



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Smetanova ulica 17
2000 Maribor, Slovenija



Andreja Fajfar

ANALIZA ZMOGLJIVOSTI MOBILNIH APLIKACIJ, TEMELJEČIH NA SPLETNIH TEHNOLOGIJAH

Magistrsko delo

Maribor, april 2016

**ANALIZA ZMOGLJIVOSTI MOBILNIH APLIKACIJ,
TEMELJEČIH NA SPLETNIH TEHNOLOGIJAH**
Magistrsko delo

Študentka: Andreja Fajfar
Študijski program: Študijski program 2. stopnje
Informatika in tehnologije komuniciranja
Smer: Informatika in tehnologije komuniciranja
Mentor: doc. dr. Boštjan Šumak, univ. dipl. inž. rač. in inf.
Lektor(ica): Tamara Korade, prof. slov.



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko
Smetanova ulica 17
2000 Maribor, Slovenija

FERI

Številka: E5012589

Datum in kraj: 04. 05. 2015, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Ur. l. RS, št. 46/2012)
izdajam

SKLEP O MAGISTRSKEM DELU

1. **Andreji Fajfar**, študentki študijskega programa 2. stopnje INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA, se dovoljuje izdelati magistrsko delo.

2. Tema magistrskega dela je pretežno s področja Inštituta za informatiko.

MENTOR: doc. dr. Boštjan Šumak

3. Naslov magistrskega dela:

ANALIZA ZMOGLJIVOSTI MOBILNIH APLIKACIJ, TEMELJEČIH NA SPLETNIH TEHNOLOGIJAH

4. Naslov magistrskega dela v angleškem jeziku:

PERFORMANCE ANALYSIS OF MOBILE APPLICATIONS BASED ON WEB TECHNOLOGIES

5. Magistrsko delo je potrebno izdelati skladno z »Navodili za izdelavo magistrskega dela«. Skladno s 6. členom *Pravilnika o postopku priprave in zagovora magistrskega dela na študijskih programih 2. stopnje UM* je bilo odobreno podaljšanje roka za oddajo magistrskega dela do 23. 05. 2016. Magistrsko delo študent-ka odda v treh izvodih (dva trdo vezana izvoda in en v spiralo vezan izvod) v pristojni referat ter elektronski izvod v Digitalno knjižnico Univerze v Mariboru.

Pravni pouk: Zoper ta sklep je možna pritožba na senat članice v roku 15 dni.

Obvestiti:

1. kandidatko
2. mentorja
3. odložiti v arhiv



Dekan:
red. prof. dr. Borut Žalik

ZAHVALA

Zahvaljujem se staršem, ki so mi omogočili študij, bratu za motivacijo, Jessici R. Mattiace za neizmerno podporo ter prijateljem za dolge noči, napolnjene z globokimi pogovori, smehom in ljubeznijo. Brez vas ne bi bilo tega magistrskega dela.

Prav tako gre posebna zahvala mentorju g. Boštjanu Šumaku za strokovno vodenje, motiviranje in pomoč.

Analiza zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah

Ključne besede: Mobilne aplikacije, spletne tehnologije, zmogljivost, analiza zmogljivosti, zmogljivost mobilnih aplikacij, spletne mobilne aplikacije, domorodne aplikacije.

UDK: 621.397.7-026.26:004.774(043.2)

Povzetek

V magistrski nalogi raziskujemo analizo zmogljivosti mobilnih aplikacij, temelječih na domorodnih in spletnih tehnologijah. V uvodnem delu naloge predstavimo področje razvoja mobilnih aplikacij, njihove prednosti in slabosti ter razvojna orodja. V nalogi smo predlagali model za ocenjevanje zmogljivosti mobilnih aplikacij, ki smo ga preverili s pomočjo eksperimenta, v okviru katerega smo analizirali zmogljivost, zadovoljstvo uporabnikov in uporabniško izkušnjo. Predlagani model smo ovrednotili na primeru aplikacij Facebook, Twitter, YouTube in GMail. Na osnovi empiričnih podatkov, pridobljenih v eksperimentu, smo analizirali in primerjali dojemanje odzivnosti, hitrosti, enostavnosti uporabe, vizualne izdelave, uporabniške izkušnje in zadovoljstva uporabnikov. Rezultati na osnovi empiričnih podatkov so pokazali višjo stopnjo dojemanja uporabniške izkušnje ob uporabi domorodnih mobilnih rešitev v primerjavi z njihovimi spletno zasnovanimi rešitvami.

