

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleksander Gregorka

# **Platforma Trafika v HTML5**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE

RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Peter Peer

Ljubljana, 2012

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

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*



Št. naloge: 00306/2012

Datum: 15.04.2012

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

Kandidat: **ALEKSANDER GREGORKA**

Naslov: **PLATFORMA TRAFIKA V HTML5**  
**PLATFORM NAMED TRAFIKA IN HTML5**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

S pomočjo sodobnih tehnologij razvijte mobilno spletno aplikacijo prilagojeno večim mobilnim in namiznim napravam, ki služi kot elektronski bralnik digitalnih vsebin, kot so revije in časopisi. Pri razvoju se oprite na funkcionalnosti, ki jih ponuja že obstoječa implementacija v obliki domorodne mobilne aplikacije. Analizirajte razlike med razvojem spletnih in domorodnih aplikacij na primeru platforme Trafika.

Mentor:

doc. dr. Peter Peer

Dekan:

prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Aleksander Gregorka, z vpisno številko **63070103**, sem avtor diplomskega dela z naslovom:

*Platforma Trafika v HTML5*

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne 25. junija 2012

Podpis avtorja:

*Na tem mestu se zahvaljujem staršem, dekletu Tanji, prijateljem in vsem ostalim, ki so me tekom študija spodbujali in mi stali ob strani.*

*Zahvalil bi se mentorju doc. dr. Petru Peeru, ki mi je z nasveti pomagal pri izdelavi tega diplomskega dela.*

*Zahvaljujem se tudi podjetju Kliko d.o.o., ki mi je omogočilo izdelavo projektne delo diplomske naloge.*

# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Opis problema . . . . .	1
1.2	Razlogi za razvoj . . . . .	2
<b>2</b>	<b>Sorodne rešitve</b>	<b>3</b>
2.1	Zgodovina e-bralnikov . . . . .	3
2.2	Mobilno založništvo . . . . .	4
2.3	Platforma Trafika in sorodne rešitve . . . . .	4
2.3.1	Trafika . . . . .	4
2.3.2	Zinio . . . . .	5
2.3.3	BlueToad, Inc. . . . .	6
<b>3</b>	<b>Primerjava tehnologij</b>	<b>9</b>
3.1	Vodilni mobilni operacijski sistemi . . . . .	9
3.1.1	iOS . . . . .	9
3.1.2	Android . . . . .	10
3.1.3	Windows Phone 7 . . . . .	11
3.2	Razvoj aplikacij za iOS . . . . .	12
3.3	Razvoj aplikacij za Android . . . . .	14
3.4	Razvoj aplikacij za Windows Phone 7 . . . . .	15

3.5	Primerjava med razvojem domorodnih in spletnih aplikacij	17
<b>4</b>	<b>Implementirana rešitev</b>	<b>21</b>
4.1	Uporabljene tehnologije in pristopi	21
4.1.1	HTML	21
4.1.2	HTML5	22
4.1.3	CSS in CSS3	24
4.1.4	JavaScript in jQuery	27
4.1.5	Odzivno spletno oblikovanje	28
4.2	Implementacija spletnega portala	31
4.2.1	Google App Engine	31
4.3	Implementacija spletnega bralnika revij	38
4.4	Optimizacija z uporabo novih tehnologij	42
4.5	Težave pri razvoju	45
<b>5</b>	<b>Rezultati</b>	<b>47</b>
5.1	Razlika v času implementacije za različne platforme glede na funkcionalnost	48
5.2	Pridobljen del trga	48
<b>6</b>	<b>Zaključek</b>	<b>51</b>
6.1	Možne izboljšave	52
	<b>Dodatek A Skripta za nadzor lokalnega pomnilnika</b>	<b>53</b>
	<b>Literatura</b>	<b>56</b>

# Povzetek

Cilj diplomskega dela je s pomočjo sodobnih tehnologij razviti mobilno spletno aplikacijo prilagojeno več mobilnim in namiznim napravam, ki služi kot elektronski bralnik digitalnih vsebin kot so revije in časopisi. Na začetku smo na kratko predstavili zgodovino elektronskih bralnikov ter tehnologije, ki smo jih uporabili pri razvoju. Pri razvoju smo se oprli na funkcionalnosti, ki jih ponuja že obstoječa implementacija v obliki domorodne (angl. native) mobilne aplikacije. Implementacijo spletne aplikacije smo razdelili na spletni portal in bralnik revij in vsakega podrobneje opisali ter predstavili načine optimizacije. V diplomskem delu smo se posvetili tudi analizi razlik med razvojem spletnih in domorodnih aplikacij na splošno, razlike pa smo predstavili na primeru platforme Trafika.

## Ključne besede:

platforma Trafika, HTML5, CSS3, mobilne spletne aplikacije, odzivno spletno oblikovanje

# Abstract

The goal of this thesis is to develop a mobile web application designed to fit most of the mobile and desktop environments by utilizing some of the newest web technologies. The application is used as an electronic reader for digital content, such as magazines and newspapers. We briefly describe the history of electronic readers and some of the technologies that were used in the development. We will try to implement the same functionality that is already present in the native application. The implementation itself is split into two parts: the development of a web portal and of an electronic reader. We describe each part and present the work we did on the reader optimization. We compare the differences between native mobile application development and web mobile development and also present those differences on a real life implementation example – the Trafika platform.

## Key words:

Trafika platform, HTML5, CSS3, mobile web applications, responsive web design

# Poglavje 1

## Uvod

### 1.1 Opis problema

Število ljudi, ki za dostop do spleta uporabljajo izključno mobilne naprave se po zadnjih podatkih hitro povečuje, kar je sicer logična posledica masovne prodaje tako imenovanih pametnih telefonov in tabličnih računalnikov. V zadnjem letu je tako delež mobilnih dostopov do spleta po vsem svetu narasel za okoli 4% in danes obsega 10.4% vseh dostopov do spleta [10]. Lastniki mobilnih naprav v 95% primerih svojo napravo z dostopom do spleta uporabljajo za socialno interakcijo, prav tako pa v povprečju 95% vseh uporabnikov dostopa do najnovejših novic z različnih področij [11].

Ponudniki novic in drugih digitalnih vsebin se tega trenda zavedajo, zato se poslužujejo razvoja mobilnih aplikacij, ki bi branje njihovih vsebin čim bolj približale uporabniku. Ker je mobilni trg čedalje večji, si lahko mobilni založniki privoščijo uporabniku ponuditi tudi različne vrste poslovnih modelov. Svoje vsebine lahko recimo vnovčijo kot enkratni nakup številke, ponudijo lahko naročnine za različna časovna obdobja ali pa se odločijo za brezplačne vsebine s prikazom oglasov.

Stvari so na prvi pogled obetavne, vendar sta razvoj in vzdrževanje mobilnih aplikacij za več platform draga, velik problem pa predstavlja tudi

dejstvo, da gre približno tretjina zaslužka tretji osebi. Ta je v tem primeru distributer aplikacije, ki jo prek svoje spletne trgovine z aplikacijami ponuja uporabnikom in jim tudi omogoča plačevanje za vsebine.

Na srečo obstaja druga možnost ponujanja spletnih digitalnih vsebin uporabnikom, ki z nekaj kompromisi izključuje tretjo osebo. To je izgradnja spletne mobilne aplikacije prilagojene za vse platforme. S tem se izognemo razvoju več aplikacij, hkrati pa je tudi vzdrževanje lažje, saj so spremembe na vseh napravah vidne takoj, ne da bi uporabnik moral namestiti posodobitve.

Cilj diplomske naloge je razviti mobilno spletno aplikacijo prilagojeno več mobilnim napravam, ki služi kot elektronski bralnik za digitalne vsebine kot so revije in časopisi. Posvetili se bomo tudi primerjavi razvoja domorodnih mobilnih aplikacij in spletnih mobilnih aplikacij v smislu tehnologije, časa in težav pri razvoju.

## 1.2 Razlogi za razvoj

Eden izmed zgoraj omenjenih ponudnikov digitalnih vsebin je tudi podjetje Kliko d.o.o., ki v svoji mobilni aplikaciji Trafika ponuja veliko število slovenskih revij in časopisov. Ker je odziv uporabnikov na mobilno aplikacijo za operacijski sistem iOS pozitiven, smo se v podjetju odločili razviti še aplikacijo za operacijski sistem Android ter spletno različico mobilne aplikacije, ki bi pokrila ostale operacijske sisteme. Prednost razvoja spletne aplikacije Trafika je tudi v tem, da bi v njej lahko objavili revije ponudnikov ki so izrazili interes, vendar jih zaradi nedovoljenih vsebin v domorodni aplikaciji Trafika za iOS ne moremo objaviti. Podjetje Apple ima namreč zelo stroge pogoje glede vsebin, ki so na voljo v aplikaciji.

# Poglavje 2

## Sorodne rešitve

### 2.1 Zgodovina e-bralnikov

E-bralnik je elektronska mobilna naprava, ki je primarno namenjena branju knjig in publikacij. Glavni prednosti takšne naprave sta velika količina shranjenih knjig, ali publikacij brez dodane fizične teže ter enostavno nalaganje novih gradiv.

Zgodovina E-bralnikov se začne leta 1992, ko je Sony izdal Data Discman. Leta 2004 Sony izda prvi e-bralnik z e-ink tehnologijo, vendar se uporaba e-bralnikov vse do leta 2007 (ko podjetje Amazon v Ameriki izda Kindle) ne razširi. Po letu 2010 pa število uporabnikov e-bralnikov z izidom Applevega iPada le še narašča.

Vsaka naprava, ki je na ekranu sposobna prikazovati tekst je v bistvu e-bralnik, vendar jih lahko ločimo po tipu zaslona. Tablični računalniki imajo večinoma zaslone s hitrejšo frekvenco osveževanja in so tako primernejši za interakcijo, e-bralniki s tehnologijo e-ink pa bolj posnemajo pravi papir, saj njihovi zasloni v nasprotju s tabličnimi računalniki ne oddajajo svetlobe ampak jo odbijajo. E-ink tehnologija se bolje obnese pri direktni izpostavljenosti sončni svetlobi, ponaša pa se tudi z zelo majhno porabo, saj lahko tekst na zaslonu zadrži tudi brez porabe električne energije.

Prihodnost e-bralnikov bo najverjetneje usmerjena v izdelavo upogljivi-

vih zaslonov, ki bodo lahko simulirali časopis, vendar s prednostjo avtomatskega osveževanja dnevne številke.

## 2.2 Mobilno založništvo

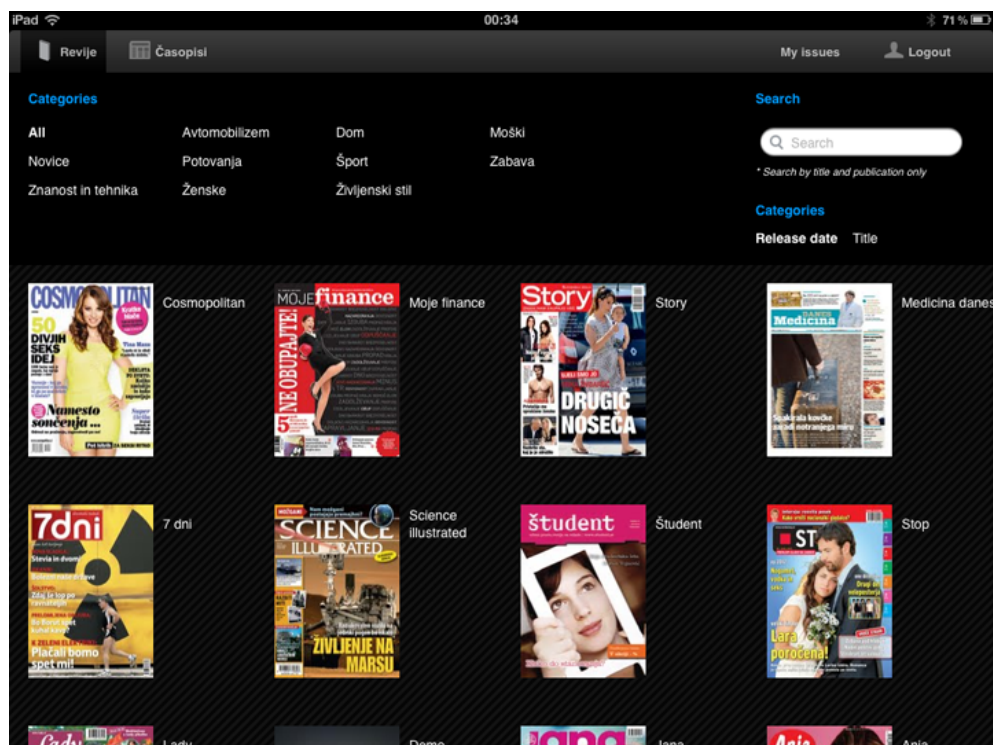
Trenutni trendi kažejo, da vedno več ljudi namesto običajnih knjig izbere njihovo digitalno različico. Razlogov za to je več, bodisi je ljudem všeč takojšnja dostava digitalne vsebine, bodisi jim je všeč ekološka plat mobilnega založništva. K temu je veliko pripomogel razvoj mobilnih naprav, ki omogočajo ogled digitalnih vsebin kjerkoli in kadarkoli, ne da bi s seboj nosili dodatno težo.

