gap = "http://phonegap.com/ns/1.0"  
    id          = "com.po2.diplomski"  
    versionCode="10"  
    version    = "1.0.0">  
  
    <name>Diplomski</name>  
    <icon src="icon.png" />  
  
    <icon src="slike/mdpi.png" gap:platform="android"  
gap:density="mdpi" />  
    <icon src="slike/hdpi.png" gap:platform="android"  
gap:density="hdpi" />  
    <icon src="slike/xhdpi.png" gap:platform="android"  
gap:density="xhdpi" />  
    <icon src="slike/ios57.png" width="57" height="57" />  
    <icon src="slike/hdpi.png" gap:platform="ios" width="72"  
height="72" />  
    <icon src="slike/ios114.png" width="114" height="114" />
```

```

<gap:splash src="slike/splash.png" />
<description>
    Aplikacija bazirana na web standardima.
</description>

    <author href="http://podvorec.info"
email="i.podvorec@gmail.com">
        Ivan Podvorec
    </author>

    <preference name="phonegap-version" value="2.2.0" />
    <preference name="permissions" value="none"/>
</widget>

```

Izvorni kod 5.10 Sadržaj *config.xml* datoteke izrađene aplikacije

Osim osnovnih podataka o samoj aplikaciji unutar konfiguracijske datoteke (**Izvorni kod 5.10**) definirane su i ikone aplikacije koje će se pojaviti nakon instalacije unutar grafičkog sučelja operacijskog sustava na kojem je instalirana aplikacija. Također definirane su i specifične dozvole koje su potrebne da bi aplikacija ispravno radila. Ova aplikacije ne zahtijeva nikakve posebne dozvole već samo onu za pristup internetu koja je zadana od strane PhoneGap razvojnog okruženja te se ne može obrisati.

Svako korištenje neke od nativnih funkcionalnosti ugrađenih u PhoneGap programsko razvojno okruženje potrebno je navesti unutar konfiguracijske datoteke. Ova aplikacija koristi jednu nativnu funkcionalnost ugrađenu u PhoneGap razvojno programsko okruženje. To je funkcionalnost početnog ekrana učitavanja aplikacije (eng. *splashscreen*) koja je definirana unutar konfiguracijske datoteke. Također unutar programskog dijela aplikacije potrebno je, kada se aplikacija u potpunosti učita, sakriti ekran učitavanja aplikacije što je prikazano na sljedećem izvornom kodu: (**Izvorni kod 5.11**)

```

document.addEventListener("deviceready", onDeviceReady, false);
function onDeviceReady() {

```

Izvorni kod 5.11 Dio koda zadužen za sakrivanje početnog ekrana učitavanja aplikacije

Ukoliko unutar hibridne mobilne aplikacije koristimo native dodatke (eng. *plugin*) iste je potrebno navesti na ispravan način unutar konfiguracijske datoteke kako bi se mogle na ispravan način inicijalizirati prilikom pokretanja aplikacije.

Unutar *index.html* datoteke potrebno je između `<head></head>` elementa dodati poziv za *phonegap.js* skriptu (**Izvorni kod 5.12**) jer se u suprotnom dijelovi aplikacija neće prikazati prilikom pokretanja aplikacije na mobilnom uređaju.