Performance analysis of mobile applications based on web technologies

Key words: Mobile applications, web technologies, performance analysis, mobile application performance, mobile web applications, hybrid mobile applications, native mobile applications.

UDK: 621.397.7-026.26:004.774(043.2)

Abstract

The following master thesis analyzes the performance abilities of native and web mobile applications. We introduce the current state of mobile applications, explore the main paradigms as well as their pros and cons, along with tools of the trade. Based on the research and gathered data, a research model is introduced with a set of confirmed performance factors. An experiment was performed in order to justify the research model and its performance factors, as well as user experience and satisfaction. The model was evaluated with mobile applications such as Facebook, Twitter, YouTube and GMail. Based on the gathered empirical data from the experiment, the data was analyzed and compared by perceived responsiveness, speed, ease of use, visual interface, user experience and user satisfaction. Results based on empirical data have shown a higher level of perceived user experience and satisfaction while using native mobile applications compared to their web counterparts.

KAZALO

1	UVOD	1
1.1	Opis področja magistrskega dela	1
1.2	Namen in cilj magistrskega dela.....	3
1.3	Predpostavke in omejitve	4
1.4	Kratek opis celotnega dela	5
2	STANJE MOBILNIH APLIKACIJ DANES	6
3	DOMORODNE MOBILNE APLIKACIJE	11
3.1	Razvoj s pomočjo SDK in API	12
3.2	Razvojna orodja.....	14
3.2.1	Android SDK.....	15
3.2.2	Xcode.....	16
3.2.3	Windows Phone SDK	18
3.3	Prednosti in slabosti domorodnih mobilnih aplikacij.....	21
4	MOBILNE APLIKACIJE, RAZVITE S SPLETNIMI TEHNOLOGIJAMI	22
4.1	Spletne mobilne aplikacije in odzivne spletne strani	24
4.2	Spletne mobilne aplikacije, WebView in WebKit	26
4.2.1	Chrome Developer Tools.....	28
4.2.2	WebView in WebKit	29
4.3	Razvojna orodja.....	31
4.3.1	JQuery Mobile in Codiqa	31
4.3.2	Sencha Touch	32
4.4	Kratka predstavitev spletnih tehnologij.....	34
4.4.1	HyperText Markup Language (HTML 5).....	34
4.4.2	Cascading Style Sheet (CSS3)	34
4.4.3	JavaScript.....	35
4.5	Prednosti in slabosti mobilnih aplikacij razvitih s pomočjo spletnih tehnologij	36

5	RAZISKAVA	38
5.1	Izvedba eksperimenta	42
5.1.1	Omejitve eksperimenta.....	44
5.1.2	Analiza veljavnosti in zanesljivosti rezultatov	45
5.1.3	Vzorec eksperimenta.....	47
5.2	Analiziranje HAR in Timeline datotek	47
5.3	Analiza podatkov	51
5.3.1	Analiza podatkov mobilne aplikacije Twitter.....	52
5.3.2	Analiza podatkov mobilne aplikacije Facebook.....	54
5.3.3	Analiza podatkov mobilne aplikacije YouTube	56
5.3.4	Analiza podatkov mobilne aplikacije GMail	58
5.3.5	Analiza podatkov za dokazovanje enostavnosti uporabe, kakovosti, odzivnosti in vizualne izdelave aplikacije	60
5.3.6	Analiza uporabniškega zadovoljstva	67
5.3.7	Analiza korelacije med številom zahtev in časom nalaganja (onLoad in DOMContentLoaded) pri spletnih mobilnih aplikacijah	69
5.3.8	Analiza zmogljivosti mobilnih aplikacij temelječih na spletnih tehnologijah	71
5.3.9	Analiza podatkov za uporabniško zadovoljstvo in uporabniško izkušnjo	77
5.4	Analiziranje raziskovalnih vprašanj	84
5.5	Analiziranje hipotez	86
6	SKLEP	88
7	VIRI	90
8	PRILOGE	97
	PRILOGA A – NAVODILA ZA IZVAJANJE EKSPERIMENTA	98
	PRILOGA B – PROCES IZVEDBE EKSPERIMENTA	101
	PRILOGA C – GRAFI IN TABELE STATISTIČNIH ANALIZ MOBILNIH APLIKACIJ ..	114

