Sveučilište u Rijeci – Odjel za informatiku

Jednopredmetna informatika

Jan Skopljak

Web aplikacije za mobilne uređaje

Završni rad

Mentor: mr.sc. Igor Jugo

Rijeka, 23.09.2016.

Sadržaj

[Uvod 4](#_Toc462357042)

[1. Mobilni web 5](#_Toc462357043)

[1.1 Povijest pametnih telefona 5](#_Toc462357044)

[1.2 Što je mobilni web? 6](#_Toc462357045)

[1.2.1 Responzivni dizajn 7](#_Toc462357046)

[1.2.2 Adaptivni dizajn na strani klijenta 7](#_Toc462357047)

[1.2.3 Adaptivni dizajn na strani servera 8](#_Toc462357048)

[1.3 Što su točno mobilne web aplikacije? 9](#_Toc462357049)

[1.4 Nativne aplikacije vs mobilne web aplikacije 9](#_Toc462357050)

[1.4.1 Hibridne aplikacije 10](#_Toc462357051)

[2. jQuery mobile 12](#_Toc462357052)

[2.1 Što je jQuery mobile? 12](#_Toc462357053)

[2.2 Alternative za jQuery Mobile 13](#_Toc462357054)

[3. Softweari za razvoj hibridnih mobilnih aplikacija 15](#_Toc462357055)

[3.1 Apache Cordova 15](#_Toc462357056)

[3.1.1 Arhitektura 15](#_Toc462357057)

[3.1.2 Webview 16](#_Toc462357058)

[3.1.3 Web app 16](#_Toc462357059)

[3.1.4 Plugini 16](#_Toc462357060)

[3.1.5 Služenje cordovom 17](#_Toc462357061)

[3.2 Adobe PhoneGap 18](#_Toc462357062)

[4. Prezentacijska hibridna aplikacija 19](#_Toc462357063)

[4.1 Backend 19](#_Toc462357064)

[4.2 Frontend 21](#_Toc462357065)

[Zaključak 22](#_Toc462357066)

[Popis literature 23](#_Toc462357067)

[Popis priloga 24](#_Toc462357068)

# Uvod

Izabrao sam ovu temu zbog toga što je usko povezano s trenutnim usmjerenjem, a to je razvoj web aplikacija. U svijetu je po nekim izvorima danas više mobilnih uređaja nego ljudi i samim time sve više potrošenog vremena na korištenje interneta putem mobilnih uređaja. Broj uređaja i verzija platformi je u konstantnom porastu i time se komplicira razvoj aplikacija za mobilne uređaje. U ovom radu proći ću kroz probleme dosadašnjeg razvoja nativnih mobilnih aplikacija i ponudit ću alternativu u obliku mobilnih web aplikacija/hibridnih aplikacija.

U prvom dijelu ovog rada dati ću kratak povijesni pregled razvoja pametnih telefona (danas poznatijih kao smartphonea) i mobilnog weba. Pojasniti ću što je točno mobilni web i na kojim bi se principima on trebao graditi te ponuditi tehnike putem kojih će prikaz web stranica na mobitelu biti prilagođen samom uređaju. U nastavku ću se osvrnuti na pojam mobilnih web aplikacija i dati usporedbu između nativnih i mobilnih web aplikacija te ponuditi rješenje koje se nalazi između, a to su hibridne aplikacije.

U drugom dijelu rada osrvnuti ću se na neke pogodnosti dobivene korištenjem mobilnih frameworka i na njihovo uklapanje u razvoj mobilnih web aplikacija i lako prilagođavanje hibridnim aplikacijama. Konkretno ću se osvrnuti na jQuery mobile jer sam njega koristio u svojem projektu, ali kako je već danas jQuery mobile zbog dugog ne ažuriranja frameworka zastario ponudit ću i neke danas popularnije alternative.

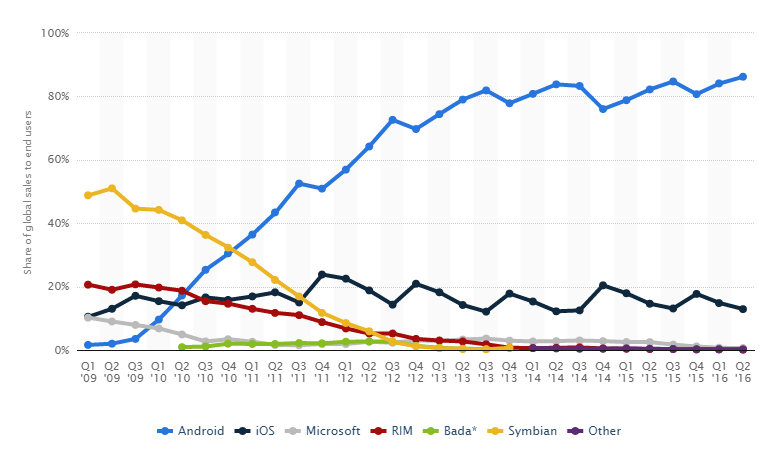
Treći dio rada sastoji se od konkretnih tehnologija koje su potrebne kako bi se mobilna web aplikacija prebacila u hibridnu aplikaciju. Osvrnut ću se na danas vodeći softwear, Apache Corodvu i na distribuciju Apache Cordove, PhoneGap.

Završnom radu priložit ću i funkcionalnu hibridnu aplikaciju koja je u potpunosti napravljena ovim tehnologijama te ću se u četvrom dijelu rada osvrnuti na neke bitne stvari kod samog razvoja takve aplikacije.

