

RAZVOJ HIBRIDNE MOBILNE APLIKACIJE

Muharem Bašanović, Amel Toroman, Majda Kajtažović
Univerzitet u Bihaću Tehnički fakultet Bihać, Ulica dr. Irfana Ljubijankića bb
77000 Bihać, muharem.basanovic@gmail.com
Univerzitet u Bihaću Tehnički fakultet Bihać, Ulica dr. Irfana Ljubijankića bb
77000 Bihać, amel.toroman@gmail.com
Univerzitet u Bihaću Tehnički fakultet Bihać, Ulica dr. Irfana Ljubijankića bb
77000 Bihać, majdak@windowslive.com

Ključne riječi: mobilne aplikacije, hibridne aplikacije, mobilni uređaji, web aplikacije, HTML, CSS, JavaScript, PhoneGap

SAŽETAK:

Hibridne aplikacije predstavljaju kombinaciju nativnih i web aplikacija, prednost ove kombinacije je u mogućnosti razvoja aplikacija na više platformi. Najčešće korištene tehnologije su: HTML, CSS i JavaScript koje se izvode preko web view kontrola (UIWebView za iOS, WebView za Android i ostalih), za korištenje nativnih funkcionalnosti kao što su: senzori, kamera i drugo koriste se biblioteke koje su namjenjene razvoju hibridnih aplikacija. Na primjeru aplikacije prilagođene izvršavanju na Android platformi prikazan je razvoj hibridne mobilne aplikacije sa mogućnošću prilagođavanja iste za izvršavanje na različitim mobilnim platformama.

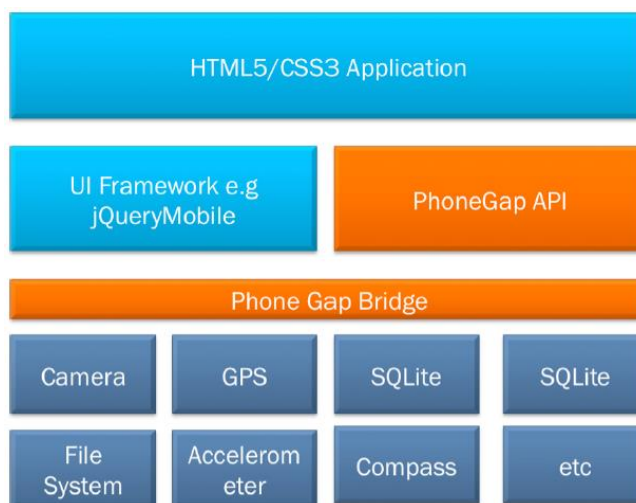
1. UVOD

Popularnost „pametnih“ telefona i drugih sličnih uređaja u stalnom je porastu[1], razvoj aplikacija namjenjenih takvim uređajima može biti zasnovan na različitom pristupu. *Nativne aplikacije* – u ovom pristupu aplikacija je razvijena koristeći nativni programski jezik mobilne platforme (nap. xCode/Objective-C – iOS, Eclipse/Java – Android, Visual Studio/C# – Windows Phone i drugih). Prednost programiranja nativnih aplikacija vidljiva je u visokim performansama i bogatom korisničkom iskustvu. Nedostatak nativnog razvoja aplikacija je da tako napravljene aplikacije je uglavnom teško prebaciti na drugu platformu, bez da se cijeli programski kôd ne piše ispočetka. *Mobilne web aplikacije* – ovaj pristup se temelji na dokazanim standardnim web tehnologijama. To predstavlja najjednostavniji pristup razvoja aplikacija za mobilne uređaje, bez puno ulaganja. Iako je ova opcija fleksibilna i atraktivna, korisničko iskustvo je ograničeno na skup osnovnih funkcija, bez mogućnosti pristupa nativnim funkcionalnostima mobilnog uređaja. *Hibridne mobilne aplikacije* – dizajnirane su da se iskoristi nativni i web pristup, koriste fleksibilne mogućnosti web tehnologija sa kombinacijom nativnih funkcionalnosti uređaja, prednost svega toga je da se takve aplikacije bez ili

sa malom promjenom u kôdu mogu pokrenuti na različitim podržanim platformama. Svaki od pristupa u razvoju mobilnih aplikacija ima svoje prednosti i nedostatke. U nastavku rada opisan je razvoj hibridne mobilne aplikacije.