KAZALO SLIK

SLIKA 2.1: STATISTIČNA SLIKA RAZVOJA APLIKACIJ ZA RAZLIČNE PLATFORME [69]	7
SLIKA 2.2: NAPOVED RAZVOJNIH PRISTOPOV ZA KONČNE UPORABNIKE V LETU 2016 [69].....	8
SLIKA 2.3: NAPOVED RAZVOJNIH PRISTOPOV ZA PODJETJA V LETU 2016 [69].....	8
SLIKA 3.1: POMEMBNOST BIDIREKTIVNOSTI V MOBILNIH APLIKACIJAH [31]	14
SLIKA 3.2: FAZE RAZVOJA DOMORODNE ANDROID MOBILNE APLIKACIJE [3]	16
SLIKA 3.3: RAZVOJ DOMORODNE APLIKACIJE ZA APPLE PLATFORMO [35].....	17
SLIKA 3.4: PRILAGAJANJE WINDOWS APLIKACIJ NA RAZLIČNIH NAPRAVAH [37]	19
SLIKA 3.5: TEHNIKA NADOMEŠČANJA UI-ELEMENTOV [37]	20
SLIKA 3.6: PRELOMNE TOČKE V WINDOWS DOMORODNI APLIKACIJI [38].....	20
SLIKA 4.1: PRIMERJAVA ODZIVNOSTI SPLETNE STRANI GLEDE NA NAPRAVO [58]	26
SLIKA 4.2: PRIKAZOVANJE SPLETNE MOBILNE APLIKACIJE V ANDROID BRSKALNIKU [2]	27
SLIKA 4.3: DELOVNI TOK DELOVANJA WEBKIT OGRODJA [27]	30
SLIKA 4.4: PRIMER SPLETNE MOBILNE APLIKACIJE USTVARJENJE S POMOČJO JQUERY MOBILE [41]	31
SLIKA 4.5: PRIMER SPLETNE MOBILNE APLIKACIJE, USTVARJENE S POMOČJO SENCHA TOUCH 2 [65].....	33
SLIKA 4.6: ODZIVNOST SPLETNE STRANI [61].....	35
SLIKA 5.1: TEORETIČNI MODEL ANALIZE	40
SLIKA 5.2: UVODNI DEL ANKETE ZA EKSPERIMENT.....	41
SLIKA 5.3: ANALIZA ANKETE PO KONČANEM EKSPERIMENTU	42
SLIKA 5.4: POTEK EKSPERIMENTA.....	43
SLIKA 5.5: ANALIZA HAR DATOTEKE EKSPERIMENTA	48
SLIKA 5.6: PRIMER OBDELAVE HAR DATOTEKE Z AVTORSKIM PROGRAMOM	49
SLIKA 5.7: PRIDOBIVANJE PODATKOV S POMOČJO FIDDLER ORODJA	50
SLIKA 5.8: PREGLED ČASOVNICE V CHROME DEVTOOLS ORODJU.....	51
SLIKA 5.9: GRAF PRIKAZOVANJA UPORABNIŠKEGA ZADOVOLJSTVA MED POSAMEZNIMI TIPI APLIKACIJ	68
SLIKA 5.10: ŠTEVILO ZAHTEV V KORELACIJI Z ONLOAD ČASOM NALAGANJA	71
SLIKA 5.11: ŠTEVILO ZAHTEV V KORELACIJI Z DOMCONTENTLOADED ČASOM NALAGANJA.....	71
SLIKA 8.1: PRIKAZ SESTAVLJENOSTI POSAMEZNIH ZAHTEV VSAKEGA UDELEŽENCA – TWITTER	117
SLIKA 8.2: PRIKAZ SESTAVNIH DELOV PO VELIKOSTI POSAMEZNE ZAHTEVE – TWITTER.....	118
SLIKA 8.3: KOMPLEKSNOŠT VSEBINE PRI SPLETNI RAZLIČICI TWITTER APLIKACIJE	118
SLIKA 8.4: POVPREČNE VREDNOSTI ŠTEVILA ZAHTEV V KORELACIJI Z NJIHOVO VELIKOSTJO – TWITTER..	119
SLIKA 8.5: PRIKAZOVANJE INICIALIZACIJSKEGA ČASA TWITTER APLIKACIJE	119
SLIKA 8.6: KATEGORIZIRANI ČAS INICIALIZACIJE TWITTER APLIKACIJE.....	120
SLIKA 8.7: POVPREČEN ČAS PORABLJEN V BRSKALNIKU – TWITTER APLIKACIJA.....	120
SLIKA 8.8: DOSTOPNOST TWITTER APLIKACIJE PRI POSAMEZNI SEJI UDELEŽENCA	121
SLIKA 8.9: KONSISTENTNOST ODZIVA TWITTER APLIKACIJE V POSAMEZNI SEJI UDELEŽENCA	121
SLIKA 8.10: ČAS NALAGANJA TWITTER.....	122