V vlogi elektronskega založnika trenutno vodi Amazon [13] s svojo spletno knjigarno Kindle Store, v kateri je bilo ob zagonu okoli 88.000 digitalnih naslovov, danes pa jih je že 1.314.655 [12]. Resnejši tekmeci so Barnes and Noble, Sony Reader in Apple iBookstore.

## 2.3 Platforma Trafika in sorodne rešitve

### 2.3.1 Trafika

Trafika je platforma za distribucijo digitalnih vsebin, ki jo je razvilo slovensko podjetje Kliko d.o.o.. Založnikom preko enostavnega administracijskega vmesnika omogoča nalaganje novih številok njihove revije in spremljanje statistike prenosov in nakupov posamezne številke. Založniki strani ene številke na strežnik naložijo v formatu PDF in številko opremijo z naslovom, opisom ter sliko naslovnice. Ti podatki se na strežniku zapakirajo in se, ko uporabnik v iPad aplikaciji Trafika (slika 2.1) kupi številko, distribuira na napravo. Trafika podpira tudi interaktivne številke, kar pomeni, da lahko založniki znotraj posamezne številke definirajo območja, na katerih se ob dotiku zgodijo akcije, na primer pošiljanje elektronske pošte, predvajanje videa ipd. Za potrebe evidence aktivnih naročnin in kupljenih

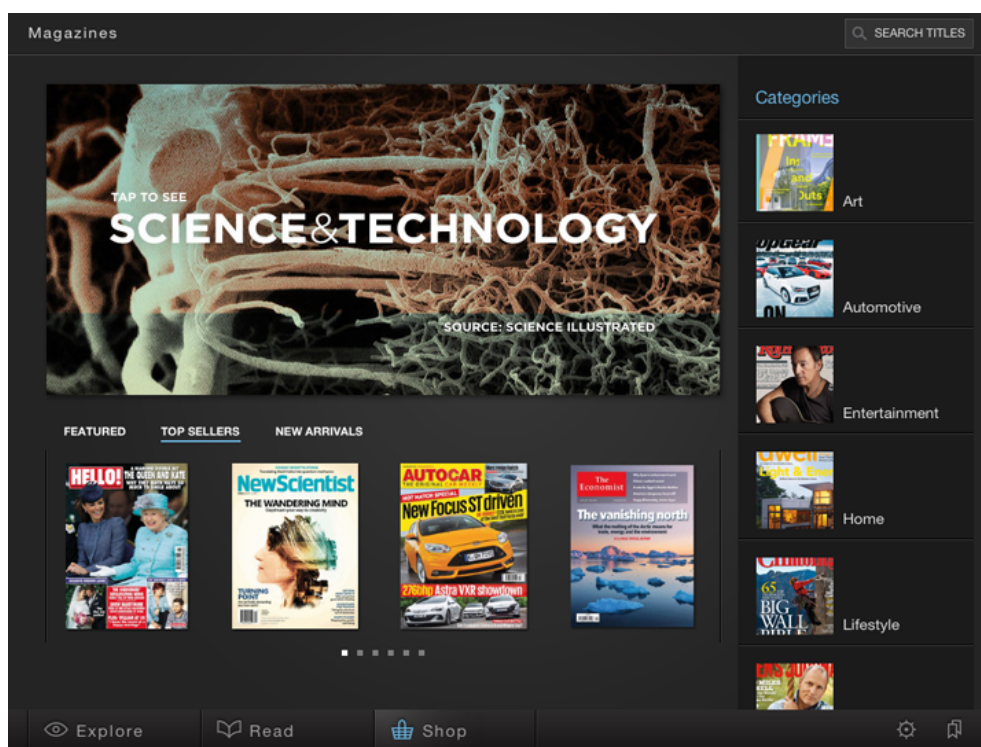


Slika 2.1: Izgled iPad aplikacije platforme Trafika

številke se morajo uporabniki znotraj aplikacije registrirati in prijaviti v sistem. Vse številke, ki jih je uporabnik že prebral tudi ostanejo na napravi, zato jih lahko prebira tudi brez internetne povezave.

### 2.3.2 Zinio

Ameriško podjetje Zinio je globalno eden največjih digitalnih kioskov s pestro izbiro revij, knjig in katalogov. Zinio ponuja branje digitalnih vsebin na iOS (slika 2.2), Android in Web OS napravah z domorodnimi aplikacijami ter tudi namiznih računalnikih z Adobe AIR aplikacijo. Skupno je v Zinio trgovini na voljo več kot 1700 revij od 350 založnikov, več kot 160 katalogov, letno pa Zinio več kot 3,5 milijonom naročnikov proda več kot 500.000 digitalnih vsebin [20]. Zinio svojim strankam ponuja orodja za nalaganje revij v PDF formatu. Založniki lahko številko opremijo z

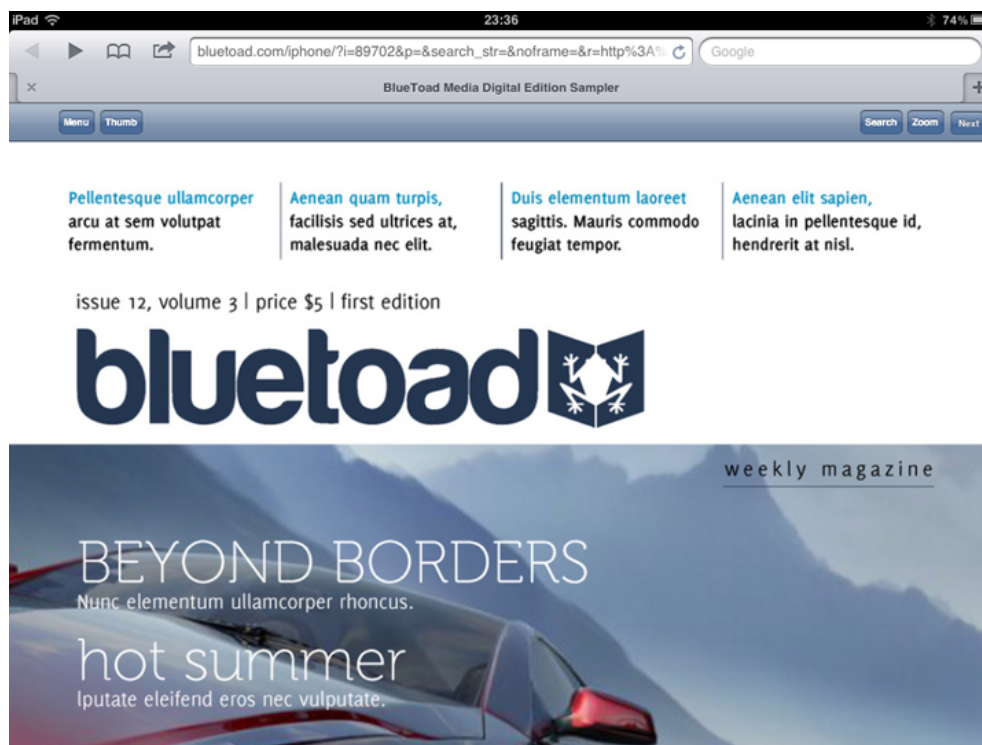


Slika 2.2: Izgled iPad aplikacije platforme Zinio

naslovom in opisom, v priročnem urejevalniku Zinio Fusion pa lahko dodajajo tudi multimedijske vsebine [21]. Uporabniki se morajo za nakup, tako kot pri Trafiki, prijaviti v sistem. S tem jim je omogočeno branje ene kupljene številke na različnih napravah. Aplikacija podpira tudi branje brez internetne povezave.

### 2.3.3 BlueToad, Inc.

Podobno kot Zinio je BlueToad, Inc. eden izmed večjih distributerjev digitalnih vsebin, saj imajo v trgovini kar 5.000 založnikov s skupno več kot 10.000 naslovi [14]. Njihove vsebine si lahko ogledamo na spletu, na voljo pa so tudi iPad aplikacije, ki jih BlueToad prilagodi naročnikom. Sami ponujajo tudi spletni bralnik revij, ki je po funkcionalnosti podoben našemu (slika 2.3) – številke revij iz PDF formata pretvori v slike in nam



Slika 2.3: Izgled spletnega bralnika platforme BlueToad na iPadu

jih prikaže. V primerjavi z našim bralnikom je ta počasnejši in ne podpira naprednejših gest. Podobno kot pri Trafiki in Ziniu založniki revije v PDF formatu naložijo na strežnik in jih opremijo s potrebnimi informacijami. Spletni bralnik ne podpira branja brez internetne povezave.



# Poglavje 3

## Primerjava tehnologij

### 3.1 Vodilni mobilni operacijski sistemi

#### 3.1.1 iOS

iOS, v prejšnjih različicah imenovan tudi iPhone Operating System, je mobilni operacijski sistem, ki ga je razvilo podjetje Apple za njihov mobilni telefon iPhone. Predstavljen je bil v začetku leta 2007 skupaj z mobilnikom iPhone. Najdemo ga tudi na drugih prenosnih napravah podjetja Apple, kot so na primer iPod Touch, iPad, Apple TV. Uporabniški vmesnik je preprosto zasnovan in je znan po tem, da je uporabniku zelo prijazen. Navigacija po telefonu je preprosta in sicer preko zaslona na dotik. Če želimo odpreti aplikacijo, se je preprosto dotaknemo s prstom, ob zaprtju pa pritisnemo na tipko na sprednji strani naprave.

V prvi četrtini leta 2012 je imela ta mobilna platforma 23% tržni delež na področju pametnih telefonov, takoj za Androidom (59%). Največjo zaslugo za takšen uspeh ima zagotovo Applova trgovina z aplikacijami App Store, kjer se trenutno nahaja že več kot 550.000 aplikacij. Veliko dobrih aplikacij lahko Applovi uporabniki naložijo tudi zastonj, pri ostalih pa je potreben nakup. Skupaj so pri Applu zabeležili že 25 milijard prenosov aplikacij [1], kar za razvijalce aplikacij pomeni zelo dobro priložnost, še

posebej, če aplikacijo ponudijo po ugodni ceni. Če je popularna, lahko razvijalec zasluži na ogromni masi ljudi. Ceno aplikacije lahko razvijalec določi na podlagi vnaprej pripravljenih cenovnih razredov. Od zaslužka vsake prodane aplikacije oziroma nakupa dobrine znotraj aplikacije same pa Apple vzame 30% delež [19].

Če želi fizična oseba postati razvijalec aplikacij na iOS platformi, mora kupiti licenco, ki stane 99\$ letno. Za to vsoto prejme brezplačen dostop do testnih različic iOSa in razvojnega okolja Xcode, lahko objavi neomejeno število aplikacij v trgovini App Store, poleg tega pa razvijalec brezplačno dostopa do velikega števila napotkov za razvijanje aplikacij, ki se nahajajo na Applovi spletni strani.

### 3.1.2 Android

Ta operacijski sistem temelji na Linuxu in je namenjen pametnim telefonom in tabličnim računalnikom. Z razvojem je pričelo podjetje Android Inc., leta 2005 pa ga je prevzelo podjetje Google. Trenutno operacijski sistem razvija združba 86 podjetij (razvijalci in proizvajalci strojne opreme, telekomunikacijska podjetja ipd.) z Googlom na čelu. Njihov cilj je razvijanje odprte platforme za mobilne naprave [3], zato je operacijski sistem Android v osnovi odprtokoden. Izgled uporabniškega vmesnika si proizvajalci telefonov ponavadi prilagodijo po meri, da ustvarijo unikatno izkušnjo (naprimer HTC Sense in Samsung TouchWiz). Navigacija po operacijskem sistemu se od iOSa malo razlikuje, saj je uporabniku poleg seznama aplikacij na voljo tudi glavni zaslon, na katerem se ponavadi nahajajo bližnjice za zagon aplikacij ipd. Za zagon aplikacije se preprosto dotaknemo njene ikone, za zaprtje pa pritisnemo tipko na sredini večine naprav. Android naprave se od iOS naprav razlikujejo tudi po tem, da imajo na sprednji strani več funkcijskih tipk. Dodani sta tipka za prikaz menija in tipka za vrnitev na prejšnji zaslon.

Verzija 1.0 je bila predstavljena septembra 2008, od takrat pa je Android hitro zrastel v operacijski sistem z največjim deležem na trgu mo-

bilnih naprav. Po podatkih s konca februarja 2012 je vsak dan na novo aktiviranih 850.000 novih Android naprav, skupno pa je vseh naprav, ki jih poganja Android več kot 300 milijonov [4].