# 1. Mobilni web

## 1.1 Povijest pametnih telefona

Prije skoro 10 godina, 29. lipnja 2007. godine izašao je prvi Iphone pod sloganom "Ovo je samo početak" i zauvijek revolucionarizirao način na koji ljudi gledaju na prijenosne telefone. Bio je to prvi "smartphone" s primamljivim korisničkim iskustvom, mnoštvom aplikacija i igara, jednostavnim i intuitivnim pristupom internetu, kamerom, žiroskopom. Nakon količine popularnosti koju je Iphone stekao s inačicom 3g nije bilo potrebno dugo kako bi se i druge kompanije uključile u utrku ka dominaciji tržištem pametnih telefona. Iz godine u godinu broj proizvođača pametnih telefona rastao je te samim time i broj uređaja. Kako je rastao broj uređaja rastao je i broj operacijskih sustava na kojima ti uređaji funkcioniraju. Tako su se na tržištu našli, osim Apple iOSa koji pokreće Iphone, Symbian, Android (iz kojega je kasnije nastao danas najpopularnija platforma, googleov OS - Android OS, Blackberry OS, Windows Phone OS itd.). Nakon nekog vremena većina ovih operacijskih sustava ispala je iz konkurencije te su ostala danas dva najveća konkurenta, Apple iOS i Android OS te prateći Windows Phone OS. Apple iOS se pokreće isključivo na Iphoneima, Windows Phone na Nokiama te Android na velikoj većini ostalih uređaja. Kako su tehnološke mogućnosti i želje korisnika pametnih telefona rasle, razvijale su se i novije verzije svakog od operativnih sustava. Razviti mobilnu aplikaciju koju će podržavati sve ove uređaje, gotovo je nemoguće.



Slika 1. - podijeljenost tržišta po platformama pametnih telefona

Kako su napredovali mobilni uređaji napredovale su i internet tehnologije. Tako je izašla nova verzija standarda HTML, HTML5 koji je donio nove tehnologije u svijet markup jezika: browserska geolokacija, semantički bolje razrađena struktura html dokumenta uz nove elemente (header, footer, nav, article itd.), poboljšana podrška za audio i video medije, apiji (canvas, drag and drop, web storage itd.). Uz HTML5, izašla je i nova verzija CSS-a, CSS3 koji je uz svoj media-query donio preporod za responzivni dizajn. Responzivni web dizajn je puno napredovao od doba tablica i statičnih dimenzija te je sada lakše nego ikada prije razviti web stranicu koja je pregledna na svim rezolucijama.

## 1.2 Što je mobilni web?

Uz konstantan rast korisnika mobilnih uređaja, već 2006., W3C (World Wide Web Consortium) je počeo promovirati ideju „Jednog Weba“ što u konačnici znači dostupnost istih informacija i usluga korisnicima neovisno o uređaju kojeg koriste. Međutim, to ne znači da su sasvim jednake informacije dostupne na mobilnim uređajima i na desktop računalima. Čimbenici kao što su kontekst primjene mobilnih web stranica, sposobnosti samog uređaja, problemi propusnosti podataka i problemi mobilnih mreža utječu na reprezentativnost samog prikaza podataka. Nadalje, neke usluge i informacije su pogodnije za određene uređaje.

U početku prikaza internet sadržaja na mobilnim uređajima većina stranica koristila se trikom da ukoliko se korisnik povezuje s mobilnog uređaja, preusmjeri ga se na poddomenu stranice s prefixom m. (<http://m.example.com>). Na takvoj poddomeni obično se nalazila potpuno druga stranica napravljena isključivo za mobilne uređaje. To se kosi s idejom o Jednom Webu te je tehnološkim napretkom takvo razvijanje mobilnog weba u potpunosti nestalo.

Kako bi se ideja o Jednom webu realizirala web developeri se mogu osloniti na responzivni i adaptivni dizajn.

### 1.2.1 Responzivni dizajn

Responzivni dizajn uvelike se oslanja na CSS3 novitet, media query. Responzivni dizajn danas je standard u web developmentu i broj responzivnih stranica je u stalnom rastu. Glavna stavka ovog pristupa je jednostavnost implementacije responzivnosti. Kroz sami CSS developeri mogu postaviti kako će se na kojim veličinama ekrana web stranice ponašati.

Uz to što web developeri sami mogu razvijati responzivne web stranice, u stalnom je porastu i broj User interface frameworka od kojih se najviše ističe Twitterov Bootstrap. Negativna strana responzivnog dizajna je u tome što se većina web stranica mora raditi iz nule s mobilnim prikazom na prvom mjestu.



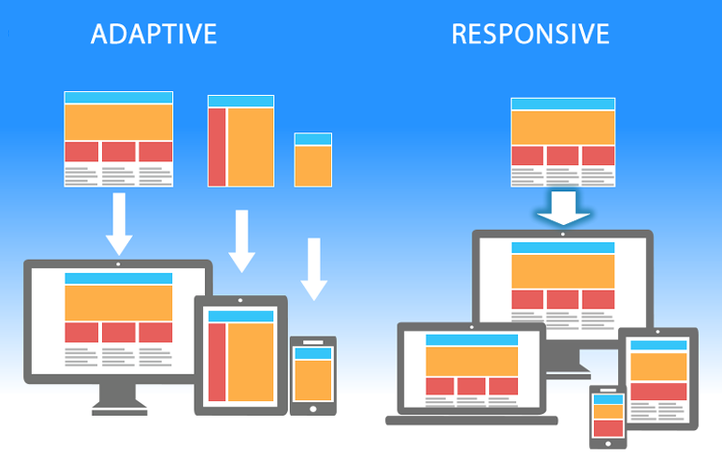
Slika 2. - ilustracija responzivnog prikaza na različitim uređajima

### 1.2.2 Adaptivni dizajn na strani klijenta

Adaptivni dizajn građen je na principima responzivnog dizajna da se korisničko iskustvo poboljša ovisno o specifičnom uređaju i kontekstu. Adaptivni dizajn na strani klijenta koristi JavaScript kako bi obogatio web stranicu naprednim funkcionalnostima i prilagodbama. Nije potrebno stranicu izrađivati od nule, već se na već postojeći template nadodaju funkcionalnosti potrebne za mobilne uređaje. Prednost ovog pristupa je personalizirana vrsta prikaza ovisno o uređaju s kojeg se pristupa web stranici. Na primjer, adaptivne web stranice dostavit će slike Retina kvalitete samo uređajima koje koriste Retina ekran dok će ostalim uređajima biti dostavljene slike niže kvalitete.