SLIKA 8.11: UPORABNIŠKA IZKUŠNJA PRI TWITTER APLIKACIJI	122
SLIKA 8.12: PRIKAZ SESTAVLJENOSTI POSAMEZNIH ZAHTEV VSAKEGA UDELEŽENCA – FACEBOOK	123
SLIKA 8.13: PRIKAZ SESTAVNIH DELOV PO VELIKOSTI POSAMEZNE ZAHTEVE – FACEBOOK	123
SLIKA 8.14: KOMPLEKSNOŠT VSEBINE PRI SPLETNI RAZLIČICI FACEBOOK APLIKACIJE.....	124
SLIKA 8.15: POVPREČNE VREDNOSTI ŠTEVILA ZAHTEV V KORELACIJI Z NJIHOVO VELIKOSTJO – FACEBOOK	124
SLIKA 8.16: PRIKAZOVANJE INICIALIZACIJSKEGA ČASA FACEBOOK APLIKACIJE.....	125
SLIKA 8.17: KATEGORIZIRANI ČAS INICIALIZACIJE FACEBOOK APLIKACIJE	125
SLIKA 8.18: POVPREČEN ČAS PORABLJEN V BRSKALNIKU - FACEBOOK APLIKACIJA	126
SLIKA 8.19: DOSTOPNOST FACEBOOK APLIKACIJE PRI POSAMEZNI SEJI UDELEŽENCA	126
SLIKA 8.20: KONSISTENTNOST ODZIVA FACEBOOK APLIKACIJE V POSAMEZNI SEJI UDELEŽENCA	127
SLIKA 8.21: UPORABNIŠKA IZKUŠNJA PRI FACEBOOK APLIKACIJI	127
SLIKA 8.22: ČAS NALAGANJA FACEBOOK	128
SLIKA 8.23: PRIKAZ SESTAVLJENOSTI POSAMEZNIH ZAHTEV VSAKEGA UDELEŽENCA – YOUTUBE.....	128
SLIKA 8.24: PRIKAZ SESTAVNIH DELOV PO VELIKOSTI POSAMEZNE ZAHTEVE – YOUTUBE	129
SLIKA 8.25: KOMPLEKSNOŠT VSEBINE PRI SPLETNI RAZLIČICI YOUTUBE APLIKACIJE	129
SLIKA 8.26: POVPREČNE VREDNOSTI ŠTEVILA ZAHTEV V KORELACIJI Z NJIHOVO VELIKOSTJO – YOUTUBE	130
SLIKA 8.27: PRIKAZOVANJE INICIALIZACIJSKEGA ČASA YOUTUBE APLIKACIJE	130
SLIKA 8.28: KATEGORIZIRANI ČAS INICIALIZACIJE YOUTUBE APLIKACIJE	131
SLIKA 8.29: POVPREČEN ČAS PORABLJEN V BRSKALNIKU - YOUTUBE APLIKACIJA	131
SLIKA 8.30: DOSTOPNOST YOUTUBE APLIKACIJE PRI POSAMEZNI SEJI UDELEŽENCA.....	132
SLIKA 8.31: KONSISTENTNOST ODZIVA YOUTUBE APLIKACIJE V POSAMEZNI SEJI UDELEŽENCA.....	132
SLIKA 8.32: ČAS NALAGANJA YOUTUBE	133
SLIKA 8.33: UPORABNIŠKA IZKUŠNJA YOUTUBE APLIKACIJE	133
SLIKA 8.34: PRIKAZ SESTAVLJENOSTI POSAMEZNIH ZAHTEV VSAKEGA UDELEŽENCA – GMAIL	134
SLIKA 8.35: PRIKAZ SESTAVNIH DELOV PO VELIKOSTI POSAMEZNE ZAHTEVE – GMAIL.....	134
SLIKA 8.36: KOMPLEKSNOŠT VSEBINE PRI SPLETNI RAZLIČICI GMAIL APLIKACIJE	135
SLIKA 8.37: POVPREČNE VREDNOSTI ŠTEVILA ZAHTEV V KORELACIJI Z NJIHOVO VELIKOSTJO – GMAIL... ..	135
SLIKA 8.38: PRIKAZOVANJE INICIALIZACIJSKEGA ČASA GMAIL APLIKACIJE	136
SLIKA 8.39: KATEGORIZIRANI ČAS INICIALIZACIJE GMAIL APLIKACIJE.....	136
SLIKA 8.40: POVPREČEN ČAS PORABLJEN V BRSKALNIKU - GMAIL APLIKACIJA.....	137
SLIKA 8.41: DOSTOPNOST GMAIL APLIKACIJE PRI POSAMEZNI SEJI UDELEŽENCA.....	137
SLIKA 8.42: KONSISTENTNOST ODZIVA GMAIL APLIKACIJE V POSAMEZNI SEJI UDELEŽENCA.....	138
SLIKA 8.43: ČAS NALAGANJA GMAIL.....	138
SLIKA 8.44: UPORABNIŠKA IZKUŠNJA GMAIL APLIKACIJE	139