Aplikacije za operacijski sistem Android si lahko uporabniki prenesejo preko distribucijske platforme Google Play (bivši Android Market), kjer je danes na voljo že več kot 500.000 aplikacij, skupno pa beležijo že več kot 15 milijard prenosov [5]. Ceno aplikacije razvijalec določi sam, vendar Google vzame 30% zaslужka vsake prodaje [18].

Razvijalska orodja ter odlična dokumentacija za razvoj Android aplikacije so zastoj, za objavo na platformi Google Play pa morajo razvijalci po novem plačati enkratni znesek v višini 25\$. Google je ta ukrep sprejel pred kratkim, da bi zvišal kakovost objavljenih aplikacij.

### 3.1.3 Windows Phone 7

Windows Phone 7 je operacijski sistem, ki ga razvija podjetje Microsoft. Je naslednik operacijskega sistema Windows Phone, vendar aplikacije niso združljive za nazaj. Pomembna novost pri novem sistemu je prehod na nov uporabniški vmesnik imenovan Metro, ki je značilen po postavitvi elementov v obliki ploščic.

Windows Phone 7 je bil uradno predstavljen februarja 2010, sicer pa je bil v razvoju od leta 2008. Ker so pri Microsoftu veliko časa namenili optimizaciji hitrosti sistema, je bil ta že ob predstavitvi zelo odziven in prijeten za uporabo, vendar je na ta račun izgubil nekaj uporabnih lastnosti modernih mobilnih operacijskih sistemov kot naprimer večopravnost.

Aplikacije, glasba in video so na voljo prek Microsoftove digitalne trgovine Windows Phone Marketplace, kjer je danes na voljo že več kot 92.000 aplikacij. Na dan je v povprečju dodanih okoli 250 novih aplikacij. 68% vseh aplikacij je zastoj, 22% je plačljivih, 10% pa je plačljivih z možnostjo preizkusnega obdobja [6].

V prvi četrtini leta 2012 je imel Windows Phone 7 / Windows Mobile 2,6% tržni delež, vendar se številka vzpenja zaradi poslovnega dogovora

med Microsoftom in finskim proizvajalcem mobilnih telefonov Nokia.

Razvijanje aplikacij je mogoče v okolju Visual Studio 2010, kjer se razvijalska orodja za Windows Phone namestijo kot dodatek. Razvijalec mora imeti na svojem računalniku nameščen operacijski sistem Windows Vista SP2 ali novejšo verzijo. Da lahko razvijalec svojo aplikacijo objavi na Windows Phone Marketplace, mora plačati letno članarino v višini 99\$ [7]. Razvijalec ceno svoje aplikacije določi sam, vendar Microsoft vzame 30% zaslужka vsake prodaje.

## 3.2 Razvoj aplikacij za iOS

Razvoj aplikacij za operacijski sistem iOS poteka v integriranem razvojnem okolju Xcode v programskem jeziku Objective-C (slika 3.1). Xcode je brezplačen in ga preprosto namestimo z Applove trgovine z aplikacijami. Je enostaven za uporabo in poleg navadnega ponuja tudi grafično okolje za postavitve gradnikov po aplikacijskem oknu. Xcode ponuja tudi odlična orodja za razhroščevanje aplikacij in upravljanje pomnilnika, ki je nemalokrat razlog za sesutje kakšne aplikacije. Te lahko programer testira na vključenem iOS simulatorju, za kar ne potrebuje dodatnih certifikatov. To pomeni, da lahko aplikacije razvijajo in testirajo tudi programerji, ki niso kupili Applove licence. Ta je sicer potrebna le za objavo aplikacije na trgovini App Store in testiranje na fizičnih napravah.

Objective-C je objektno orientiran programski jezik z značilno sintakso:

```
[objekt metoda:argument];
```

v primerjavi z ekvivalentom v programskem jeziku C++:

```
objekt->metoda(argument);
```

Začetniki imajo s takšno sintakso na začetku težave, vendar se hitro privadijo opisnega pisanja funkcijskih klicev, kar na koncu programski kodi daje boljšo berljivost.



Slika 3.1: Izgled integriranega programskega okolja Xcode

Do nedavnega je moral uporabnik pri razvoju iOS ali Mac OS X aplikacije za dodeljevanje in sproščanje pomnilnika skrbeti sam, kar je včasih povzročalo velike probleme, saj je kakšen nehote sproščen objekt povzročil sesutje celotne aplikacije ali pa se je spomin nekontrolirano dodajal objektom do te mere, da je aplikacija dosegla maksimalno dovoljeno količino dodeljenega spomina in se je prisilno zaprla. Kljub odličnim orodjem v Xcode je bilo včasih odkrivanje takšnih problemov zelo zahtevno opravilo. Ko je Apple z verzijo Xcode 4.2 predstavil funkcionalnost LLVM prevajalnika imenovano ARC (angl. Automatic Reference Counting), je programerjem močno olajšal delo. Po novem je treba objektom še vedno dodeliti prostor, sproščanje pa poteka samodejno, ko na objekt ni vezana nobena referenca več. Aplikacije so tako postale stabilnejše in tudi hitreje, razvoj pa manj kompliciran in bolj zabaven. Po novem je tudi učna krivulja manj strma in učenca hitreje nagradi.

Funkcionalnosti iOS-a so omejene na zbirke knjižnic, ki jih ponuja Apple. Če želimo aplikacijo objaviti na App Store, se moramo pravil teh

knjižnic striktno držati, sicer bo naša aplikacija zavrnjena. S tem ima Apple odlično kontrolo nad zaprtostjo in varnostjo aplikacij, saj komunikacija med njimi na ravni datotečnega sistema ni dovoljena. Seveda iOS podpira tudi integracijo knjižnic tretjih oseb, a se morajo tudi te držati pravil podjetja Apple.

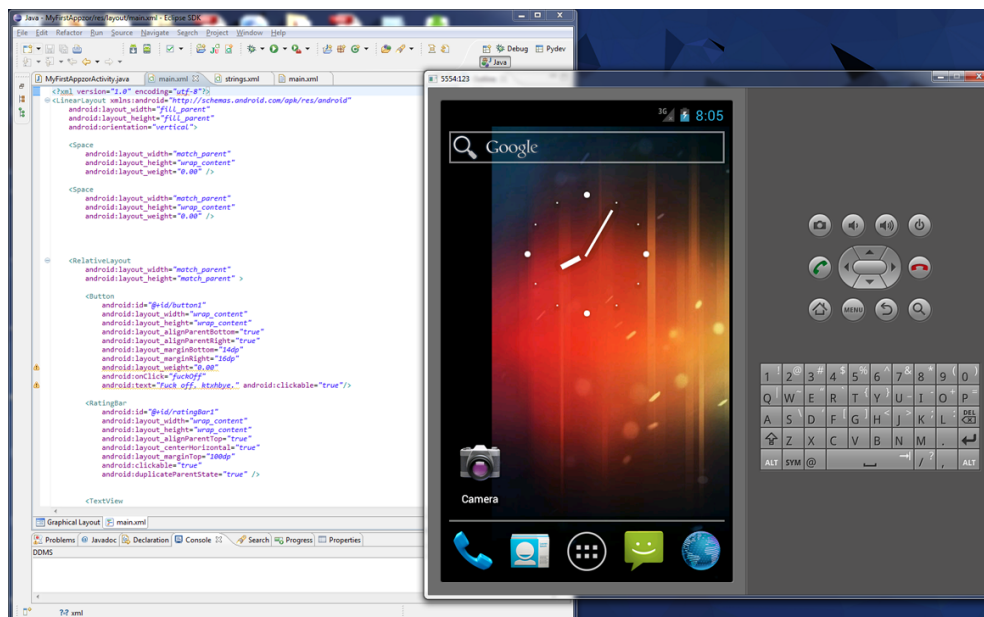
### 3.3 Razvoj aplikacij za Android

Android aplikacije razvijamo v neplačljivem integriranem razvojnem okolju Eclipse v programskem jeziku Java (slika 3.2). Eclipse je v osnovi namenjen razvoju v Javi, vendar ga lahko s pomočjo razširitev uporabimo za razvoj v drugih programskih jezikih kot na primer: C, C++, Python, Ruby, PHP, Scala. Eclipse ponuja tudi širok nabor orodij za razhroščevanje aplikacij, s čimer se uvršča med kvalitetnejša neplačljiva večnamenska integrirana razvojna okolja. Za testiranje aplikacij Android za razliko od iOS-a uporablja emulator namesto simulatorja. Razlika je v tem, da emulator oponaša tudi delovanje strojne opreme, medtem ko simulator simulira le programsko opremo. Posledično je v praksi testiranje aplikacij za operacijski sistem Android počasnejše od tistega za iOS, saj iOS simulator teče bolj gladko, čeprav ne ponuja vedno vseh funkcionalnosti dejanske naprave.

Programski jezik Java je dokaj enostaven za začetnike, saj ponuja odlično dokumentacijo in skrbi za samodejno sproščanje pomnilnika, zato se lahko programer bolj osredotoči na reševanje danega problema. Java je objektno orientiran statično tipiziran programski jezik, s sintakso podobno Cjevi. Primer kratkega programa v Javi:

```
public static void main(String [] args) {  
    System.out.println(" Pozdravljen , svet!");  
}
```

Podobno kot pri iOSu imajo aplikacije omejen dostop do datotečnega sistema in si lahko z drugo aplikacijo izmenjujejo datoteke le, če je programer aplikaciji namenil vlogo ponudnika datotek. Za objavo na Google

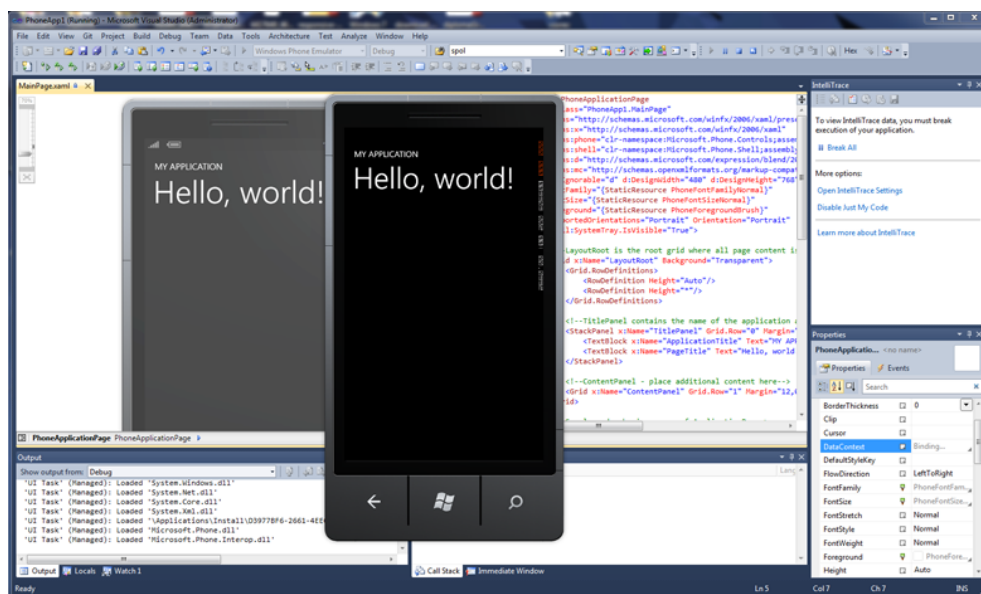


Slika 3.2: Izgled integriranega programskega okolja Eclipse in Android emulatorja

Play trgovini mora distributer aplikacije kupiti licenco, postopek objave aplikacije pa na srečo ni tako strog kot pri Applu.

### 3.4 Razvoj aplikacij za Windows Phone 7

Razvoj aplikacij poteka v neplačljivem integriranem razvojnem okolju Visual Studio 2010 Express for Windows Phone v programskem jeziku C# (slika 3.3). Visual Studio 2010 ponuja odlična orodja za razhroščevanje aplikacij in tako kot Xcode in Eclipse uporabniku med pisanjem programske kode ponuja samodejno dokončanje ukazov in kode. Obstajajo tudi plačljive različice okolja Visual Studio, ki imajo vključena najrazličnejša orodja za razvoj aplikacij za operacijski sistem Windows, razvoj naprednih spletnih aplikacij ipd. Za testiranje aplikacij se tako kot na Android platformi uporablja emulator namesto simulatorja, vendar je ta v primerjavi z Androidovim zelo hiter.