### 1.2.3 Adaptivni dizajn na strani servera

Prednost adaptivnog dizajna na strani servera je što se logika za prepoznavanje uređaja s kojega se pristupa web stranici nalazi na serverskoj strani te se ovisno o uređaju šalje drugačiji template. U konačnici to znači manja mobilna web stranica što znači i brže učitavanje. Doduše implementacija ovakvog rješenja često se zna odužiti i poskupiti samo održavanje projekta. Ukoliko je server pod visokim opterećenjem ovakav pristup može ga dodatno opteretiti što u konačnici rezultira i lošijim korisničkim iskustvom.[1]



Slika 3. - ilustracija templating sustava ovisno o pristupu

## 1.3 Što su točno mobilne web aplikacije?

Kako broj korisnika pametnih mobilnih uređaja raste, raste i potražnja za mobilnim aplikacijama. Svaka malo ozbiljnija tvrtka posjeduje svoju inačicu mobilne aplikacije. Isto tako svaka ozbiljnija tvrtka posjeduje i web stranicu potpuno prilagodljivu mobilnim uređajima. Mobilne web aplikacije su u principu web stranice koje u svakom pogledu izgledaju kao nativne mobilne aplikacije, ali nisu implementirane kao takve. Pokreću se putem browsera koristeći HTML5. Korisnici im pristupaju putem interneta i nemaju potrebu instalirati web aplikaciju na vlastite uređaje. Izlaskom HTML5 standarda i pristupom nativnim funkcijama uređaja(camera, GPS, kontakti itd.) ljudi su počeli shvaćati kako potreba za samostalnom nativnom aplikacijom sve više gubi smisao. No, mobilne web aplikacije pokretane putem browsera i dalje nemaju pristup notifikacijama koje se pokreću u pozadini, akcelerometarskim informacijama, kompleksnim gestama, a za pokretanje im je isključivo potreban internet pristup.

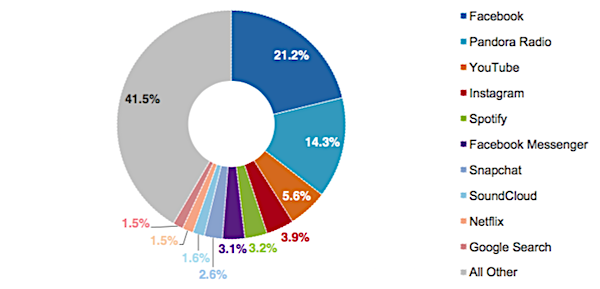
## 1.4 Nativne aplikacije vs mobilne web aplikacije

Sve veća popularnost pametnih telefona rezultirala je u ogromnoj količini različitih mobilnih aplikacija. Svi su htjeli ili napraviti svoju aplikaciju kako bi postali planetarno popularni ili, nekad možda i nepotrebno, napraviti nativnu aplikaciju iz nule za svoju tvrtku. Rezultat toga je ogromna količina aplikacija koje se ne koriste i koje su zaboravljene.

„Mogli bismo ugasiti oko 95% mobilnih aplikacija u našem ekosustavu i nitko ne bi ni trepnuo. To je zato što u većini slučajeva, mobilni web može postići iste rezultate – jeftinije i fleksibilnije“ Aaron Strout, Marketing Land

U lipnju 2016. godine se na Android storeu nalazi okvirno oko 2,200,000 aplikacija dok ih se na Apple storeu nalazilo oko 2,000,000. Većina ovih aplikacija od iste su kompanije, ali su morale biti razvijane posebno za Android, a posebno za iOS uređaje. Velika količina verzija platformi i različitih uređaja za koju se mora razvijati nativna aplikacija dovodi do glavne boljke ovakvog pristupa. Razvijanje aplikacije za više platformi poskupljuje razvijanje same aplikacije, a broj verzija za koju se razvija aplikacija nerijetko znatno produžuje razvijanje mobilne aplikacija. Dosta je bitno uvidjeti uz ovako važne faktore (trošak i vrijeme razvoja), je li osobi koja stoji iza razvitka nativne aplikacije uistinu isplativo i razviti nativnu aplikaciju.

Uz to, količina aplikacija dovela je do toga da se par aplikacija izoliralo na vrhu kao aplikacije koje zauzimaju najveći postotak korištenja na mobilnom uređaju dok se većina ostalih pokrene možda jednom i nakon toga su izbrisane.



Slika 4. – statistički zapis postotka najkorištenijih aplikacija

Od 41,5% ostalih aplikacija 25% aplikacija su one koje su u svojoj kategoriji prve ili najpoznatije. Slično kao Google Search, ako si na vrhu ili si opće poznat, posjećenost će ti eksponencijalno rasti.

Višeplatformski pristup, kao i vrijeme i trošak razvoja jedna su od najvećih prednosti mobilnih web aplikacija. Održavanje same mobilne web aplikacije jednostavnije je jer se ne mora paziti na količinu verzija za koje je aplikacija razvijana već je baza univerzalna za sve platforme i verzije. Kao što je ranije navedeno nije ni ovakav način pristupanja mobilnom tržištu savršen jer web aplikacije moraju imati stalan pristup internetu, nešto su sporije od nativnih aplikacija i nemaju pristup svim funkcijama samog uređaja. Kako bi se tome doskočilo postoji treća vrsta mobilnih aplikacija, a to su Hibridne aplikacije.[2]