KAZALO TABEL

TABELA 3.1: PRIMERJAVA RAZVOJNIH OGRODIJ ZA DOMORODNE MOBILNE APLIKACIJE [34]	12
TABELA 4.1: ZAHTEVE IN NJIHOVE ARHITEKTURNE REŠITVE ZA APLIKACIJSKO MOBILNOST [40]	23
TABELA 4.2: RAZLIKA MED SPLETNIMI MOBILNIMI APLIKACIJAMI IN STRANMI [34].....	24
TABELA 5.1: UPORABLJENI STROJNA IN PROGRAMSKA OPREMA PRI EKSPERIMENTU.....	44
TABELA 5.2: ZANESLJIVOST PODATKOV PO CRONBACH MERILU	46
TABELA 5.3: IZRAČUN ZANESLJIVOSTI PODATKOV ZA AFEKTIVNO ZADOVOLJSTVO.....	46
TABELA 5.4: ANALIZA PODATKOV TWITTER APLIKACIJE.....	53
TABELA 5.5: ANALIZA PODATKOV FACEBOOK APLIKACIJE	54
TABELA 5.6: ANALIZA PODATKOV YOUTUBE APLIKACIJE	57
TABELA 5.7: ANALIZA PODATKOV GMAIL APLIKACIJE.....	59
TABELA 5.8: DESKRIPTIVNA STATISTIKA ZA APLIKACIJO FACEBOOK.....	61
TABELA 5.9: MANN-WHITNEYJEV U-TEST ZA APLIKACIJO FACEBOOK	61
TABELA 5.10: STATISTIČNI REZULTATI ZA APLIKACIJO FACEBOOK	61
TABELA 5.11: DESKRIPTIVNA STATISTIKA TESTA ZA APLIKACIJO GMAIL	63
TABELA 5.12: MANN-WHITNEYJEV U-TEST ZA APLIKACIJO GMAIL.....	63
TABELA 5.13: STATISTIČNI REZULTATI ZA APLIKACIJO GMAIL.....	64
TABELA 5.14: DESKRIPTIVNA STATISTIKA TESTA ZA APLIKACIJO TWITTER.....	64
TABELA 5.15: MANN-WHITNEYJEV U-TEST ZA APLIKACIJO TWITTER	65
TABELA 5.16: STATISTIČNI REZULTATI ZA APLIKACIJO TWITTER.	65
TABELA 5.17: DESKRIPTIVNA STATISTIKA TESTA ZA APLIKACIJO YOUTUBE	66
TABELA 5.18: MANN-WHITNEYJEV U-TEST ZA APLIKACIJO YOUTUBE	66
TABELA 5.19: STATISTIČNI REZULTATI ZA APLIKACIJO YOUTUBE.....	67
TABELA 5.20: IZRAČUN POVPREČNIH VREDNOSTI UPORABNIŠKEGA ZADOVOLJSTVA	67
TABELA 5.21: STATISTIČNI IZRAČUNI ZA UPORABNIŠKO ZADOVOLJSTVO	67
TABELA 5.22: KORELACIJA MED ŠTEVILOM ZAHTEV IN ONLOAD ČASOM NALAGANJA	69
TABELA 5.23: KORELACIJA MED ŠTEVILOM ZAHTEV IN DOMCONTENTLOADED ČASOM NALAGANJA.....	69