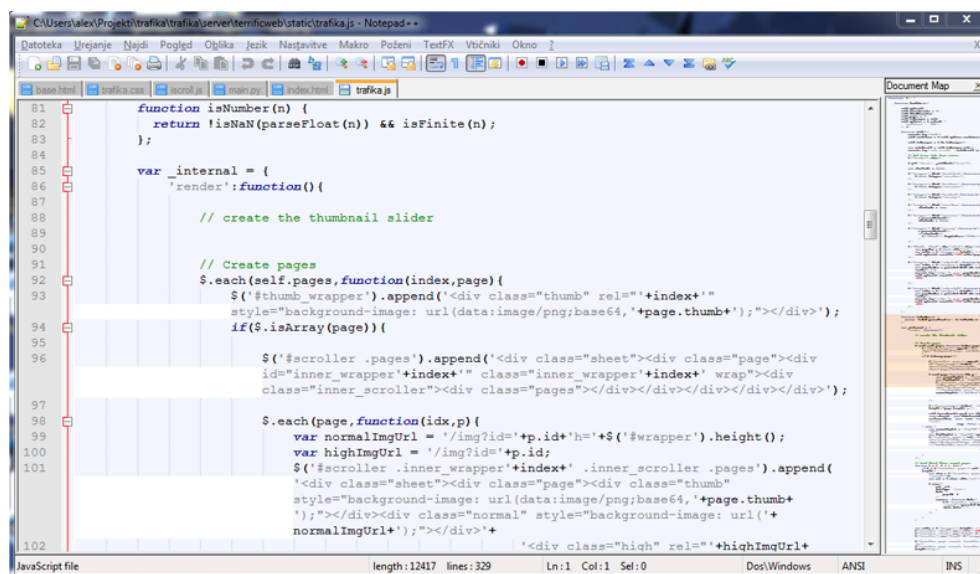


Slika 3.3: Izgled integriranega programskega okolja Visual Studio 2010 Express for Windows Phone in Windows Phone 7 emulatorja

C# je moderen in enostaven programski jezik primeren tudi za začetnike. Razvijalcem je na voljo odlična dokumentacija in zahvaljujoč veliko razvijalcem tudi veliko razprav na temo razvoja. Za sproščanje pomnilnika C# skrbi sam in s tem zmanjšuje možnost vsiljenega zaprtja aplikacije. Je objektno orientiran statično tipiziran programski jezik, s sintakso podobno Javini. Primer kratkega programa v C#:

```
static void Main()
{
    Console.WriteLine("HelloWorld!");
}
```

Podobno kot pri iOS in Androidu imajo aplikacije omejen dostop do datotečnega sistema in drugih aplikacij. Za objavo v trgovini Windows Phone Marketplace mora distributer aplikacije kupiti licenco, postopek objave pa je podoben kot pri Androidu. Distributer izpolni podatke o aplikaciji, nastavi ceno in aplikacija je kmalu objavljena v trgovini.

The image shows a Notepad++ window with a JavaScript file named 'trafika.js'. The code defines a function 'isNumber' and a 'render' function. The 'render' function uses jQuery to create a thumbnail slider and a page scroller. It iterates over 'pages' to create thumbnails and full-page views. The status bar at the bottom indicates the file is a JavaScript file, has a length of 12417, and contains 329 lines of code.

```
81 function isNumber(n) {
82     return !isNaN(parseFloat(n)) && isFinite(n);
83 };
84
85 var _internal = {
86     'render':function() {
87         // create the thumbnail slider
88
89
90
91         // Create pages
92         $.each(self.pages, function(index, page) {
93             $('#thumb_wrapper').append('<div class="thumb" rel="'+index+'"'
94             style="background-image: url(data:image/png;base64,'+page.thumb+');"></div>');
95             if($.isArray(page)) {
96
97                 $('#scroller .pages').append('<div class="sheet"><div class="page"><div
98                 id="inner_wrapper'+index+'" class="inner_wrapper'+index+' wrap"><div
99                 class="inner_scroller"><div class="pages"></div></div></div></div>');
100
101                 $.each(page, function(idx, p) {
102                     var normalImgUrl = '/img?id='+p.id+'h='+$('#wrapper').height();
103                     var highImgUrl = '/img?id='+p.id;
104                     $('#scroller .inner_wrapper'+index+' .inner_scroller .pages').append(
105                     '<div class="sheet"><div class="page"><div class="thumb"
106                     style="background-image: url(data:image/png;base64,'+page.thumb+
107                     ');"></div><div class="normal" style="background-image: url('+
108                     normalImgUrl+');"></div>'+
109                     '<div class="high" rel="'+highImgUrl+'
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
```

Za razvoj zaledja spletne aplikacije je potrebna namestitev željenega programskega jezika, strežnika in podatkovne baze. Na voljo so tudi orodja, ki razvijalcem spletnih aplikacij pomagajo pri razvoju, kot je na primer Adobe Dreamweaver. Ta razvijalcu ponuja samodejno generiranje HTML kode, vendar se profesionalni razvijalci takšnim orodjem raje izognejo, saj rezultat velikokrat vsebuje odvečno kodo, ki jo je potem težje vzdrževati. Nasprotno pa je za začetnike ta način vabljev, ker nudi enostavne odgovore na vprašanja o načinu implementacije določenega problema.

S stališča razvijalca je razvoj spletne aplikacije v primerjavi z domorodno s tehničnega vidika lažji. Aplikacije ni potrebno zapakirati in distribuirati preko namenskih aplikacijskih trgovin, ampak jo razvijalec enostavno postavi na spletni strežnik in jo oblikuje na način, ki omogoča pregleden prikaz na mobilni napravi. Ostale prednosti in slabosti razvoja spletnih aplikacij pred domorodnimi so naštetje v spodnjih dveh seznamih.

**Prednosti:**

- enostavna distribucija brez posebnih omejitev,
- krajši čas razvoja,
- enostavnejši razvoj,
- minimalne prilagoditve za ostale operacijske sisteme,
- enostavno vzdrževanje, popravki so vidni takoj in na vseh napravah,
- na voljo je veliko knjižnic, ki posnemajo izgled in funkcionalnosti nekaterih operacijskih sistemov (Sencha Touch, jQuery mobile, itd.),
- zaslužek ima prednost pred domorodnimi aplikacijami, saj se izognemo plačilu določenega odstotka našega zaslužka distributerju.

**Slabosti:**

- potrebna internetna povezava (ne nujno v vseh primerih),
- omejen prostor za lokalno shranjevanje datotek,
- ni dostopa do knjižnic, ki jih ponuja operacijski sistem, na primer zajem slike s kamero ali pošiljanje elektronskega sporočila,
- testiranje na veliko različnih napravah (podobno kot Android), v primerjavi z iOS platformo,
- v JavaScriptu je težko dobro oponašati prepoznavanje vzorcev, na primer povečevanja ali potegov,
- rešitve procesorsko zahtevnejših problemov kljub vsemu v domorodnih aplikacijah tečejo bolj tekoče kot v spletnih alternativah.
- zaslužek je negativna lastnost, ker imamo na voljo manj poslovnih modelov za zaslužek s spletno aplikacijo, kot jih imamo pri domorodni aplikaciji. Pri spletni aplikaciji se moramo večinoma zanašati na prihodek od oglasov, razen če imamo specifično aplikacijo, ki zahteva registracijo in spletni nakup. Domorodne aplikacije nam omogočajo nakup same aplikacije, potem pa še kupovanje dobrin znotraj aplikacije, odvisno od poslovnega modela.



# Poglavje 4

## Implementirana rešitev

### 4.1 Uporabljene tehnologije in pristopi

#### 4.1.1 HTML

Jezik za označevanje nadbesedila HTML (angl. Hyper Text Markup Language) je skriptni jezik za izdelavo spletnih strani. Danes ga srečujemo vedno bolj pogosto in je definitivno prestopil okvire znotraj katerih je bil načrtan.

HTML je zapisan v obliki elementov, ki jim v slovenščini pravimo značke. Te so sestavljene iz imena elementa in lomljenih oklepajev, na primer:

```
<html> </html>.
```

Večinoma so značke zapisane v parih kot v primeru zgoraj, obstajajo pa tudi elementi brez končne značke. Znotraj značk lahko gnezdimo druge značke oziroma pišemo poljubno vsebino. Dokumenti HTML so v bistvu navodila namenjena spletnemu brskalniku in označujejo vsebino spletne strani, zato ob prikazu spletne strani ne vidimo HTML značk, ampak urejeno vsebino. Jezik ima dokaj enostavno učno krivuljo, saj je število pomembnih značk zelo majhno, za pisanje kode pa ne potrebujemo velikih

strojnih zmogljivosti, saj lahko HTML kodo pišemo v kateremkoli urejevalniku besedil.

Zgodovina jezika HTML sega v leta 1980-1990, ko je Tim Berners-Lee, sicer fizik zaposlen v Evropski organizaciji za jedrske raziskave, definiral HTML in konec leta 1990 implementiral enostaven strežnik in spletni brskalnik. Na začetku je bil jezik sestavljen iz 18 značk, od katerih se jih je do danes ohranilo 11.

Od leta 2000 naprej je uradna verzija HTML-ja 4.01, v razvoju pa je verzija 5.0, o kateri bomo izvedeli več v nadaljevanju.

HTML dokumente lahko prenašamo tako kot katerekoli druge računalniške datoteke, vendar je daleč najpogostejše oddaljeno dostopanje preko spletnega brskalnika, ki se na strežnik poveže s HTTP protokolom in nato prenese vse informacije potrebne za prikaz spletne strani na uporabnikovem računalniku ali mobilni napravi. Eden izmed razširjenih načinov prejemanja HTML dokumentov je tudi elektronska pošta, ki s tem omogoča oblikovanje elektronskega sporočila, na primer dodajanje barv ali slik v besedilo. Takšna sporočila uporabniku nudijo bolj osebno izkušnjo in enostavnejši dostop do pomembnih informacij.

### 4.1.2 HTML5

Kot omenjeno je HTML5 naslednik verzije 4.01 in prinaša kar nekaj pomembnih novosti, sicer pa je trenutno še vedno v razvoju. Cilj te verzije je boljša podpora za večpredstavnostne vsebine, hkrati pa zagotavljanje kompatibilnosti s starejšo verzijo. Ker je HTML5 zgrajen tudi z ozirom na počasnejše naprave, je odličen kandidat za spletne aplikacije za pametne telefone in tablične računalnike [25] [27].

HTML5 je do sedaj predstavil tudi nekaj sintaktičnih izboljšav kot na primer značke:

```
<video> <audio> in <canvas>.
```

S temi značkami lahko v našo spletno stran brez težav vključimo večpredstavn-

ostne vsebine in grafične elemente, brez uporabe dodatnih vključkov za brskalnik. Med pomembnejše nove elemente spadajo tudi:

`<section>`, `<article>`, `<header>`, `<footer>` in `<nav>`,

ki so mišljeni za boljšo semantično razdelitev dokumenta in nadomeščajo uporabo generičnih značk kot sta na primer:

`<div>` in `<span>`.

Poleg nekaterih novih značk ter nekaterih opuščeni pa HTML5 vključuje tudi nekaj novih programskih vmesnikov. Mednje spadajo:

- platno,
- spletne aplikacije v brezpovezavnem načinu,
- povleci in spusti,
- pošiljanje enostavnih sporočil med različnimi dokumenti,
- upravljanje z zgodovino brskalnika,
- shranjevanje podatkov v brskalniku,
- vmesnik za geolokacijo.

HTML5 sam ne more animirati elementov, zato je za animacijo potrebna uporaba CSS3 ali JavaScripta.

Čeprav je HTML5 še vedno v razvoju, se število spletnih stvari grajenih s to tehnologijo hitro večja, saj so te spletne aplikacije inovativne, odzivne ter uporabniku privlačne.

Velik napredek se dogaja tudi pri uporabi jezika HTML5 za gradnjo mobilnih spletnih aplikacij, kjer velikokrat niti ne razlikujemo med domorodno in mobilno spletno aplikacijo, saj lahko slednje odlično imitirajo domorodno aplikacijo. Prednost tega je seveda razvoj za več platform hkrati [29] in s tem prihranek pri času ter denarju, je pa treba paziti na

strojne zmogljivosti ciljne naprave. Odločitev za razvoj mobilne spletne aplikacije namesto domorodnih je seveda odvisna od vrste aplikacije, saj je dobro, če ta ni preveč kompleksna in strojno zahtevna.

Čeprav je bil HTML5 že od samega začetka dobro poznan med programerji, pa je širšo medijsko pozornost dobil leta 2010, ko je bivši direktor podjetja Apple, Steve Jobs, v javnem pismu izrazil svoje mnenje, v katerem je zaključil, da Adobe Flash ni več potreben za predvajanje večpredstavnostnih vsebin oziroma predstavitev kakršnihkoli informacij s spleta ter da prihaja nova doba, v kateri bo zmagal HTML5. Novembra 2011 je podjetje Adobe oznanilo, da ukinja razvoj tehnologije Flash za mobilne naprave.