```
<head>
...
<script src="phonegap.js"></script>
...
</head>
```

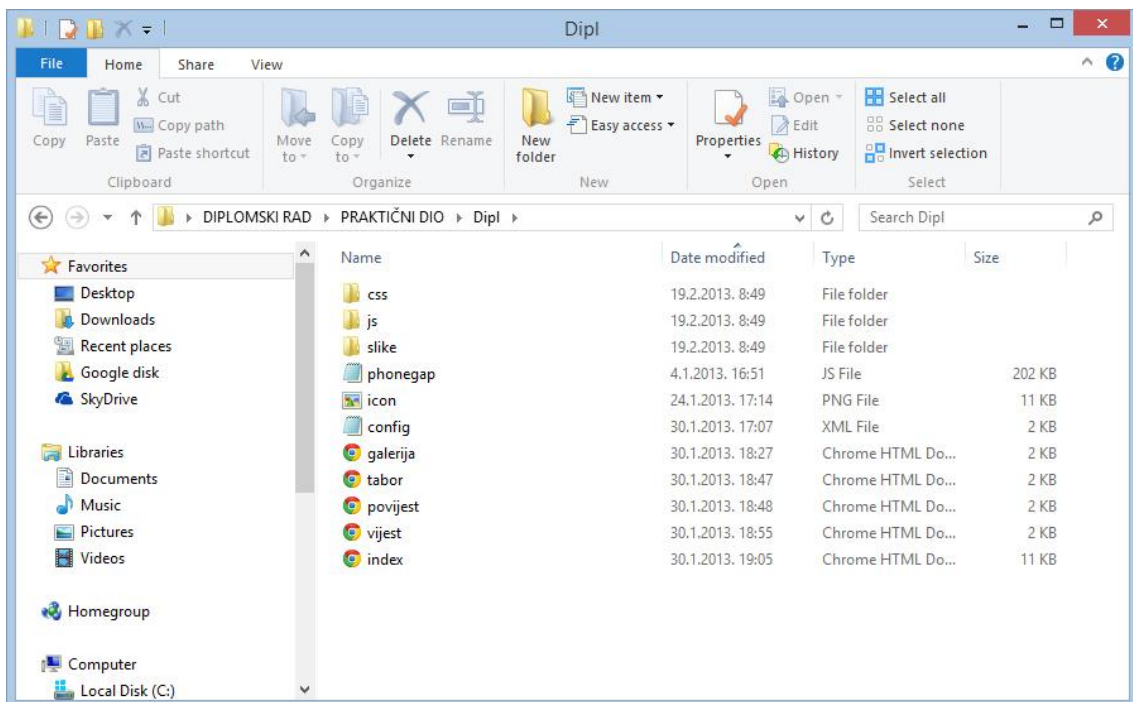
Izvorni kod 5.12 Poziv PhoneGap JavaScript skripte

5.2.2. PhoneGap Build izrada nativnih instalacija

Nakon uspješno izrađene konfiguracijske datoteke aplikacije za PhoneGap Build web servis potrebno je sve datoteke na pravila način zapakirati u *.zip* datoteku kako bi se moglo jednostavno prenijeti na web servis (**Error! Reference source not found.**).

Općenito za izradu nativnih instalacija PhoneGap Build web servisom u korijenskom direktoriju (eng. *root*) *.zip* datoteke jedina obavezna datoteka je *index.html* dok sve ostale su opcionalne. Kako se u aplikaciji koristi nativna funkcionalnost početnog ekrana učitavanja aplikacije obavezna je i gore navedena konfiguracijska datoteka (*config.xml*). Nakon uspješno zapakirane *.zip* datoteke istu je potrebno prenijeti na web servis PhoneGap Build. Za generiranje Windows Phone i Symbian aplikacije nije potreban privatni ključ dok za generiranje Android, Blackberry te iOS aplikacije je potrebno na PhoneGap Build servis prenijeti privatne ključeve kako bi se iste mogle generirati. Kako generiranje privatnog ključa za iOS platformu nije

besplatno te je potrebno registracija isključivo s Macintosh računala ista ne može biti



Slika 5.8 Struktura .zip datoteke izradene aplikacije

generirana.

Proces izrade nativnih instalacija na PhoneGap Build web servisu može trajati od 5 minuta pa sve do 30 i više minuta. Nakon što je web servis generirao sve moguće aplikacije, a što je ovisno o isporučenim privatnim ključevima, iste se mogu preuzeti te su spremne za objavu na pripadajućima dućanima aplikacija.

Generirane aplikacije su dostupne za preuzimanje sa adrese: www.podvorec.info/dipl.

6. ZAKLJUČAK

U praktičnom dijelu rada je prikazana izrade hibridne mobilne aplikacije zasnovane na web standardima. Sama svrha izrade ove aplikacije je istraživanja mogućnosti i ograničenja ovakvih vrsta aplikacija.

Razvojem i napretkom tehnologije, sve većom dostupnošću bežičnog mobilnog interneta pametni telefoni i tableti sve su pristupačniji i zanimljiviji sve većem broju korisnika. Tako nam je i najjednostavnije doprijeti do postojećih, ali i novih klijenata i/ili korisnika izradom mobilne aplikacije za što veći broj mobilnih platformi.

Kako je izrada aplikacije u nativnom programskom jeziku poprilično komplicirana s obzirom na sve veći broj mobilnih operacijskih sustava, a samim time i cijena aplikacije dostupne na što većem broju mobilnih operacijskih sustava drastično raste. Izradom hibridne mobilne aplikacije bazirane na web standardima vrijeme i cijena izrade drastično pada jer se jednom napisani kod aplikacije može uz eventualne minimalne prilagodbe „reciklirati“ na više mobilnih operacijskih sustava. Međutim, za odluku na koji način će se izraditi