### 1.4.1 Hibridne aplikacije

Hibridne aplikacije su mobilne web aplikacije razvijane u HTMLu, CSSu i JavaScriptu kao i normalne web aplikacije i one se pokreću u browseru, ali je browser u pozadini implementiran u hibridnu aplikaciju i nije vidljiv (nemoguće je pristupati ostalim web stranicama kroz hibridnu aplikaciju). Koristeći alate kao što je PhoneGap ili Sencha touch moguće je vrlo lako razviti multiplatformsku hibridnu aplikaciju. Nakon što je aplikacija razvijena, može se pustiti u app store kako bi se mogla preuzeti na uređaj. Uz to, činjenica da se može koristiti svim funkcijama samog uređaja kroz dobro razrađen API ide im u prilog u odnosu na obične mobilne web aplikacije.

|  |  |  |
| --- | --- | --- |
|  | Mobilna web aplikacija | Nativna aplikacija |
| Funkcije samog uređaja | Ne može pristupiti svim funkcijama | Može im pristupiti |
| Nepovezanost s internetom | Mora biti povezana | Ne mora nužno biti povezana |
| Mogućnost otkrivanja aplikacije | Isto kao i na desktop webu | Bolje nego na webu kroz kategorizaciju |
| Brzina | Nešto sporija | Brža od web aplikacije |
| Instalacija | Nema instalacije | Zahtjeva instalaciju |
| Održavanje | Jednostavnije | Kompliciranije |
| Platformska nezavisnost | Nezavisna | Zavisna |
| Troškovi izrade | Manji troškovi | Veći troškovi |

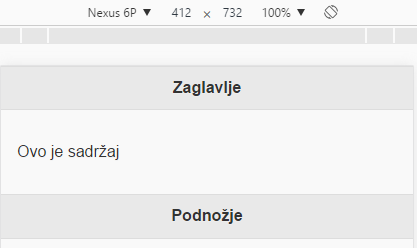
Tablica 1. – prednosti i mane ovisno o načinu razvijana mobilne aplikacije

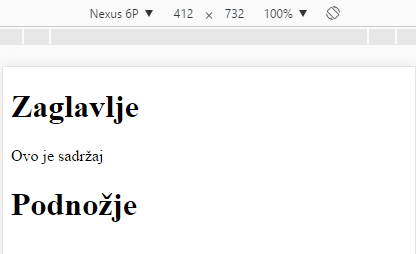
Hibridna aplikacija negdje je u sredini te preuzima najbolje od obje strane. Kako bi se bolje demonstrirale mogućnosti hibridne aplikacije razvijena je prezentacijska hibridna aplikacija o kojoj će se nešto kasnije raspravljati.

# 2. jQuery mobile

## 2.1 Što je jQuery mobile?

11. Kolovoza 2010, John Resig (kreator najpoznatijeg JavaScript librarya – jQuery) najavio je projekt jQuery Mobile. Zamišljen je kao framework za korisničko sučelje, ali je sam jQuery korišten kao alat kako bi sam framework učinio pogodnijim za mobilne uređaje. JQuery mobile na jednostavan način pruža mogućnost pretvaranja HTML-a (i CSS-a) u stranicu pogodnu za mobilni uređaj. Na primjeru se vidi kako uz jednostavnu implementaciju par jQuery mobile naredbi (korištenjem čistog HTML-a) web stranica izgleda puno preglednije na mobilnim uređajima.

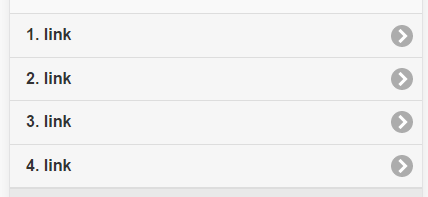


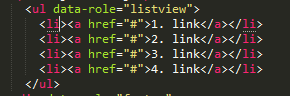


Slika 5. – usporedba prikaza stranice na mobilnim uređajima (slijeva – bez jQuery mobilea, zdesna – uključen jQuery)

Za razliku od ostalih frameworka jQuery Mobile je fokusiran isključivo na HTML. Zapravo, za framework usko povezan s jQueryem i JavaScriptom jako puno stvari se može postići bez korištenja i linije JavaScripta. Sencha Touch je alternativa jQuery Mobileu koji se koristi JavaScriptom kako bi postigao responzivnost i bolje pripremio sadržaj za pregled na mobilnom uređaju. JQuery mobile je puno usmjereniji ljudima koji bolje barataju HTMLom nego JavaScriptom. Jquery Mobile poboljšava web stranicu i nadopunjuje ju (povećava gumbe za submit/reset forme, provodi jasnu distinkciju između strukture stranice, omogućava lijepe prijelaze između stranica putem AJAX-a itd.) što uvelike olakšava rad na manjim uređajima.

Način na koji funkcionira je da se putem **data**atributa (još jedan od noviteta HTML5) označe elementi stranice. Dalje jQuery mobile iz tog atributa zaključi koja je svrha tog elementa i prilagođava ga mobilnim uređajima.

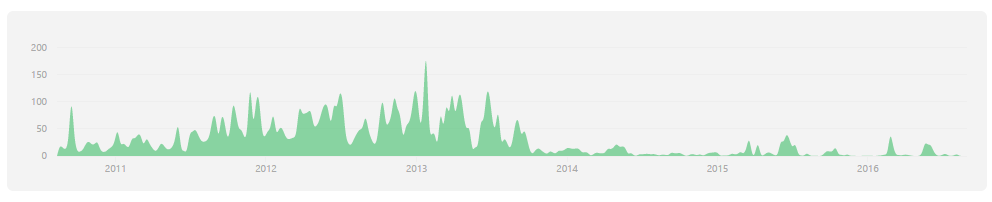




Slika 6. – primjer korištenja data atributa

Cijeli framework je opensource, kao i jQuery. Ako mislite da nedostaje neka funkcionalnost vrlo lako ju je commitati na jQuery Mobile projekt, a ukoliko vam se ne sviđa kako određeni dio funkcionira isto tako je taj dio lako izmjeniti.[3]