HTML5 podpirajo najrazličnejši spletni brskalniki, vključno z mobilnimi. Dobro podporo imajo: mobile Safari na iOS, Android brskalniki na Androidu, Internet Explorer 9 mobile na Windows Phone 7, Blackberry brskalniki ali Opera Mobile na Blackberry napravah, Bolt na Symbianu, Dolfin na Bada OS, brskalniki Web na Web OS ipd. [22]

### 4.1.3 CSS in CSS3

Če HTML skrbi za strukturo dokumenta oziroma spletne strani, stilska predloga CSS (angl. Cascading Style Sheets) skrbi za njeno obliko. Osnovni namen predlog je definicija enostavnih pravil za prikaz elementov na strani, ki jih zapišemo v obliki enostavnega skriptnega jezika.

S hitro širitvijo svetovnega spleta se je pojavila tudi potreba po stiliziranju dokumentov, saj standardni HTML ni več zadovoljeval vse zahtevnejše množice uporabnikov. Leta 1994 sta Håkon Wium in Bert Bos s skupnimi močmi definirala in predstavila prvo verzijo kaskadnih predlog, leta 1996 pa je bila ta uradno razglašena kot priporočilo. Že pred razglasitvijo sta jo implementirala spletna brskalniki Netscape in Microsoft Internet Explorer in s tem pripomogla k širitvi specifikacije.

Vse verzije CSS-a uporabljajo enako sintakso, ki je sestavljena iz opisa elementa in seznama parov lastnosti in njihovih vrednosti, kot je prikazano

na spodnjem primeru:

```
#id_elementa { width: 200px; height: 100px;
                background-color: #bada55; }.
```

Zelo uporabna lastnost stilskih predlog je ponovna uporaba definiranih pravil. Zgornji primer kaže določitev sloga točno določenemu elementu, saj se na element sklicujemo le po njegovem unikatnem identifikatorju. To je včasih seveda priročno, saj se nam nemalokrat zgodi, da želimo točno določenemu elementu iz skupine elementov nastaviti določeno lastnost, večinoma pa uporabljamo druge načine naslavljanja elementov. Tem načinom z drugo besedo rečemo tudi izbirniki (angl. selectors). Pri pisanju stilskih predlog moramo paziti na vrstni red pravil, saj pravila, ki so zapisana nižje v zunanji CSS datoteki prevladajo nad tistimi, ki so napisana višje. Izjema so močnejši izbirniki, saj ti povežijo tudi vrednosti zapisane pod njimi. Eden takih je izbirnik # (izbirnik po unikatnem identifikatorju). Sicer si CSS definicije po prioriteti sledijo glede na lokacijo v HTML dokumentu. Vrstični (angl. inline) stili, ki so definirano direktno na elementu, imajo največjo prednost, sledijo jim vgrajeni stili (največkrat v glavi HTML dokumenta med značkami <style>), potem pa zunanji stili, ki so definirani v datoteki tipa .css in vključeni v dokument HTML. Izjema je možnost direktnega nastavljanja najvišje prioritete, kar storimo tako, da vrednosti pripišemo zaključek "!important", na primer:

```
p { text-align: center !important; }
p { text-align: left; },
```

kjer bo tekst znotraj odstavka poravnano sredinsko zaradi prevlade zgornjega pravila.

Ostali pogosto uporabljeni izbirniki:

- izbirnik po vrsti značke,
- izbirnik po razredu elementa,
- iskanje po otrocih na kateremkoli nivoju,

- iskanje po otrocih na prvem nivoju,
- iskanje po bratih

Kot primer si lahko predstavljamo, da želimo vsem elementov tipa `<div>` in `<article>`, ki so otroci elementa z razredom "rumen", nastaviti rdečo barvo pisave, vsem njihovim otrokom tipa `<span>` na prvem nivoju pa modro barvo pisave. Ostali otroci imajo pisavo rdeče barve.

```
.rumen div , .rumen article { color: red; }  
.rumen div > span , .rumen article > span { color: blue; }
```

Velik problem pri oblikovanju elementov pa žal še vedno povzročajo različni spletni brskalniki, saj se CSS in CSS3 pravila v vseh ne obnašajo enako. Pri razvoju spletne strani se zato del časa razvoja nameni zagotavljanju kompatibilnosti med vsemi brskalniki, pri čimer prevladuje optimizacija spletne strani za Internet Explorer 7 in 8 [23].

CSS3 [27] je nadgradnja različice CSS 2.1 in je razdeljena v več modulov, kar omogoča lažjo nadgradnjo. Trenutno je večina modulov še v razvoju, uradno sprejeti pa so naslednji trije: barve, imenski prostori (angl. namespaces) in izbirniki.

Poleg omenjenih modulov CSS3 prinaša veliko izboljšav in novosti, kot so na primer animacije, transformacije (2D in 3D), večja kontrola nad robovi (zaobljeni robovi) in druge (skupaj več kot 40 modulov). Vse te izboljšave in novosti predstavljajo zelo močna orodja za izdelavo lepih in interaktivnih spletnih aplikacij, predvsem pa minimizirajo uporaba JavaScripta, kar spletnim aplikacijam daje tudi večjo stabilnost in odzivnost.

Trenutno CSS3 podpirajo vsi moderni spletni brskalniki (Chrome, Firefox, Opera, Safari in tudi Internet Explorer 9), zato število spletnih aplikacij s to tehnologijo hitro raste, kljub slabi združljivosti z Internet Explorerjem verzije 7 in 8. V tem primeru se spletni razvijalci poslužujejo skript kot je Modernizr [17] [27], ki zazna funkcionalnosti podprte s strani brskalnika in vam omogoča enostavno implementacijo alternativ (sicer je detektiranje podpore določenim HTML5 elementom in CSS3 pravilom do-

kaj zahtevno opravilo). Primer detekcije HTML5 video podpore s skripto Modernizr:

```
if (Modernizr.video) {  
    // HTML5 video  
} else {  
    // ni podpore za HTML5 video, uporabimo Flash  
}
```

#### 4.1.4 JavaScript in jQuery

JavaScript je objektno orientiran, dinamično in šibko tipiziran jezik. Navkljub imenu nima prav dosti skupnega s programskim jezikom Java razen C-ju podobne sintakse in nekaj Javi sorodnih knjižnic. JavaScript je v desetih dneh za podjetje Netscape razvil Brendan Eich. V tekmovanju z Microsoftom je Netscape želel uporabnikom ponuditi lahko in prenosno različico Jave, ki bi ojačala uporabniško izkušnjo s spletno stranjo. Zaradi vsesplošnega uspeha ga je nato v svojem spletnem brskalniku Internet Explorer leta 1996 podprl tudi Microsoft.

JavaScript se primarno uporablja za pisanje metod, ki so vključene v HTML in manipulirajo z DOM (angl. Document Object Model) drevesom strani. Nekaj primerov uporabe JavaScripta:

- dinamično nalaganje delov spletne strani ali pošiljanje podatkov na strežnik brez osveževanja celotne strani,
- animacija elementov,
- uporaba v igrah, omogoča navigacijo po spletni strani s pomočjo bližnjic na tipkovnici,
- validacija spletnih obrazcev,
- matematične operacije, operacije z datumi.

Ker za izvajanje JavaScripta skrbi spletni brskalnik nameščen na uporabnikovem računalniku, je JavaScript zelo hiter in odziven, ta hitrost pa je v veliki meri odvisna od optimizacije brskalnikovega JavaScript pogona.

JavaScript znotraj spletnega brskalnika teče v t.i. peskovniku, kar pomeni, da aplikacije nimajo dostopa do uporabnikovega datotečnega sistema in tako ne morejo ustvarjati datotek ali nameščati zlonamernih programov. Druga vrsta zaščite uporabnika pa je zmožnost izvajanja samo tistih skript, ki prihajajo iz istega izvora. Pod isti izvor spadajo vsi dokumenti, ki se nahajajo znotraj iste domene, protokola in vrat. Primer istega izvora:

```
http://www.fri.uni-lj.si/index.html
http://www.fri.uni-lj.si/index2.html.
```

Primer, kjer je izvor drugačen (druga domena in protokol):

```
https://fri.uni-lj.si/index.html
http://www.fri.uni-lj.si/index2.html.
```

Na ta način se lahko prepreči nedovoljeno zbiranje in posredovanje informacij, kot so na primer shranjena gesla in aktivne seje.

Zaradi splošne priljubljenosti in enostavnosti JavaScripta, se je ta razširil tudi onkraj meja spletnih brskalnikov. Dandanes se tako uporablja tudi za razvoj razširitev za spletne brskalnike kot skriptni jezik v Photoshopu, za pisanje iger v pogonu Unity ipd.

Pri implementaciji naše spletne aplikacije smo veliko uporabljali knjižnico jQuery [24], ki je namenjena poenostavitvi nekaterih JavaScript funkcij za manipulacijo DOM drevesa, ponuja pa tudi odličen sistem vtičnikov.

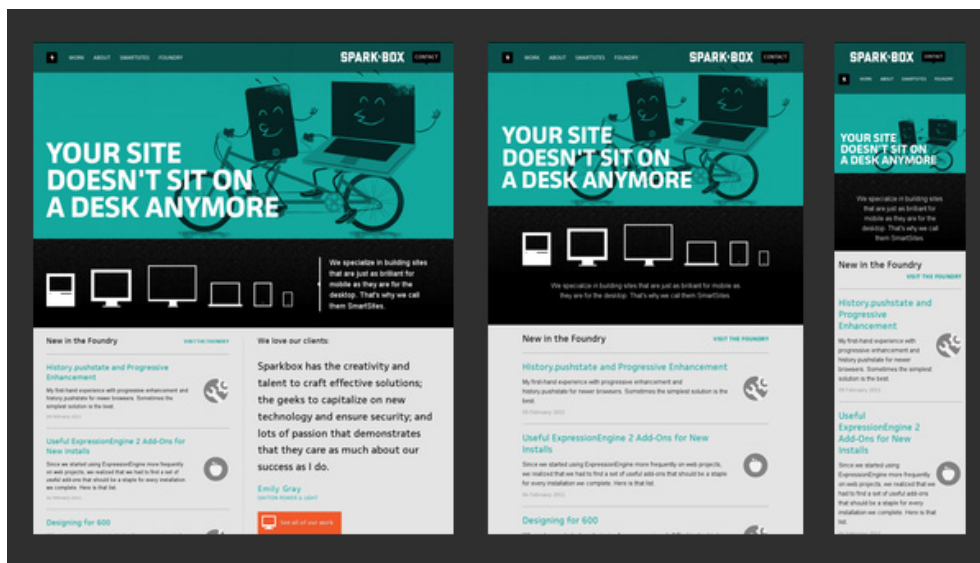
### 4.1.5 Odzivno spletno oblikovanje

S prihodom prvih mobilnih naprav z dostopom do spleta so se pojavile tudi prve spletne strani optimizirane za te naprave. Mobilne različice spletnih strani se ponavadi nahajajo na drugem spletnem naslovu kot izvorna stran, na primer na poddomeni tipa **m.example.com**. Če spletna stran

ni eksplicitno preverjala s kakšne naprave prihaja uporabnik, se je znalo zgoditi to, da je uporabnik z mobilne naprave prijateljem preko socialnega omrežja delil povezavo na mobilno verzijo spletne strani. Ko so ti prijatelji povezavo odprli, se jim je pred očmi prikazala ozka in okleščena verzija spletne strani in jih je do standardne verzije pripeljal šele klik na povezavo na dnu strani. Enako lahko velja tudi v obratni smeri, ko uporabnik na mobilni napravi odpre spletno stran in je ta prikazana v pomanjšani obliki, da zapolni celoten ekran. Uporabnik mora nato želen del vsebine povečati in stran premikati tako vertikalno kot tudi horizontalno. Zgoraj navedena primera poslabšata uporabniško izkušnjo, kar pomeni, da se bralec ob branju člankov na naši spletni strani najverjetneje ne bo zadržal prav dolgo [28].

Tukaj nastopi odzivno spletno oblikovanje. Tehnologije za razvoj spletne strani z odzivnim spletnim oblikovanjem obstajajo že nekaj časa, vendar se je pojem začel uveljavljati šele v zadnjem času, saj lahko ta efekt s pomočjo CSS3 dosežemo enostavneje. Ideja je, da za namen prikaza strani na mobilni napravi ne naredimo nove spletne strani oziroma nove poddomene, ampak obstoječo stran z manjšimi popravki prilagodimo tako, da se bo znala lepo prikazati na napravi (slika 4.1). Ta koncept je vedno bolj pomemben, saj dostop do spleta z mobilnih naprav strmo narašča, zato morajo biti spletne strani prilagodljive.

CSS3 omogoča definicijo enostavnih skupin pravil s katerimi prikaz strani omejimo na določeno širino, elemente strani pa na pravo pozicijo umestimo s tekočo postavitvijo.