željena mobilna aplikacija treba razmotriti ostale prednosti i nedostatke hibridnih mobilnih aplikacija, a ne samo vrijeme i cijenu razvoja. Ukoliko željena aplikacija nije izuzetno grafički zahtjevna te ne zahtijeva posebne native mogućnosti, osim onih uključenih u razvojno okruženje za izradu hibridnih mobilnih aplikacija (PhoneGap), kao primarni način izrade mobilne aplikacije u tom slučaju trebalo bi razmotriti aplikaciju zasnovanu na web standardima. S druge strane ukoliko aplikacija zahtijeva native mogućnosti koje nisu podržane od strane razvojnog okruženja ili je grafički zahtjevna kao primarni način izrade aplikacije trebalo bi odabrati programiranje aplikacije u nativnom programskom jeziku.

7. LITERATURA

- 1 *** http://www.openhandsetalliance.com/press_110507.html – Open Handset Alliance, studeni 2007.
- 2 *** <http://android-developers.blogspot.com/2008/09/announcing-android-10-sdk-release-1.html> – rujan 2008.
- 3 *** <http://developer.android.com/tools/sdk/ndk/index.html> - Android
- 4 *** <http://www.businessweek.com/news/2012-10-29/google-says-700-000-applications-available-for-android-devices> - Listopad 2012.
- 5 *** <http://www.webcitation.org/5oqYO6thc> – Apple, srpanj 2010.

- 6 [http://msdn.microsoft.com/library/windowsphone/develop/ff402529\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windowsphone/develop/ff402529(v=vs.105).aspx) – Microsoft, Studeni 2012

- 7 [http://msdn.microsoft.com/library/windowsphone/help/jj206719\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windowsphone/help/jj206719(v=vs.105).aspx) – Microsoft, Studeni 2012
- 8 *** <http://www.marketwire.com/press-release/RIM-Unveils-The-BlackBerry-PlayBook-NASDAQ-RIMM-1325727.htm> - Rujan 2010.
- 9 *** <https://developer.blackberry.com/java/download/eclipse/> - BlackBerry
- 10 *** <http://www.w3.org/html/> - World Wide Web Consortium (W3C)
- 11 <http://www.w3.org/2011/02/htmlwg-pr.html> - W3C
- 12 <http://www.strategyanalytics.com/default.aspx?mod=pressreleaseviewer&a0=5145> – Strategy Analytics, Prosinac 2012.
- 13 *** <http://www.w3.org/Style/CSS/> - W3C
- 14 Scott Duffy (2003). *How to do everything with JavaScript*, McGraw-Hill/Osborne, ISBN 0-07-222887-3
- 15 *** <http://www.json.org>
- 16 *** <http://iceniium.com/community/blog/iceniium-team-blog/2012/06/14/what-is-a-hybrid-mobile-app-> - Doug Seven, Lipanj 2012.

POPIS PRILOGA

PRILOG BR. 1

CD s izvornim kodom praktičnog dijela aplikacije

POPIS SLIKA

Slika 2.1 Android korisničko sučelje, verzija Androida 4.0.x <i>http://developer.android.com/about/versions/android-4.0-highlights.html</i>	2
Slika 2.2 Eclipse razvojno okruženje	3
Slika 2.3 iOS 5 sučelje, iPhone http://photos.appleinsider.com/ios5-110607-6.jpg	4
Slika 2.4 Xcode 4 sučelje http://upload.wikimedia.org/wikipedia/en/8/8c/Xcode4.png	5