Nažalost, zadnja nadogradnja jQuerya mobile dogodila se 2013. godine i od tada jQuery mobile stoji na mjestu. Verzija 1.5 koju su na velika vrata najavljivali kasni više od godinu dana i nije izgledno kada će izaći.Uz to najnovija verzija jQuerya, verzija 1.4.5, ne funkcionira sa verzijama jQuerya iznad 2.x.x (trenutna jQuery verzija je 3.1.0). Prema navodima jednog od jQuery developera to je iz razloga što se jQuery Mobile i jQuery UI (koji isto nije nadograđen neko vrijeme) spajaju u jedan framework.[4]



Slika 7. – doprinosi master branchu na službenom jQuery Mobile githubu

## 2.2 Alternative za jQuery Mobile

U to isto vrijeme razvilo se par alternativnih frameworka koji već sada bolje funkcioniraju od jQuery mobilea. Iz tog razloga u nastavku prolazimo kroz najbolje alternativne frameworke kako bi pregled web stranica na mobitelima bio čim bolji i intuitivniji. Najpoznatiji mobile frameworci su trenutno Ionic framework, OnsenUI, gore navedeni Sencha Touch, Framework 7 i Mobile Angular UI.

Ionic je po popularnosti trenutačno vodeći framework za hibridne aplikacije jer je u samom frameworku već implementirano Cordova okružje te je vrlo lako prebaciti običnu mobilnu web aplikaciju u nativnu aplikaciju. Framewok je besplatan i nadogradnje su konstante i idu ukorak s tehnologijom i novim zahtjevima zajednice. Zajednica je s druge strane velika i nije teško pronaći materijale za korištenje ovog frameworka. Koristi se s Angular JS frameworkom u pozadini i to je izvor snage koju ovaj framework sadrži.

OnsenUI je noviji framework koji je odmah zasjeo mjesto iza Ionic frameworka i postao glavni konkurent za preuzimanje tržišta. Open source je i izdaje se pod Apache licencom, a u pozadini se vrti isto na Angular JS-u. Za ljubitelje jQuerya dolazi s osnovnim jQuery paketima i korisniku se daje na izbor ukoliko želi nastaviti razvijati hibridnu aplikaciju u jQueryu. OnsenUI sadrži veliku kolekciju komponenti koje su responzive i spremne za korištenje odmah. Na jednostavan način se kroz PhoneGap i Cordovu prebaci u nativnu aplikaciju.

Ostaje pitanje jesu li mobilni frameworci sa svim svojim manama uistinu potrebna komplikacija za razvoj hibridne aplikacije? Kroz gotove responzivne frameworke kao što su megapopularni Bootstrap i nešto manje popularni Foundation vrlo je lako razviti hibridnu aplikaciju kroz sam PhoneGap ili direktno Apache Cordovu. O ovim softwearima raspravljati će u sljedećem poglavlju.[5]

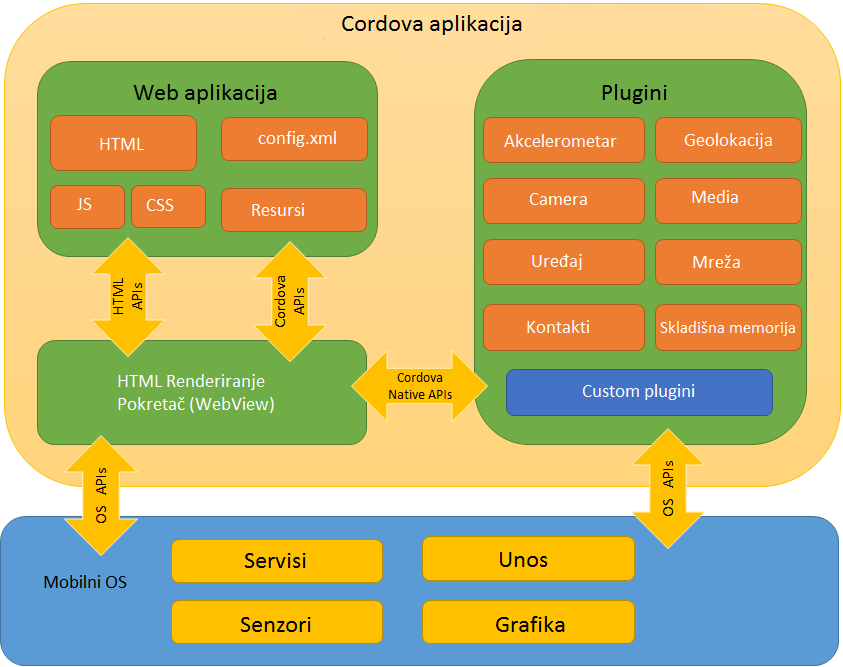
# 3. Softweari za razvoj hibridnih mobilnih aplikacija

## 3.1 Apache Cordova

Open source framework za razvoj hibridne mobilne aplikacije koji nam dopušta da uz korištenje standardnih web tehnologija, HTML5, CSS3 i JavaScript, razvijemo hibridnu mobilnu aplikaciju neovisno o platformi. Aplikacija se vrti unutar wrappera targetiranog po platformi i oslanja se na API konekcije zadane standardom kako bi se pristupilo nekim funkcionalnostima samog uređaja.

### 3.1.1 Arhitektura

Postoji par važnih komponenata za cordova aplikaciju.



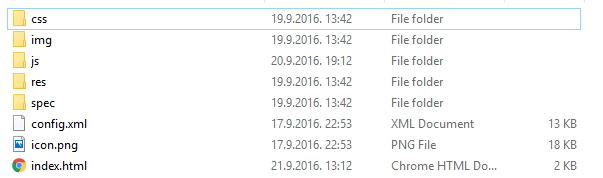
Slika 8. – apstraktni prikaz arhitekture cordova aplikacije

### 3.1.2 Webview

Cordova aplikacije su obično implementirane kao WebView zasnovan na browseru unutar nativne mobilne aplikacije. Cordova ga koristi kako bi implementirala aplikaciju na platformu i kroz njega pristupila Apijima za korištenje funkcionalnosti samog uređaja.

### 3.1.3 Web app

Mjesto u kojem se nalazi sav naš kod jednako kao da razvijamo običnu web aplikaciju.