Slika 4.1: Prikaz spletne strani z odzivnim spletnim oblikovanjem na različnih širinah spletnega brskalnika

Primer definicije skupine pravil "media":

```
@media screen and (max-width: 480px) {
    #content {
        width: 60%;
        padding: 3% 4%;
    }
}
```

Zgornji primer prikazuje CSS pravilo, ki se upošteva le v primeru, ko je širina strani manjša ali enaka 480 slikovnih elementov. Enako naredimo za ločljivosti ostalih naprav, ki jih želimo podpreti in nato našo spletno stran testiramo tako, da zmanjšamo okno brskalnika.

Poleg dinamičnega zmanjševanja velikosti elementov in pisave pa je zelo pomembno dinamično zmanjševanje slik, saj so te največkrat vzrok za pokvarjeno postavitev strani. Ta problem rešimo z enostavnim pravilom:

```
img { max-width: 100%; }
```

## 4.2 Implementacija spletnega portala

Glede na to, da je zaledje domorodnih aplikacij sistema Trafika zgrajeno na Googlovi platformi App Engine, smo se odločili, da bomo tudi spletno aplikacijo Trafika zgradili na tej platformi.

### 4.2.1 Google App Engine

Google App Engine [26] je v osnovi brezplačna platforma za gostovanje spletnih aplikacij v oblaku. Naša spletna aplikacija je razdeljena med večimi spletnimi strežniki, ki jih upravlja Google. V primeru, da se obisk naše spletne aplikacije zelo poveča, App Engine sam poskrbi za skaliranje – do količine, ki smo jo zakupili. Za manjše aplikacije je storitev brezplačna, cena večjih aplikacij pa je odvisna od števila porabljenih procesorskih ur in števila dostopov.

App Engine podpira programske jezike Python, Java in Go. Odločili smo se, da bomo strežniški del spletnega portala implementirali v Pythonu, skupaj s programskim ogrodjem webapp, ki ga je razvil Google, sicer pa je podoben znanemu ogrodju Django.

Namestitev na našem sistemu je bila enostavna, saj smo imeli Python že nameščen, zato nam je preostala le še namestitev programa Google App Engine Launcher, ki nam je omogočal lokalno testiranje projekta.

Ko smo namestili vsa potrebna orodja za delo, smo po principu MVC (angl. Model–View–Controller) definirali vstopne točke v našo spletno aplikacijo in pričeli z implementacijo zaledja aplikacije. Definirali smo potrebne poizvedbe iz baze in izbrali HTML predlogo, ki se napolni s podatki glede na izbrani pogled. Ta korak nam omogoča odličen sistem predlog programskega ogrodja webapp.

Začetni pogled, ki je enak za vse uporabnike, je seznam revij, ki so na voljo za branje (slike 4.3, 4.4, 4.5). Pri tem pogledu se s strežnika v formatu JSON poleg predloge prenesejo podatki o vseh razpoložljivih revijah skupaj z opisi, naslovi in podatki o aktualni številki posamezne

revije. Naslednji primer prikazuje, kako v HTML kodi uporabimo sistem predlog z ogrodjem webapp za dinamično generiranje HTML elementov, ki služijo prikazu zadnje številke revije:

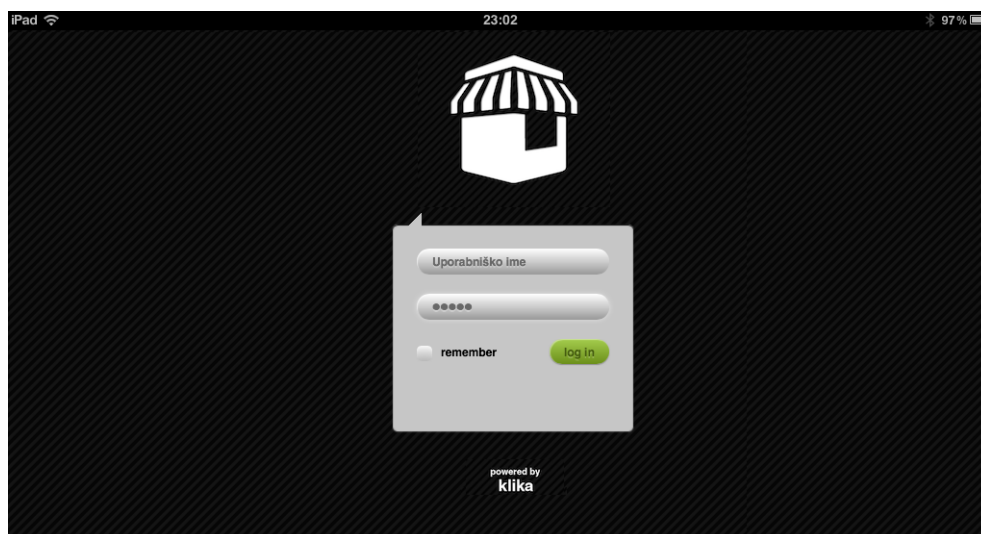
```
{% for publication in data.publications %}
<div class='publication' rel='{{ publication.name }}'>
    <span class='title'>{{ publication.title }}</span>
    <img rel='{{ publication.lastIssue.thumbnail.urls.0 }}'>
</div>
{% endfor %}
```

Ko smo zgradili osnovo za prikaz revij, smo definirali CSS pravila za postavitev teh na strani. Po načelih odzivnega spletnega oblikovanja smo definirali še CSS pravila za prikaz na mobilnih napravah, kjer se na zaslonu naenkrat prikaže manj revij. V JavaScriptu smo implementirali dogodke, ki se zgodijo, ko se naloži stran. S pomočjo knjižnice jQuery smo vsaki publikaciji dodelili akcijo, ki se zgodi ob kliku oziroma dotiku in sicer je to prikaz okna, ki vsebuje zadnje številke te publikacije. Obenem smo opisali tudi dogodke, ki se zgodijo, ko uporabnik zapre okno s številkami publikacije. Implementacija začetnega klica v JavaScriptu:

```
TrafikaWeb.init = function ()
{
    TrafikaWeb.requestPublications ();

    $('div.publication').die('click').live('click', function
    () {
        TrafikaWeb.requestIssues ($ ( this ). attr ( ' rel ' ));
    });

    $('div.btnClose').click ( function () {
        $("div.publications").fadeIn ();
        $("div.toolbar").fadeOut ();
        $("div.issues").fadeOut ();
    });
};
```



Slika 4.2: Prijava v sistem

Klice na strežnik, ki se zgodijo ob klicu posamezne publikacije, smo implementirali s tehnologijo AJAX (angl. Asynchronous JavaScript and XML), ki nam je omogočila dinamičen prikaz vsebin brez osveževanja cele spletne strani. Ta princip je pri mobilnih spletnih aplikacijah zelo pomemben, saj če naj bi spletna aplikacija posnemala domorodno mobilno aplikacijo, moramo paziti na to, da je uporabniški vmesnik čimbolj tekoč, odziven.

Preden lahko uporabnik začne z branjem številke publikacije, se mora prijaviti (slika 4.2) s svojim uporabniškim imenom in geslom, ki sta na vseh napravah enaka. Ta korak je pomemben zato, da lahko v zaledju preverimo, če je uporabnik izbrano številko slučajno že kupil oziroma je naročnik publikacije ali pa jo mora še kupiti. Z ozirom na kompleksnost implementacije plačilnega sistema, smo tega za potrebe diplomske naloge izpustili in predvideli, da so vse številke vseh revij zastoj. Za implementacijo plačilnega sistema bi morali v platformo Trafika implementirati podporo HTML modulu, ki ga je razvilo podjetje Klika d.o.o. za spletno plačevanje pri drugih projektih. Ta v ozadju s spletnimi servisi združuje več plačilnih sistemov in podpira na primer plačevanje s kreditnimi karti-

cam, sistem NLB Klik, ipd. HTML modul uporabnika ob plačilu ponavadi preusmeri (odvisno od storitve) na spletno stran ponudnika, kjer se opravi transakcija. Ponudnik nato preko spletnega servisa HTML modul obvesti o uspešnosti transakcije. Ker plačilnega sistema zaenkrat nimamo, lahko uporabnik s klikom oziroma dotikom gumba začne brati številko (slika 4.6).

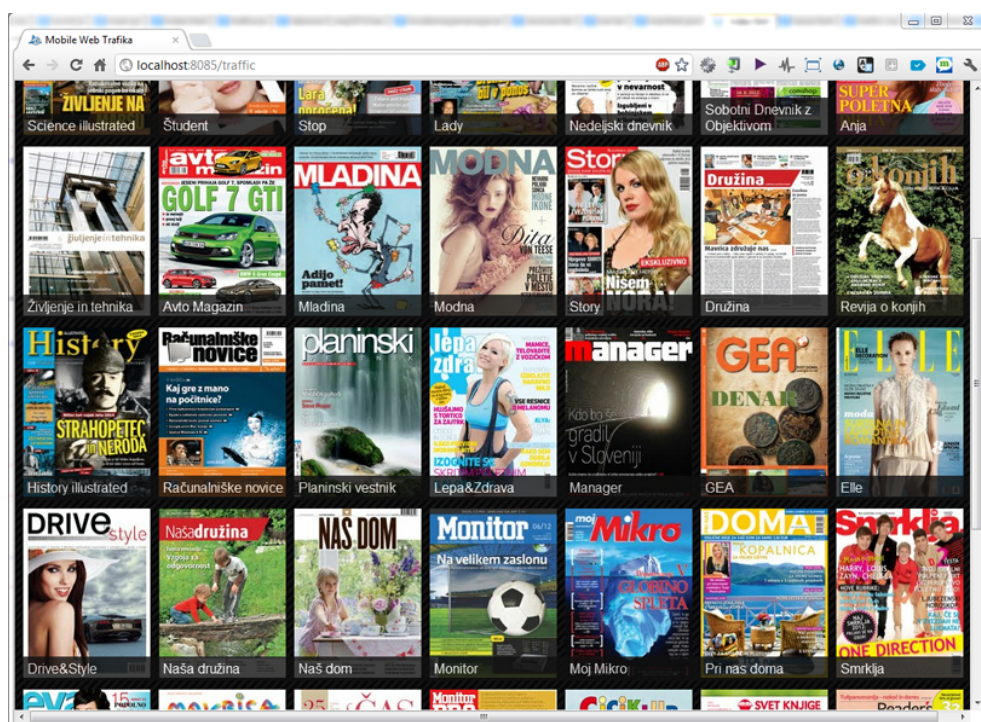
Na koncu smo dodali še ikono spletne aplikacije, ki se prikaže, če uporabnik aplikacijo preko brskalnika kot zaznamek shrani na namizje svoje naprave (slika 4.7).



Slika 4.3: Dostop do spletnega portala s tabličnim računalnikom iPad



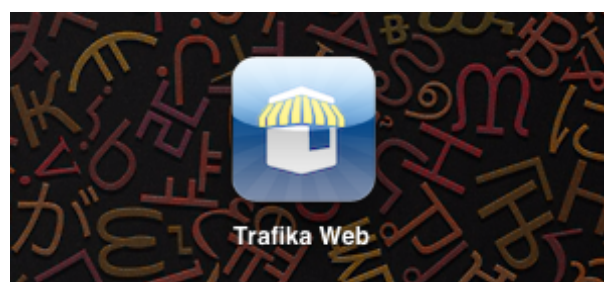
Slika 4.4: Dostop do spletnega portala z Android mobilnim telefonom



Slika 4.5: Dostop do spletnega portala z namiznim računalnikom



Slika 4.6: Izbiranje številke za branje



Slika 4.7: Ikona spletne aplikacije na namizju iPada

### 4.3 Implementacija spletnega bralnika revij

Spletni bralnik je najkompleksnejši del naše spletne aplikacije in nam je kot tak za implementacijo vzel tudi največ časa. Najprej smo se morali odločiti za format revij, saj so bile do sedaj vse revije v PDF formatu. Ker nismo našli ustreznega in dovolj hitrega načina za prikaz in navigacijo po PDF dokumentih znotraj spletnega brskalnika, smo se odločili za pretvarjanje strani revij v slike formata JPG. Vsako stran številke revije ob uvozu v administracijskem vmesniku Trafike iz PDF formata pretvorimo v več slik različnih dimenzij. Na ta način optimiziramo velikost datotek in s tem zmanjšamo stroške prenosa slik na mobilno napravo. Večjim napravam, kot na primer iPad in namiznim računalnikom, serviramo slike v standardni velikosti  $488 \times 656$  slikovnih elementov in visokokakovostne v velikosti  $1058 \times 1422$  slikovnih elementov. Za manjše naprave je standardna velikost slike  $320 \times 430$  slikovnih elementov, visokokakovostna velikost pa enaka kot pri ostalih napravah –  $1058 \times 1422$  slikovnih elementov. Če bi želeli ločljivost visokokakovostnih slik povečati (na primer na dvakratno velikost –  $2116 \times 2844$  slikovnih elementov), bi zaradi omejitev glavnega pomnilnika morali implementirati metodo za prikazovanje slik visokih ločljivosti [30] pri kateri bi bil v glavnem pomnilniku le del slike, ki je trenutno prikazan, ostali deli pa bi se s strežnika naložili po potrebi.