Slika 2.5 Windows Phone 7 početni ekran http://www.hindustantimes.com/Images/HTEditImages/Images/1_WP7_start_screen.png	6
Slika 3.1 Usporedba prikaza HTML-a. S lijeve strane bez primjene CSS-a, a s desne strane uz primjenu CSS-a	13
Slika 4.1 Sučelje web alata za izradu tema za jQuery Mobile	17
Slika 4.2 Prikaz PhoneGap arhitekture aplikacija	18
Slika 4.3 Sučelje web servisa PhoneGap build	21
Slika 5.1 Shema rada aplikacije	25
Slika 5.2 Prikaz ikona aplikacije	26
Slika 5.3 Ekran učitavanja aplikacije	26
Slika 5.4 Prikaz ekrana uređaju u portretnom položaju. Slika desno bez otvorenog izbornika, slika lijevo s otvorenim izbornikom	27
Slika 5.5 Prikaz zaglavlja i galerije	28
Slika 5.6 Prikaz galerije preko cijelog ekrana	29
Slika 5.7 Prikaz sučelja aplikacije u pejzažnom položaju	29
Slika 5.8 Struktura .zip datoteke izrađene aplikacije	46

POPIS TABLICA

Tablica 4.1 Podržane native mogućnosti PhoneGap razvojne okoline..... 19

Tablica 4.2 Usporedba mogućnosti web, hibridnih i nativnih mobilnih aplikacija..... 22

POPIS IZVORNIH KODOVA

Izvorni kod 3.1 Jednostavan HTML dokument.....	9
Izvorni kod 3.2 Primjer linijskog CSS-a	10
Izvorni kod 3.3 Isječak HTML koda s uključenim unutarnjim CSS-om	11
Izvorni kod 3.4 Jednostavna CSS dokument za oblikovanje HTML-a	11
Izvorni kod 3.5 HTML dokument s uključenim vanjskim CSS dokumentom.....	12
Izvorni kod 3.6 Jednostavni JavaScript program	14
Izvorni kod 3.7 Primjer JSON objekta	15
Izvorni kod 3.8 Primjer JSON polja	15
Izvorni kod 4.1 Isječak JavaScript koda za snimanje i prikaz slike	20
Izvorni kod 5.1 CSS3 medijski upiti za prilagodbu izgleda sučelja aplikacije	31
Izvorni kod 5.2 Isječak koda za provjeru pripremljenosti uređaja	32
Izvorni kod 5.3 Struktura HTML dokumenta.....	34
Izvorni kod 5.4 Isječak koda zadužen za prikaz i sakrivanje izbornika	34
Izvorni kod 5.5 Primjer JSON podataka koje isporučuje web poslužitelj.....	35
Izvorni kod 5.6 Isječak koda za dohvat podataka te prikaz istih unutar liste u grafičkom sučelju	38
Izvorni kod 5.7 Isječak koda za dohvat i prikaz opširnije vijesti	40
Izvorni kod 5.8 Primjer dohvat podataka za galeriju, generiranje HTML-a i pozivanje funkcije galerije	41
Izvorni kod 5.9 Dio koda zadužen za funkcioniranje galerije slika	43
Izvorni kod 5.10 Sadržaj <i>config.xml</i> datoteke izrađene aplikacije	44
Izvorni kod 5.11 Dio koda zadužen za sakrivanje početnog ekrana učitavanja aplikacije	45
Izvorni kod 5.12 Poziv PhoneGap JavaScript skripte	45

POPIS OZNAKA I KRATICA

Oznaka / Kratica	Puni naziv
ADT	Android Development Tools
API	Application Programming Interface
CR	Candidate recommendation
CSS	Cascading Style Sheets
DDMS	Dalvik Debug Monitor Server
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
NDK	Native Development Kit
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
RIM	Research In Motion
SDK	Software Development Kit
UI	User Interface
XML	eXtensible Markup Language