2. HIBRIDNE MOBILNE APLIKACIJE

Hibridne mobilne aplikacije zasnivaju se na web tehnologijama prvenstveno na novoj verziji HTML5 (*engl. HyperText Markup Language*), koja je donijela nove mogućnosti u kombinaciji sa JavaScriptom i CSS-om (*engl. Cascading Style Sheets*), a pokreću se u web pregledniku mobilnog uređaja (*engl. web view*) kontroli. Web preglednici mobilnih uređaja u samom početku podržavali su osnovne funkcionalnosti HTML5 tehnologije uz prividno kašnjenje u izvršavanju i obradi informacija. Napretkom u razvoju mobilnih uređaja razvijeni su napredniji i brži mobilni web preglednici (*web view kontrole*) koje danas podržavaju većinu funkcionalnosti koje nudi HTML5 tehnologija[2]. Primjenom web tehnologije i spajanjem nativnih funkcionalnosti uređaja nastale su hibridne mobilne aplikacije. Izvedeno je uspostavljanjem apstraktnog sloja koji spaja nativni API (*engl. Application Programming Interface*) sa JavaScript API-em. Problem ovog pristupa je u činjenici da za svaku platformu treba biti razvijen poseban kontejner (*biblioteka*), koja se nadograđuje za svaku novu funkcionalnost uređaja. Danas postoje različite biblioteke za razvoj hibridnih mobilnih aplikacija među koje spada i PhoneGap biblioteka otvorenog kôda (*engl. open source framework*) koja omogućuje izradu hibridnih aplikacija korištenjem standardizovanog web API-a za različite platforme, na slici 1. prikazana je arhitektura PhoneGap aplikacije[3].



Slika 1: Arhitektura PhoneGap aplikacije[4]

Pristup API-u uređaja vrši se pomoću JavaScript interfejsa kojem se pristupa preko programskog kôda aplikacije. Interfejs poziva nativnu realizaciju određene funkcionalnosti. Nativne realizacije se nalaze u PhoneGap biblioteci za svaku pojedinačnu platformu, PhoneGap trenutno podržava sljedeće API-e[5] za devet mobilnih platformi: *accelerometer, camera, capture, compass, connection, contacts, device, events, file, geolocation, globalization, inappbrowser, media, notification, splashscreen, storage*. Ovakav pristup eliminira potrebu za poznavanjem nativnog jezika pojedine

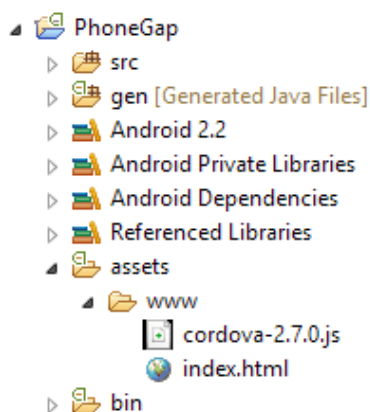
razvojene platforme usmjeravajući fokus na višeplatformski razvoj aplikacije. Često se javlja potreba za realizacijom nativnih funkcionalnosti uređaja koje nisu podržane od strane PhoneGap API-a. U takvim slučajevima koriste se dodaci (*engl. plugin*) koji se realizuju preko JavaScript interfejsa a komuniciraju preko *cordova.exec* funkcije sa nativnim implementacijama[6]. Uslov za realizaciju ovog pristupa zahtjeva poznavanje razvojnih tehnologija, platformi za koje se realiziraju nativne funkcionalnosti.