Slika 9. – www folder tj. web app folder

Web app se nalazi u folderu www u rootu projekta. Kod instalacije, sastoji se od foldera css, img, js, res, spec mapa te config.xml, icon.png i index.html datoteka.Prva 3 foldera imaju dobro poznatu svrhu dok res folder služi kao spremište za splashscreenove i ikone. Spec folder je folder u kojem su sadržani testovi za JavaScript, a config.xml file je konfiguracijski file za aplikaciju i pruža određene informacije o aplikaciji (npr. odgovara li aplikacija na rotaciju uređaja)

### 3.1.4 Plugini

Plugini su integralni dio cordovinog ekosustava. Pružaju sučelje kojim na vrlo lak način nativne komponenta komuniciraju s cordovinom aplikacijom. Putem JavaScripta na vrlo lak se način poziva cordovin API koji u konačnici poziva nativne funkcije samog uređaja. To je korisno kada se razvija za više platformi jer cordovin plugin zna o kojoj se platformi radi i pozove funkciju te platforme. Recimo da ne razvijamo preko cordove trebali bismo posebno pozivati API za Android, a posebno za iOS. Cordova dolazi sa setom Core plugina tj pluginima za dohvat svih funkcija samog uređaja kao što su baterija, kamera, kompas, kontakti itd. Cordova pruža i mogućnost izrade vlastitih plugina ukoliko su nam takvi potrebni. Treba nadodati kako cordova ne dolazi sa već instaliranim pluginima nego se oni moraju ručno dodavati što u krajnjoj liniji ne zakrčuje projekt

### 3.1.5 Služenje cordovom

Cordova se koristi putem CLI-a koji se vrti na node.jsu. Vrlo je lako služiti se komandama samo je potrebno prije toga instalirati node.js. Kako bi instalirali cordovu potrebno je pokrenuti naredbu koja automatski downloada i instalira potrebne pakete:

$ npm install –g cordova

Nakon toga putem console dođemo do foldera u kojem želimo napraviti projekt i jednostavno unesemo komandu:

$ cordova create <ime\_projekta>

Platforme dodajemo putem sljedećih komandi:

$ cordova platform add android --save

$ cordova platform add browser --save

Kako bi buildali svoju aplikaciju za sve platforme:

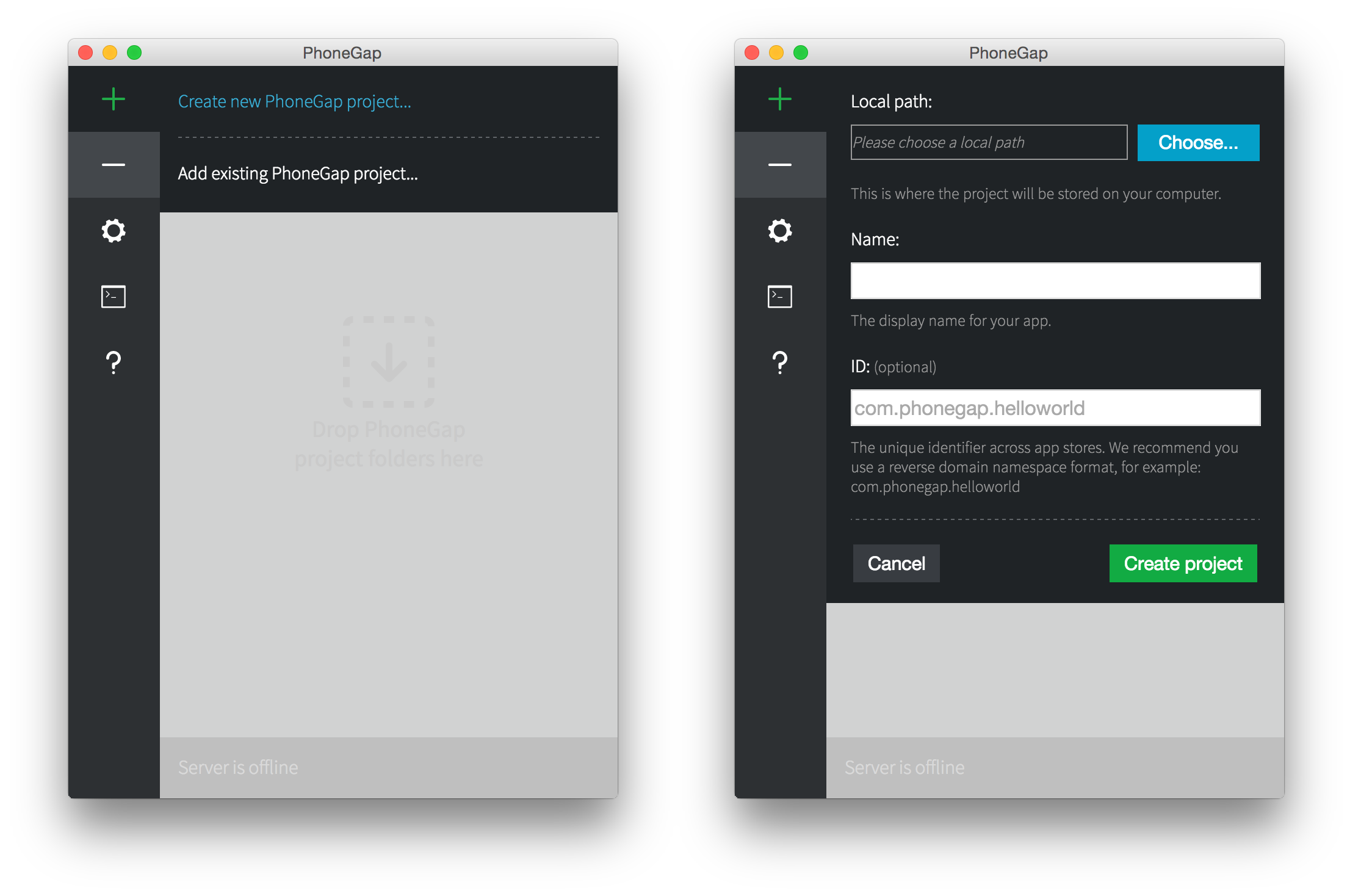
$ cordova build

I nakon toga, ako je sve prošlo u redu, jednostavnom naredbom pokrenemo aplikaciju:

$ cordova run <ime\_platforme>

## 3.2 Adobe PhoneGap

PhoneGap je još jedan od alata koji se koristi kako bi se što jednostavnije mobilna web aplikacija prebacila na mobilni uređaj. Open source je distribucija Apache Cordove i u neku ruku olakšava cijeli proces tranzicije mobilne web aplikacije u hibridnu aplikaciju. Razlikuje se od Apache Cordove po tome što dolazi s Desktop aplikacijom, mogućnošću buildanja jednostavne aplikacije za bilo koju platformu putem jednostavnog online sučelja pod nazivom Adobe PhoneGap Build i PhoneGap developer appom putem kojeg se na jednostavan način testira funkcionalnost aplikacije na mobilnom uređaju.[6]



Slika 10. – izgled korisničkog sučelja PhoneGap desktop aplikacije

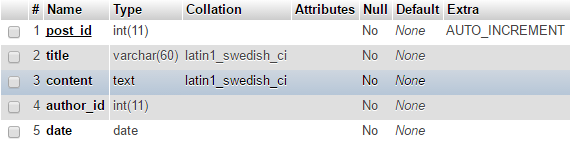
# 4. Prezentacijska hibridna aplikacija

Prezentacijska aplikacija zamišljena je da funkcionira kao facebook u smislu da ima glavnu stranicu na kojoj se nalaze „postovi“ svih korisnika. Kako bi se aplikaciji pristupilo potrebno se registrirati. Nakon registracije korisnik može napisati novi post, ali samo na svojem profilu ili promijeniti sliku na način da se koriste osnovne funkcije uređaja to jest kamera. Korisnik je u mogućnosti i pregledavati tuđe profile, ali na tuđim će profilima pronaći samo njihovu profilnu sliku i ispis svih postova.

Kroz cijeli rad praćena je i ideja o „Jednom webu“ kroz korištenje mobilnog frameworka jQuery mobile te prebacivanje u nativnu aplikaciju korištenjem Apache Cordove.

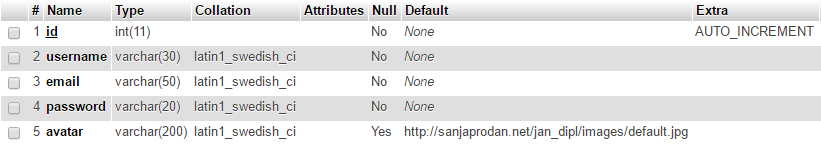
## 4.1 Backend

Kako je cijela aplikacija mobilna ne može sadržavati serverske skripte to jest PHP u ovom slučaju. Da bi se doskočilo ovom problemu cijela komunikacija sa serverom mora biti asinkrona tj nakon generiranja HTML-a šalje se AJAX request. Zamišljeno je da baza ima dvije tablice: users i posts. Struktura posts tablice navedena je na slici ispod:



Slika 11. – struktura posts tablice

Struktura users tablice navedena je na slici ispod:



Slika 12. – struktura users tablice

Kada se stranica aplikacije loada, putem AJAX requesta poziva se serverska skripta koja u konačnici vraća željeni rezultat u JSON formatu.



Slika 13. – primjer AJAX poziva

U slici iznad naveden je primjer ispisa svih postova. U *url* varijablu spremljena je putanja do index.php filea u kojem se svi postovi putem sql querya dohvaćaju i pomoću funkcije *json\_encode()* prevode u JSON format. Nakon toga kada se vrate podaci u našu skriptu na aplikaciji svaki post se pomoću jQuery funkcije *append()* nadodaje u element s id-jem 'posts'. Ovaj proces se ponavlja više puta, a ovisno o podacima koje moramo dohvatiti iz baze i s kojim uvjetima request može biti POST ili GET.

Kako bi započeli sesiju, a ne možemo pristupiti superglobalnoj varijabli $\_SESSION, koristi se još jedan od HTML5 noviteta, a to je Storage object – window.sessionStorage. Dok window.localStorage funkcionira kao i cookie (nema rok trajanja), window.sessionStorage je simulacija sessiona i traje dok korisnik ne zatvori aplikaciju ili tab u browseru. Primjer naredbe kojom postavljamo window.sessionStorage attribut unutra login.html dokumenta:

var username = $("#username").val();

window.sessionStorage.setItem("username", username);

Upload fotografije se pokazao kao najkompliciraniji dio zadatka. Bilo je potrebno povezati cordovin camera plugin s filetransfer pluginom zaslužnim za upload. Navigator.camera je objekt s metodom *getPicture()* koji kao parametre prima success i error callbackove te options objekt. Success callback se poziva ukoliko metoda uspije napraviti svoju zadaću, a error callback se poziva ukoliko metoda ne uspije.

navigator.camera.getPicture(onCapturePhoto, onFail, {

quality: 100,

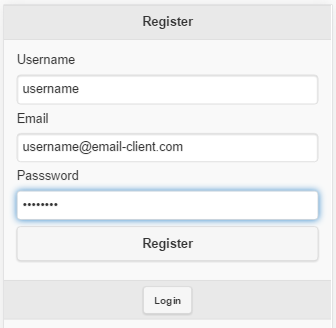
destinationType: destinationType.FILE\_URI

});

Nakon što je fotografija uspješno zabilježena poziva se success callback (*onCapturePhoto*() u ovom slučaju) u kojem se automatski provodi upload fotografije. Kreira se FileTransfer objekt te putem *upload()* metode se fotografija uploada na net. Upload metoda prima kao prvi parametar fileURI koji se automatski prosljeđuje success callbacku metode getPicture. FileURI je uniformni identifikator fotografije na samom uređaju i iz njega se povlači resurs za upload. Drugi parametar upload metode je url na skriptu koja procesuira upload na serverskoj strani. Nakon toga su success i error callbackovi te options objekt u koje se prosljeđuju post varijable, ime datoteke, key vrijednost za superglobalno polje $\_FILES u koji se file pošalje i iz kojega skripta na serverskoj strani može pristupiti samoj fotografiji. Ja sam naišao na problem gdje se file nije uploadao, ali je success callback bio pozvan. To se događalo iz razloga što je chunckedMode opcija uključena po osnovi. U principu ona razdvaja datoteku, u ovom slučaju fotografiju i šalje ju u paketima. Tu je dolazilo do greški što je rezultiralo u praznim $\_FILES i $\_POST superglobalnim poljima.