Bralnik smo zastavili kot samostojni jQuery vtičnik z namenom enostavne integracije na drugih spletnih straneh ali produktih. Ob inicializaciji se na strežnik pošlje zahteva po informacijah o trenutni številki. Klic na strežnik:

```
$.get('/issue', _callbacks['issue']);
```

Obenem smo se registrirali na številne dogodke, ki so večinoma povezani z detekcijo uporabnikovih potez na zaslonu mobilne naprave. Zaznavamo dotike, vlečenje prsta po ekranu in geste za navigacijo na naslednjo ali prejšnjo stran. Definirali smo tudi akcijo, ki se zgodi, ko uporabnik izbere številko strani iz priročne navigacije s predogledom.

Ko smo s strežnika dobili informacije o trenutni številki, smo sprožili generiranje HTML elementov. Vsaki generirani strani revije smo najprej kot sliko ozadja nastavili njeno sliko predogleda, ki smo jo kot v Base64 besedilnem formatu prejeli kar skupaj z JSON odgovorom. Slika predogleda je od navadne slike desetkrat manjša, zato na mestu slike za nekaj trenutkov zagledamo zelo zamegljeno sliko, ki se v nadaljevanju navidezno izostrí. Skupaj s predogledom in prostorom z naslovom slike prave velikosti znotraj ene strani številke revije definiramo še prostor, ki vsebuje pot do visokokakovostne slike. Ko smo z generiranjem končali, smo naložili prve tri slike v standardni velikosti in jih prikazali nad njihovimi raztegnjenimi predogledi. Na ta način smo dosegli efekt postopnega nalaganja slike. Skupaj z generiranjem strani smo ustvarili tudi slikovni predogled vseh strani, do katerega uporabnik dostopa s kratkim dotikom kjerkoli na ekranu (slika 4.8).



Slika 4.8: Prikaz predogleda oziroma hitre navigacije po vseh straneh

Na tem mestu smo imeli pripravljeno ogrodje našega bralnika. Ker smo želeli funkcionalnost bralnika čim bolj približati domorodnim aplikacijam, smo implementirali prehode med stranmi in sistem, ki na levi in desni strani trenutne strani naloži dve strani standardne kakovosti. Tiste slike strani, ki padejo iz tega okvirja, se zbríšejo in s tem zmanjšajo odtis aplikacije v



Slika 4.9: Prikaz branja revije v spletnem bralniku na tabličnem računalniku iPad. Prikazana stran je povečana

pomnilniku. Zaradi optimizacije (poglavje 4.4) se ob vrnitvi na izbrisano stran, ta s strežnika ne naloži ponovno, ampak se iz lokalnega pomnilnika naloži v glavni pomnilnik.

Manjkala nam je še implementacija povečave strani. Ker je detekcija geste povečave v JavaScriptu zelo zahtevna zaradi omejenega dostopa do podatkov o dotikih na ekranu, smo se po pomoč najprej obrnili na knjižnico jQuery Mobile, vendar tam te funkcionalnosti nismo našli, zato smo na koncu za potrebe povečave uporabili odprtokodno knjižnico iScroll [9]. V tej knjižnici smo dobili dostop do dogodka, ki se proži vsakič, ko uporabnik poveča stran. Na podlagi tega dogodka smo implementirali funkcijo, ki glede na trenutno povečavo inteligentno menja med prikazom standardne ločljivosti slike strani in prikazom visokokakovostne slike strani. Slika visoke kakovosti se pokaže vsakič, ko je povečava večja od 1 in zbriše, ko je povečava enaka 1. Pazili smo tudi na situacijo prehoda na naslednjo ali



Slika 4.10: Prikaz branja revije v spletnem bralniku na mobilnem telefonu z operacijskim sistemom Android



Slika 4.11: Prikaz branja revije v spletnem bralniku na namiznem računalniku

prejšnjo stran ob trenutni povečavi večji od 1. Takrat smo na ustrezna mesta okoli trenutne strani ustrezno naložili slike primerne kakovosti (slike 4.9, 4.10 in 4.11). Ker nam knjižnica iScroll nudi tudi proženje dogodka ob zaustavitvi drsenja strani, smo lahko naš bralnik razširili še z nalaganjem slik okoli ciljne strani v primeru, da je uporabnik na to stran zdrsel zelo hitro.

Naš bralnik podpira tudi dvodimenzionalno listanje revije, kar pomeni, da so lahko posamezni članki ali zgodbe v reviji razvrščene horizontalno druga ob drugi, za branje pa bralec po posamezni zgodbi lista vertikalno. To smo enostavno rešili z razširitvijo vtičnika Trafika tako, da zna zaznati, če ima stran zgeneriranih več podstrani. To pomeni, da bo moral vtičnik na tem mestu dovoliti tudi vertikalno listanje. Ko uporabnik konča z vertikalnim listanjem zgodbe, mu ni treba listati nazaj na vrh zgodbe, ampak lahko enostavno nadaljuje horizontalno naprej po zgodbah.

Uporabniku, ki do bralnika dostopa z namiznim računalnikom (slika 4.11) so na voljo podobne geste kot na mobilnih napravah, z izjemo povečevanja, ki je omogočeno s premikanjem kolesca na miški.

## 4.4 Optimizacija z uporabo novih tehnologij

Med implementacijo bralnika smo prišli na idejo, da bi že naložene slike shranjevali v predpomnilnik in se tako izognili nalaganju s strežnika. Na srečo je to z uporabo HTML5 lokalnega pomnilnika (angl. local storage) sedaj mogoče, vendar je velikost tega na žalost omejena. Na voljo imamo nekako največ 10MB prostora. HTML5 lokalni pomnilnik je v bistvu podatkovna baza znotraj uporabnikovega brskalnika, ki omogoča shranjevanje parov ključ/vrednost, tako kot na primer zgoščevalna tabela. Idejo smo realizirali tako, da smo napisali jQuery vtičnik za delo z lokalnim pomnilnikom. Izvorna koda je na voljo v prilogi tega diplomskega dela. Ker smo slike standardne ločljivosti najprej nalagali tako, da smo elementu nastavili lokacijo do slike na strežniku, smo na tem mestu naleteli na težavo,

saj v lokalni pomnilnik datotek nismo mogli shranjevati. Na srečo smo predoglede slik že prej prenašali kot besedilo v formatu Base64 in jih nato s pomočjo CSS lastnosti `background-image` pretvarjali nazaj v slike, zato smo lahko isti postopek uporabili tudi pri slikah standardne velikosti. Sedaj smo imeli sliko v tekstovni obliki in smo jo lahko shranili v lokalni pomnilnik. Glede na to, da imamo v vsakem trenutku v glavnem pomnilniku najmanj tri slike in največ pet, smo funkcionalnost implementirali tako, da v lokalni pomnilnik shranjuje slike, ki bi jih sicer iz glavnega pomnilnika morali brisati (se pravi, da so od trenutne slike oddaljene več kot dve mesti). Ko se uporabnik z listanjem pomika nazaj po reviji in naleti na stran, ki jo je v preteklosti že naložil, se ta zelo hitro naloži iz lokalnega pomnilnika (slika 4.12).

10MB prostora v lokalnem pomnilniku zadostuje za približno 100 slik standardne velikosti, kar se v osnovi mogoče sliši veliko, vendar je za prenos celotne številke publikacije skupaj z večjimi slikami na žalost premalo.

Naša aplikacija uporablja veliko animacij, predvsem za prehajanje med stranmi in povečevanje. Kot primer vzemimo animacijo prehoda na naslednjo stran oziroma animacijo pozicije elementa. Ponavadi bi to implementirali v JavaScript knjižnici jQuery s pomočjo funkcije `animate`, kjer bi nastavili vrednost premika in dolžino animacije, kot na primer:

```
$(".stran"+stevilkaStrani).animate({ "left": "+=500px" },  
1000);
```

Problem te implementacije je, da se zgoraj navedena koda izvede v centralni procesni enoti. To na počasnejših napravah povzroči zatikanje in poslabša uporabniško izkušnjo. V naši spletni aplikaciji smo zato uporabili transformacije, ki jih ponuja CSS3. Te v naprednejših mobilnih brskalnikih pri animaciji uporabljajo strojno pospeševanje [15], kar pomeni računanje animacij s pomočjo grafične procesne enote. Primer CSS različice zgornje funkcionalnosti za pogon WebKit:

```
-webkit-transition: all 1s ease-in;  
-webkit-transform: translate3d(500px,0px,0px);
```



Slika 4.12: Slikovna predstavitev optimizacije bralnika s pomočjo tehnologije lokalnega pomnjenja

## 4.5 Težave pri razvoju

Pri razvoju spletnega portala in bralnika smo imeli največ težav z detekcijo uporabniških dotikov in potez z JavaScriptom in pravilno animacijo povečevanja ter prehajanja med stranmi. Na težave smo naleteli tudi pri obremenitvi naprave, če smo naenkrat naložili preveč slik. V takem primeru je lahko prišlo tudi do sesutja mobilnega spletnega brskalnika, zato smo morali implementirati pameten sistem nalaganja slik.

Manjšo težavo so nam predstavljali tudi izzivi pri načrtovanju razvoja, saj nismo bili prepričani kateri pristop je najboljši pri generiranju slik. Na začetku smo se odločili za generiranje slik na zahtevo glede na velikost naprave. Na koncu smo ugotovili, da je večina naprav standardnih velikosti in se odločili za tri velikosti slik:  $320 \times 480$ ,  $488 \times 656$  in  $1058 \times 1422$  slikovnih elementov.



# Poglavje 5

## Rezultati

S stališča razvoja aplikacije Trafika, je bila implementacija na domorodnih napravah dokaj podobna in se je začela z vzpostavitvijo glavnega pogleda. Potem smo se lotili komunikacije s strežnikom, saj smo morali naložiti seznam revij in njihove naslovnice z opisi. Poleg logike za prikaz številke, smo implementirali prenos številke s strežnika v aplikacijo in lokalno shranjevanje za kasnejše branje. Nato je nastopil težji del – implementacija samega bralnika. Na iOS platformi to niti ni bil problem, saj Apple ponuja odlično dokumentacijo na temo branja PDF datotek. Ostalo nam je še prikazovanje, ki pa nam ni predstavljalo problema, saj so že osnovni gradniki omogočali podobno funkcionalnost in smo jih le razširili. Nekaj dela smo vložili tudi v sistem predpomnenja PDF strani za bolj tekoče prikazovanje na napravah in manjši odtis v pomnilniku, podobno kot smo to počeli v spletni aplikaciji. Na težave smo naleteli na Android platformi, saj je dokumentacija na temo branja PDF datotek zelo slaba. Sorodne implementacije so bile sicer že razvite, vendar nam večina performančno ni ustrezala, vendar nam je na koncu implementacija z veliko truda le uspela. Kot že omenjeno v poglavju o implementaciji samega spletnega bralnika, je bila ta od treh najenostavnejša.

V trenutni fazi funkcionalno spletna aplikacija za domorodnima aplikacijama zaostaja, vendar smo v primerjavi upoštevali implementacijo enakih

oziroma podobnih funkcionalnosti.

## 5.1 Razlika v času implementacije za različne platforme glede na funkcionalnost

V smislu implementacije samega bralnika revij smo pri HTML5 rešitvi porabili približno enako časa kot pri razvoju ekvivalentne rešitve za operacijski sistem iOS in sicer okoli dva inženir tedna. Precej več pa je trajal razvoj bralnika za operacijski sistem Android in sicer več kot en inženir mesec. Problem je bil v tem, da je bilo treba s poskušanjem implementirati najboljšo izmed knjižnic za branje in prikaz datotek v PDF formatu.

Pri implementaciji prikaza seznama publikacij in njihovih številke ter pripravo na branje vodi HTML5 rešitev, saj smo za implementacijo porabili zgolj 4 inženir dan. Za ekvivalentno rešitev smo pri iOS in Android porabili več kot 11 inženir dni za vsako od implementacij, predvsem na račun procesiranja informacij in prenos številke na napravo ter lokalno shranjevanje.

Skupen čas razvoja za posamezno platformo si lahko ogledamo v tabeli ??.

čas implementacije [ID]	
spletna aplikacija	<b>19 dni</b>
iOS aplikacija	<b>25 dni</b>
Android aplikacija	<b>40+ dni</b>

Tabela 5.1: Skupen čas implementacije v inženir dneh

## 5.2 Pridobljen del trga

Glede na porazdelitev trga smo na trgu mobilnih naprav z našo spletno aplikacijo pokrili dodatnih 18% mobilnih naprav [2]. V ta delež spadajo operacijski sistemi Symbian, BlackBerry OS, Linux, Windows Phone 7 ipd.