TABELA 5.24: FREKVENČNA ANALIZA FACEBOOK APLIKACIJE.....	74
TABELA 5.25: FREKVENČNA ANALIZA GMAIL APLIKACIJE	74
TABELA 5.26: FREKVENČNA ANALIZA TWITTER APLIKACIJE	75
TABELA 5.27: FREKVENČNA ANALIZA YOUTUBE APLIKACIJE	76
TABELA 5.28: POVZETEK OCEN FREKVENČNE ANALIZE VSEH APLIKACIJ.....	77
TABELA 5.29: ANKETNA VPRAŠANJA ZA DOLOČEVANJE AFEKTIVNEGA IN KOGNITIVNEGA ZADOVOLJSTVA ..	78
TABELA 5.30: KRITERIJI ZA UPORABNIŠKO ZADOVOLJSTVO.....	78
TABELA 5.31: KRITERIJI ZA UPORABNIŠKO IZKUŠNJO.....	79
TABELA 5.32: OCENA UPORABNIŠKE IZKUŠNJE TWITTER APLIKACIJE.....	81
TABELA 5.33: OCENA UPORABNIŠKE IZKUŠNJE GMAIL APLIKACIJE.....	81

TABELA 5.34: OCENA UPORABNIŠKE IZKUŠNJE YOUTUBE APLIKACIJE	81
TABELA 5.35: OCENA UPORABNIŠKE IZKUŠNJE FACEBOOK APLIKACIJE	82
TABELA 5.36: OCENA UPORABNIŠKEGA ZADOVOLJSTVA PRI TWITTER APLIKACIJI	83
TABELA 5.37: OCENA UPORABNIŠKEGA ZADOVOLJSTVA PRI GMAIL APLIKACIJI	83
TABELA 5.38: OCENA UPORABNIŠKEGA ZADOVOLJSTVA PRI YOUTUBE APLIKACIJI	83
TABELA 5.39: OCENA UPORABNIŠKEGA ZADOVOLJSTVA PRI FACEBOOK APLIKACIJI	83
TABELA 5.40: REZULTAT UPORABNIŠKE IZKUŠNJE IN ZADOVOLJSTVA ZA VSE APLIKACIJE.....	84
TABELA 8.1: NOTRANJE KORELACIJE MED ELEMENTI KOGNITIVNEGA ZADOVOLJSTVA	114
TABELA 8.2: CELOTNA STATISTIČNA POSTAVKA ELEMENTOV KOGNITIVNEGA ZADOVOLJSTVA.....	114
TABELA 8.3: CELOTNA STATISTIČNA POSTAVKA ELEMENTOV UPORABNIŠKE IZKUŠNJE	115
TABELA 8.4: NOTRANJE KORELACIJE MED ELEMENTI AFEKTIVNEGA ZADOVOLJSTVA	115
TABELA 8.5: CELOTNA STATISTIČNA POSTAVKA ELEMENTOV AFEKTIVNEGA ZADOVOLJSTVA.....	116
TABELA 8.6: NOVI IZRAČUN NOTRANJE KORELACIJE MED ELEMENTI AFEKTIVNEGA ZADOVOLJSTVA.....	116
TABELA 8.7: NOVI IZRAČUN STATISTIČNE POSTAVKE ELEMENTOV AFEKTIVNEGA ZADOVOLJSTVA.....	117