## 4.2 Frontend

Za postizanje što bolje preglednosti na mobilnim uređajima, jQuery mobile uključen je u projekt otpočetka i uz neke preinake aplikacija je vrlo jednostavno prilagođena za mobilne uređaje. U nastavku je dan primjer kako jedna HTML forma izgleda u jQuery mobileu. Bitno je naglasiti kako nema klasičnog potvrđivanja forme već se simulira gumb za potvrdu i šalje se AJAX request.



Slika 14. - primjer forme za registraciju

# Zaključak

Neupitna je važnost i prisutnost pametnih telefona u današnjem svijetu. Gotovo svatko oko nas posjeduje pametni telefon i koristi se mnoštvom različitih aplikacija. Iako se nakon inicijalnog buma količine aplikacija na tržištu situacija malo stabilizirala i glavni konkurenti po kategorijama stoje čvrsto na vrhu, tu i tamo pojavi se mobilna aplikacija koja dobije svojih pet minuta. Takve aplikacije su prava rijetkost, ali svi teže k tome da baš njihova bude jedna od takvih. Zbog toga danas imamo na stotine tisuća aplikacija koje samo stoje u sjeni i u potpunosti su nekorištene.

Iako ni u jednu ruku nisam stručnjak za tematiku razvoja mobilnih aplikacija, kroz ovu temu dobio sam dobar dojam o kompleksnosti razvoja mobilnih aplikacija za mnoštvo uređaja koji se danas nalaze na tržištu. Ukoliko imate u planu razvijati nativnu aplikaciju trebali biste se zapitati treba li vam uistinu? Ako korisnici pristupaju vašim web stranicama samo kako bi provjerili par informacija, ulaganje u nativnu aplikaciju moglo bi ispasti loša investicija. S druge strane mogućnost pružanja offline informacija vašim korisnicima može biti od ključne važnosti u nekim trenucima. Jako je bitno dobro definirati treba li vam mobilna aplikacija i isplati li ju se razvijati.

Hibridne aplikacije odlična su alternativa skupim razvojnim procesima nativnih aplikacijama. Jedina negativna strana hibridnih aplikacija je što u većini slučajeva moraju imati stalnu konekciju s internetom.

Ostaje vidjeti hoće li nativne aplikacije moći održati korak s uvijek nadograđivanim i brzo razvijajućim web tehnologijama.

# Popis literature

|  |  |
| --- | --- |
| [1] | *The Two Flavors of a ‘One Web’ Approach: Responsive vs. Adaptive* (2013), Dostupno na: [https://www.wired.com/2013/05/the-two-flavors-of-a-one-web- approach-responsive-vs-adaptive/](https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20approach-responsive-vs-adaptive/) [18.09.2016.] |
| [2] | *Mobile: Native Apps, Web Apps, and Hybrid App* (2013), Dostupno na : <https://www.nngroup.com/articles/mobile-native-apps/> [20.09.2016.] |
| [3] | Camden R, Matthews A, 'jQuery Mobile Web Development Essentials', Packt Publishing (2013), pp. 1-30 |
| [4] | *Why jQuery mobile is not relevant any more* (2016), Dostupno na: <http://www.gajotres.net/why-jquery-mobile-is-not-relevant-any-more/> [19.09.2016.] |
| [5] | *10 Best Hybrid Mobile App UI Frameworks: HTML5, CSS and JS* (2016), Dostupno na: <http://noeticforce.com/best-hybrid-mobile-app-ui-frameworks-html5-js-css> [20.09.2016.] |
| [6] | Oehlman D, Blanc S, 'Pro Android Web Apps', Apress (2011), pp. 193-221 |

# Popis priloga

[Slika 1. - podijeljenost tržišta po platformama pametnih telefona 5](#_Toc462349949)

[Slika 2. - ilustracija responzivnog prikaza na različitim uređajima 7](#_Toc462349950)

[Slika 3. - ilustracija templating sustava ovisno o pristupu 8](#_Toc462349951)

[Slika 4. – statistički zapis postotka najkorištenijih aplikacija 10](#_Toc462349952)

[Slika 5. – usporedba prikaza stranice na mobilnim uređajima (slijeva – bez jQuery mobilea, zdesna – uključen jQuery) 12](#_Toc462349953)

[Slika 6. – primjer korištenja data atributa 13](#_Toc462349954)

[Slika 7. – doprinosi master branchu na službenom jQuery Mobile githubu 13](#_Toc462349955)

[Slika 8. – apstraktni prikaz arhitekture cordova aplikacije 15](#_Toc462349956)

[Slika 9. – www folder tj. web app folder 16](#_Toc462349957)

[Slika 10. – izgled korisničkog sučelja PhoneGap desktop aplikacije 18](#_Toc462349958)

[Slika 11. – struktura posts tablice 19](#_Toc462349959)

[Slika 12. – struktura users tablice 19](#_Toc462349960)

[Slika 13. – primjer AJAX poziva 20](#_Toc462349961)

[Slika 14. - primjer forme za registraciju 21](#_Toc462349962)

[Tablica 1. – prednosti i mane ovisno o načinu razvijana mobilne aplikacije 11](#_Toc462350036)