Seveda pa ne smemo pozabiti na ogromen delež ostalih naprav, ki dostopajo do spleta, kot so namizni in prenosni računalniki, pametne televizije, skratka vse naprave, ki imajo spletni brskalnik.



# Poglavje 6

## Zaključek

Skozi diplomsko nalogo smo spoznali, da nam je implementacija spletne aplikacije vzela manj časa in hkrati pokrila večino operacijskih sistemov. Primorani smo se bili odreči nekaterim funkcionalnostim domorodnih verzij aplikacije, na primer branje številke brez internetne povezave.

Ko sedaj primerjamo delovanje domorodnih s spletno mobilno aplikacijo, se nekako ne moremo otresti občutka, da spletna aplikacija deluje manj tekoče kot domorodna. To smo sicer pričakovali že na začetku, saj je naša aplikacija procesorsko in pomnilniško zelo zahtevna. Ker platforma Trafika sledi sodobnim trendom, bomo v našo spletno aplikacijo v prihodnosti dodali tudi funkcionalnost za prikaz posameznih člankov v tekstovni obliki. Takrat bomo lahko spletno aplikacijo še bolj približali domorodnim, saj bi lahko celotno številko teoretično shranili na uporabnikovi napravi in s tem omogočili tudi branje brez internetne povezave. Seveda bi lahko del revije, ki se shranjuje v lokalni pomnilnik uporabniku že zdaj ponudili za branje brez internetne povezave, vendar shranjeni del na žalost ne obsega celotne številke, ki je večja od dovoljene kvote shranjenih podatkov.

Razvijalcem, ki se odločajo med implementacijo spletne aplikacije in implementacijo domorodne aplikacije bi na tem mestu priporočili podrobno preučitev specifikacije potencialne aplikacije. Če ta vključuje zahtevnejše operacije in veliko animacij, naj na spletu poiščejo sorodne rešitve in pre-

verijo njihove implementacije, saj zaenkrat domorodne aplikacije v tem pogledu še vedno vodijo. Vsem ostalim bi priporočali izgradnjo spletne mobilne aplikacije.

## 6.1 Možne izboljšave

Skozi proces implementacije spletne aplikacije Trafika smo naleteli na vprašanje varnosti. Kako uporabniku preprečiti nedovoljeno shranjevanje celotne številke revije in nepooblaščen distribucijo le-te? Na slike nismo želeli dodati vodnega žiga saj nismo želeli pokvariti uporabniške izkušnje. Drugih rešitev v praksi nismo preizkušali, saj bi lahko dovolj vztrajni uporabniki preprosto zajeli zaslonsko sliko in tako pridobili kopijo strani. Vprašanje varnosti smo tako zaenkrat pustili odprto, saj predstavlja velik izziv in je mogoče predmet kakšnega drugega diplomskega dela. V domorodnih aplikacijah je trenutno v implementaciji enkripcija PDF datotek z uporabniškim ključem, kar pomeni, da bo prenesena številka na voljo le enemu uporabniku. Tudi sicer je v domorodnih aplikacijah dostop do shranjenih podatkov otežen, za kakovosten zajem zaslonske slike pa bi se morali potruditi s sestavljanjem večih slik ene strani. V spletnem bralniku smo s prikazovanjem slik v formatu Base64 sicer rešili problem enostavnega shranjevanja slike z desnim klikom miške, vendar lahko bolj izkušeni uporabniki slike vseeno shranijo brez zajemanja zaslona.

Ena izmed možnih izboljšav je implementacija interaktivnih vsebin, na primer prikaz videa ali slikovnih galerij znotraj bralnika.

Za prikaz slik višjih kakovosti bi lahko v JavaScriptu implementirali metodo za prikaz visokokakovostnih slik na mobilnih napravah [30].

# Dodatek A

## Skripta za nadzor lokalnega pomnilnika

Priložena je JavaScript implementacija skripte za nadzor lokalnega pomnilnika v obliki jQuery vtičnika. Skripta sama skrbi za preverjanje napak, uporabniku pa omogoča shranjevanje in brisanje parov ključ/vrednost ali pa brisanje celotnega lokalnega pomnilnika.

```
(function( $ ) {  
  
    function LocalStorageManager(o){  
  
        self.options = $.extend( {  
            'storageSize' : 10,  
        }, o);  
  
        self.init = function(){  
            return self;  
        };  
  
        self.cache = function(pageNum, imageStr) {  
            return _internal.add(pageNum, imageStr);  
        }  
  
        self.get = function(pageNum) {
```

```
        return _internal.get(pageNum);
    }

    self.remove = function(pageNum) {
        return _internal.remove(pageNum);
    }

    self.nuke = function() {
        return _internal.nuke();
    }

    var _internal = {
        'add' : function(pageNum, imageStr) {
            var item;
            var itemKey = "page"+pageNum;
            if (typeof(localStorage) === 'undefined' ) {
                return 0;
            } else {

                try {
                    localStorage.setItem(itemKey,
                                        imageStr);
                    return 1;
                } catch (e) {
                    if (e === QUOTA_EXCEEDED_ERR) {
                        return 0;
                    }
                }
            }
        },
        'get' : function(pageNum) {
            var itemKey = "page"+pageNum;
            if (typeof(localStorage) === 'undefined' ) {
                return 0;
            } else {
                try {
```

```
        var value = localStorage.  
            getItem(itemKey);  
        if(value) {  
            return value;  
        } else {  
            return 0;  
        }  
    } catch (e) {  
        return 0;  
    }  
}  
return 0;  
},  
'remove' : function(pageNum){  
    var itemKey = "page"+pageNum;  
    if (typeof(localStorage) == 'undefined' ) {  
        return 0;  
    } else {  
        if ( localStorage.getItem(itemKey)) {  
            localStorage.removeItem(itemKey  
                );  
            return 1;  
        }  
        else {  
            return 0;  
        }  
    }  
    return 0;  
},  
'nuke':function(){  
    if (typeof(localStorage) == 'undefined' ) {  
        return 0;  
    } else {  
        localStorage.clear();  
        return 1;  
    }  
}
```

```
        return 0;
    }

};

return self.init();
};

$.fn.lsManager = function( el ) {
return new LocalStorageManager( el );
};
})( jQuery );
```

# Literatura

- [1] (2012) Apple's App Store Downloads Top 25 Billion. Dostopno na:  
<http://www.apple.com/pr/library/2012/03/05Apples-App-Store-Downloads-Top-25-Billion.html>
  
- [2] (2012) Android- and iOS-Powered Smartphones Expand Their Share of the Market in the First Quarter, According to IDC. Dostopno na:  
<http://www.idc.com/getdoc.jsp?containerId=prUS23503312>
  
- [3] (2012) Open Handset Alliance. Dostopno na:  
[http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html)
  
- [4] (2012) Android@Mobile World Congress: It's all about the ecosystem. Dostopno na:  
<http://googlemobile.blogspot.com/2012/02/androidmobile-world-congress-its-all.html>
  
- [5] (2012) Google Play About To Pass 15 Billion App Downloads? Pssht! It Did That Weeks Ago. Dostopno na:  
<http://goo.gl/bbXTp>
  
- [6] (2012) Windows Phone Marketplace passes 90,000 apps. Dostopno na:  
[http://allaboutwindowsphone.com/news/item/14819\\_Windows\\_Phone\\_Marketplace\\_pass.php](http://allaboutwindowsphone.com/news/item/14819_Windows_Phone_Marketplace_pass.php)
  
- [7] (2012) App Hub - how it works? Dostopno na:  
[http://create.msdn.com/en-US/home/about/how\\_it\\_works](http://create.msdn.com/en-US/home/about/how_it_works)

- 
- [8] (2012) Google App Engine. Dostopno na:  
<https://developers.google.com/appengine/>
- [9] (2012) iScroll 4. Dostopno na:  
<http://cubiq.org/iscroll-4>
- [10] (2012) Mobile vs. Desktop from June 2011 to June 2012. Dostopno na:  
[http://gs.statcounter.com/#mobile\\_vs\\_desktop-ww-monthly-201106-201206](http://gs.statcounter.com/#mobile_vs_desktop-ww-monthly-201106-201206)
- [11] (2012) Mobile Intent Index. Dostopno na:  
<http://www.intentindex.com/mobile/>
- [12] (2012) Kindle Books. Dostopno na:  
<http://www.amazon.com/Books-Kindle/b?node=154606011>
- [13] (2012) Survey of Kindle, Nook, iPad, Sony and OverDrive eBook Store Collection Size. Dostopno na:  
<http://goo.gl/Wpnpe>
- [14] (2012) Why BlueToad? Dostopno na:  
<http://www.bluetoad.com/BlueToad/why-BlueToad>
- [15] (2012) Improving the performance of your HTML5 app. Dostopno na:  
<http://www.html5rocks.com/en/tutorials/speed/html5/>
- [16] (2012) HTML5 localStorage. Dostopno na:  
<http://paperkilledrock.com/2010/05/html5-localstorage-part-one/>
- [17] (2012) Modernizr. Dostopno na:  
<http://modernizr.com/>
- [18] (2012) Transaction Fees - Google Play for Developers Help. Dostopno na:  
<http://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=112622>

- 
- [19] (2012) Distribute your App - iOS Developer Program - Apple Developer. Dostopno na:  
<https://developer.apple.com/programs/ios/distribute.html>
- [20] (2012) The Future of Magazines. Dostopno na:  
<http://thefutureofpublishing.com/industries/the-future-of-magazines/>
- [21] (2012) Zinio Fusion. Dostopno na:  
<http://www.zinio.com/press/press-release.jsp?pressreleaseid=pr148400>
- [22] (2012) Estimates of Mobile Phones and Smartphones Supporting HTML5. Dostopno na:  
<http://letsgomo.com/mobile-web/estimates-of-mobile-phones-and-smartphones-supporting-html5/>
- [23] (2012) CSS contents and browser compatibility. Dostopno na:  
<http://www.quirksmode.org/css/contents.html/>
- [24] C. Lindley, *“jQuery Enlightenment”*, lulu.com, 2009
- [25] J. Meyer, *“HTML5 and JavaScript Projects”*, New York: Apress, 2011
- [26] D. Sanderson, *“Programming Google App Engine”*, California: O’Reilly Media, Inc., 2010
- [27] A. Goldstein, L. Lazaris, E. Weyl, *“HTML5 & CSS3 for the Real World”*, Australia: SitePoint Pty. Ltd., 2011
- [28] E. Marcotte, *“Responsive Web Design”*, New York: A Book Apart, 2011
- [29] M. Zimic, *“Medplatformski razvoj mobilnih aplikacij”*, diplomska naloga, Fakulteta za računalništvo in informatiko: Ljubljana, 2012, str. 21-54.

- [30] A. Sršen, “*Mobilna aplikacija za pregledovanje slik visokih ločljivosti*”, diplomska naloga, Fakulteta za računalništvo in informatiko: Ljubljana, 2012, str. 34-44.

# Tabele

5.1 Skupen čas implementacije v inženir dneh . . . . .	48
--	----



# Slike

2.1	Izgled iPad aplikacije platforme Trafika . . . . .	5
2.2	Izgled iPad aplikacije platforme Zinio . . . . .	6
2.3	Izgled spletnega bralnika platforme BlueToad na iPadu . .	7
3.1	Izgled integriranega programskega okolja Xcode . . . . .	13
3.2	Izgled integriranega programskega okolja Eclipse in Android emulatorja . . . . .	15
3.3	Izgled integriranega programskega okolja Visual Studio 2010 Express for Windows Phone in Windows Phone 7 emulatorja	16
3.4	Izgled urejevalnika besedil Notepad++ v katerem smo raz- vijali spletno aplikacijo . . . . .	17
4.1	Prikaz spletne strani z odzivnim spletnim oblikovanjem na različnih širinah spletnega brskalnika . . . . .	30
4.2	Prijava v sistem . . . . .	33
4.3	Dostop do spletnega portala s tabličnim računalnikom iPad	35
4.4	Dostop do spletnega portala z Android mobilnim telefonom	36
4.5	Dostop do spletnega portala z namiznim računalnikom . .	36
4.6	Izbiranje številke za branje . . . . .	37
4.7	Ikona spletne aplikacije na namizju iPada . . . . .	37
4.8	Prikaz predogleda oziroma hitre navigacije po vseh straneh	39
4.9	Prikaz branja revije v spletnem bralniku na tabličnem računalniku iPad. Prikazana stran je povečana . . . . .	40

4.10 Prikaz branja revije v spletnem bralniku na mobilnem telefonu z operacijskim sistemom Android . . . . .	41
4.11 Prikaz branja revije v spletnem bralniku na namiznem računalniku . . . . .	41
4.12 Slikovna predstavitev optimizacije bralnika s pomočjo tehnologije lokalnega pomnjenja . . . . .	44