Uporabljene kratice

SDK	Software Development Kit
API	Application Program Interface
HTML, HTML5	HyperText Markup Language
CSS, CSS3	Cascading Style Sheets
JS	JavaScript
RWD	Responsive Web Design
OS	Operacijski Sistem
GPS	Global Positioning System
SMS	Short Message Service
GUI	Graphical User Interface
UI	User Interface
UWP	Universal Windows Platform
URL	Uniform Resource Locator
DOM	Document Object Model
W3C	World Wide Web Consortium
WHATWG	Web Hypertext Application Technology Working Group
IDE	Integrated Development Environment, Interactive Development Environment
SCM	Supply Chain Management
DOM	Document Object Model
B	Byte
kB	KiloByte
MB	Megabyte
NPS	Net Promoter Score Definition
SPSS	Statistical Package for the Social Sciences
CPU	Central Processing Unit
s	Sekunda

1 UVOD

1.1 Opis področja magistrskega dela

Vedno bolj popularne mobilne rešitve in pogostejša uporaba mobilnih aplikacij ponujajo iz dneva v dan večji nabor možnosti razvoja. Raziskave zadnjih let dokazujejo, da 80 % uporabnikov uporablja mobilne aparate in mobilne rešitve za iskanje informacij po spletu, kar pripelje do dejstva, da kar 89 % uporabnikov iz te skupine porabi določeno količino časa za uporabo raznolikih mobilnih aplikacij – moški v povprečju 29 ur in ženske v povprečju 30 ur na mesec. Preostalih 11 % uporabnikov uporablja predvsem mobilno verzijo spletnih strani namesto aplikacijskih rešitev [21].

Dejstvo je, da uporaba mobilnih aparatov in s tem neposredno tudi uporaba aplikacij narašča saj je v zadnjem letu znašala 76 % [45]. Podatki nakazujejo, da postaja uporaba mobilnih aplikacij in mobilno prilagojenih spletnih strani kritičnega pomena za rast podjetja, posameznikov in njihovih storitev.

Poznamo tri osnovne pristope k razvoju mobilnih aplikacij, kjer so rezultat domorodne, spletne ali hibridne mobilne aplikacije. V sklopu magistrskega dela se bomo osredotočili predvsem na spletne in domorodne, saj jih tudi pozneje primerjamo glede zmogljivosti, uporabniške izkušnje in zadovoljstva. Vedno večji nabor razvojnih orodij za razvoj različnih tipov mobilnih aplikacij odpira vprašanje glede njihove zmogljivosti. Vprašanje o zmogljivosti in ali je ta primerljiva med posameznimi pristopi razvoja, predvsem med domorodnimi in spletnimi različicami, je osrednja tema magistrskega dela. Razvoj spletnih mobilnih aplikacij trenutno predstavlja slabih 6 % celotnega trga mobilnih aplikacij, kar znaša 75.000 aplikacij od približno 1,25 milijona vseh aktivnih aplikacij, ki so dostopne preko različnih kanalov (Google Play, Apple Store, Windows Sotre in Blackberry) [60]. V naslednjih poglavjih bomo podrobneje predstavili prednosti in slabosti posameznih razvojnih pristopov ter prednosti in slabosti domorodnega in spletnega razvoja mobilnih aplikacij. Vsekakor pa želimo uvodoma izpostaviti dejstvo, da popularnost hitrega razvoja aplikacij narašča zaradi vedno večje potrebe po zadovoljevanju uporabniških zahtev in trendov. Prav ta konstantna zahteva po hitrem, cenovno ugodnem, pa vendar še vedno kakovostnem razvoju je pripeljala do vprašanja, ali so mobilne aplikacije, ki so razvite s pomočjo spletnih tehnologij, enako zmogljive kot njihove sorodne rešitve.