2.1. Primjer razvoja hibridne mobilne aplikacije

U ovom dijelu prikazan je razvoj hibridne mobilne aplikacije „prilagođene“ Android platformi primjenom PhoneGap biblioteke. U osnovi postoje dva načina razvoja aplikacije: prvi se realizuje uz pomoć SDK (*engl. Software Development Kit*) i razvojnih alata za svaku platformu a uključuje podešavanje aplikacije/projekta izvršavanju PhoneGap API-a, drugi se realizuje preko PhoneGap servisa koji automatski pakuju aplikaciju za izabranu platformu[7]. Prilikom izrade aplikacije pomoću SDK i razvojnih alata za Android platformu potrebno je ispuniti određene preduslove koji uključuju[8]:

- kreiranje novog android projekta
- u direktorijumu *assets* napraviti novi *www* direktorij
- referencirati *cordova-2.7.0.jar* biblioteku te dodati *cordova-2.7.0.js* biblioteku u *www* direktorij
- kopirati *config.xml* datoteku u direktorijum *xml* iz PhoneGap resursa
- izmijeniti *AndroidManifest.xml* datoteku tako što će se dodati dozvole (*engl. permission*) za pristupanje funkcionalnostima uređaja kao što je: kamera, senzor i dr.
- izmijeniti dio kôda u glavnoj **Activity.java* klasi nasljeđivanjem kalse *DroidGap.java* i izmjenom metode *onCreate* da se učitava sadržaj iz *www* direktorijuma tj. *index.html* stranica prilikom pokretanja aplikacije.

Ispunjavanjem svih potrebnih preduslova može da se počne sa izradom aplikacije, u osnovi izrada aplikacije svodi se na izradu web stranica primjenom HTML5 tehnologije smještenih u *www* direktorijum projekta, slika 2.



Slika 2: Sadržaj *www* direktorijuma

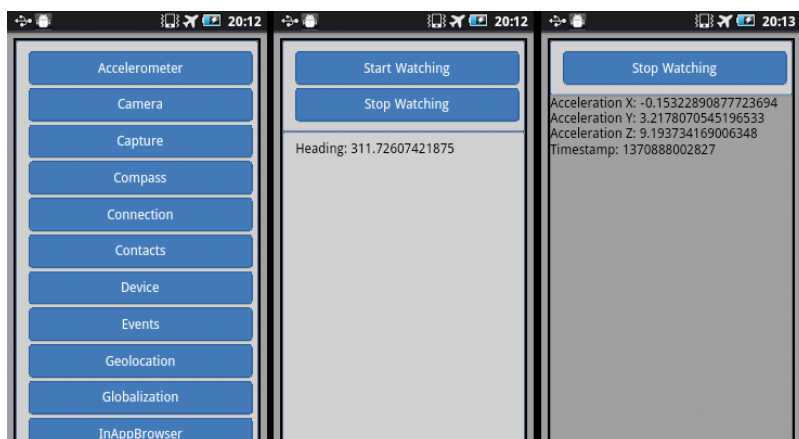
Svaka od *.html* stranica ukoliko pristupa nativnim funkcionalnostima uređaja treba biti referencirana JavaScript API-em tj. *cordova-2.7.0.js*:

```
<script type="text/javascript" charset="utf-8" src="cordova-2.7.0.js"></script>
```

koji predstavlja „vezu“ između nativnog API-a *cordova-2.7.0.jar* i web aplikacije. Ili dodavanjem vlastitih implementacija ukoliko su razvijene preko dodataka (*engl. plugin-a*). Pristup nativnim funkcionalnostima uređaja kao što je kompas (*engl. compass*) može se izvršiti pozivanjem JavaScript funkcije:

```
document.addEventListener("deviceready", onDeviceReady, false);
function onDeviceReady() {
    navigator.compass.getCurrentHeading(onSuccess, onError);
}
function onSuccess(heading) {
    alert('Heading: ' + heading.magneticHeading);
}
function onError(compassError) {
    alert('Compass Error: ' + compassError.code);
}
```

Nakon kodiranja svih *.html* stranica i provjere ispravnosti aplikacije korisnik dobiva izvršnu *.apk* datoteku koja je spremna za instalaciju na Android platformu – uređaje, na slici 3. prikazana je primjer aplikacije napravljene korištenjem PhoneGap biblioteke.



Slika 3: Primjer hibridne mobilne aplikacije razvijene sa PhoneGap

Da bi se u potpunosti iskoristile funkcionalnosti hibridnih aplikacija pokretanje na različitim platformama potrebno je kopirati sadržaj iz *www* direktorijuma u novi projekt ciljane platforme a podržane od PhoneGap-a te izmjeni nativne biblioteke prilagođene platformi, ili iskoristiti funkcionalnost PhoneGap online servisa dostavljanjem HTML, CSS i JavaScript fajlova.

3. PREDNOSTI I NEDOSTACI HIBRIDNIH APLIKACIJA

Pored mogućnosti da se hibridne mobilne aplikacije razviju za relativno kratko vrijeme upotrebom standardizovanih web tehnologija, pri čemu se isti izvorni kôd može ponovo koristiti za distribuciju aplikacije na više različitih platformi one imaju određene prednosti i nedostatke[9]:

Prednosti:

- Brže vrijeme na tržištu.
- HTML5 programeri obično su jeftiniji, troškovi održavanja su niži.
- Prenosivost kôda, pokretanje aplikacija na više platformi bez/djelimične izmjene kôda.
- Pristup nativnim funkcionalnostima.
- Jednostavno održavanje.

Nedostaci:

- Niže performanse.
- Ograničenja za nativne funkcionalnosti.
- Sigurnost.

Izbor između nativnog, web i hibridnog pristupa izrade aplikacije prvenstveno ovisi od namjene aplikacije i funkcionalnosti koje ona treba da ima.

4. ZAKLJUČAK

Hibridne mobilne aplikacije zasnovane su na web tehnologijama sa pristupom nativnim funkcionalnostima uređaja, za razliku od izvornih – nativnih aplikacija hibridne aplikacije moguće je pokrenuti na različitim platformama što za krajnji cilj ima smanjenje troškova razvoja. Sa druge strane takve aplikacije često imaju niže performanse i određena ograničenja u pristupu hardveru. Svako rješenje koje se oslanja na web tehnologije prvenstveno na HTML standard koji polako evoluira tokom godina i sa malom vjerovatnošću da će se dramatično promijeniti kao što je slučaj sa vlasničkim rješenjima nudi stabilnu osnovu na kojoj je moguće graditi aplikacije sa sigurnošću da će nastaviti raditi i u dalekoj budućnosti bez velikih promjena.

5. LITERATURA

- | | |
|---|----------------------------------|
| [1] http://www.gartner.com/newsroom/id/2335616 | pregledano: 14. 05. 2013 godine. |
| [2] http://mobilehtml5.org/ | pregledano: 10. 05. 2013 godine. |
| [3] http://docs.phonegap.com/en/2.8.0/index.html | pregledano: 10. 05. 2013 godine. |
| [4] Ghatol R., Patel Y.: Beginning PhoneGap mobile web framework for JavaScript and HTML5, Apress Media LLC New York, 2012 | |
| [5] http://phonegap.com/about/feature/ | pregledano: 14. 05. 2013 godine. |
| [6] http://www.adobe.com/devnet/html5/articles/extending-phonegap-with-native-plugins-for-android.html | pregledano: 20. 05. 2013 godine. |
| [7] https://build.phonegap.com/ | pregledano: 15. 05. 2013 godine. |
| [8] http://www.adobe.com/devnet/html5/articles/getting-started-with-phonegap-in-eclipse-for-android.html | pregledano: 20. 05. 2013 godine. |
| [9] http://orisys.in/blog/hybrid-mobile-applications/ | pregledano: 20. 05. 2013 godine. |