Čeprav 86 % uporabnikov uporablja zgolj domorodne in hibridne mobilne aplikacije, so vodilne in najbolj uporabljene predvsem domorodne [57]. Zato smo hibridne rešitve opustili. Velikani, kot sta Facebook in LinkedIn, so neuspešno uvedli hibridne rešitve, saj so se izkazale za nezadostne glede uporabniške izkušnje in zmogljivosti [57]. To je poglavitni razlog, zakaj smo se osredotočili na primerjavo domorodnih in spletnih mobilnih aplikacij. Zmogljivost mobilnih aplikacij je relevantna tako za razvijalce kot tudi za potrošnike in končne uporabnike.

Kot razvijalci se soočamo z vprašanjem, ki je povezano z razvojnim pristopom, tj., kaj omogoči enostavnejši in hitrejši razvoj, hkrati pa ne povzroči vidnih in občutnih pomanjkljivosti glede zmogljivosti. Zmogljivost se v splošnem definira kot učinkovitost sistema, da lahko obdela določeno količino podatkov v določenem času, pri tem pa poskrbi za optimizirano odzivnost in stabilnost [62]. Spletna zmogljivost, ki je tesneje povezana z mobilnimi aplikacijami, temelječih na spletnih tehnologijah, pa je povezana s hitrostjo prikazovanja vsebine oziroma nalaganja posameznih strani, kar posledično vpliva na lojalnost uporabnika, na njegovo zadovoljstvo in stopnjo zadržanosti [77]. Faktorji, ki vplivajo na zmogljivost aplikacije, so kompleksnost vsebine, hitrost samega izvajanja različnih funkcionalnosti, stabilnost in uporabljavnost virov [23]. V raziskavi, v kateri je bilo vključenih 3.500 razvijalcev mobilnih aplikacij, je bilo razvijalcem postavljeno vprašanje glede razvojnih pristopov, njihovih prednosti in slabosti. Rezultati raziskave so bili v prid mobilnim aplikacijam, ki temeljijo na spletnih tehnologijah [47]. Med vodilnimi pristopi so čiste spletne mobilne aplikacije (razvite s pomočjo označevalnega jezika HTML5, oblikovalskega jezika CSS3 in skriptnega jezika JavaScript) z vodilnimi 41 % ter hibridne aplikacije z 32 %. Kar 40 % razvijalcev v tej raziskavi je priznalo, da pogosto zamenjajo razvojne pristope (prehod iz domorodnega razvoja v spletni razvoj), saj so mobilne aplikacije, ki so razvite s pomočjo spletnih tehnologij, dovolj zmogljive, da izpolnijo osnovne in poslovne zahteve strank [47]. Čeprav tudi leta 2016 prevladuje skepticizem glede spletnih mobilnih aplikacij, je eksperiment primerjave med spletno HTML5 mobilno aplikacijo in domorodno Facebook aplikacijo pokazal presenetljive rezultate. V ogromno kriterijih se je HTML5 aplikacija namreč izkazala za boljšo glede zmogljivosti v primerjavi z domorodno različico [68]. Mobilne aplikacije, temelječe na spletnih tehnologijah, predstavljajo skupino mobilnih aplikacij, ki so zgrajene na tehnologijah, kot so označevalni jezik HTML, jezik za oblikovanje CSS in skriptni jezik JavaScript (jQuery). Medtem ko se za razvoj domorodnih mobilnih aplikacij uporabljajo predvsem domorodne rešitve (Android SDK, Xcode, Windows Phone SDK), lahko pri spletnih rešitvah zasledimo razvojna orodja v obliki platform (jQuery Mobile, Codiqa, Sencha Touch in Kendo UI) [59] oziroma vsa orodja, kjer lahko razvijamo aplikacije

s pomočjo spletnih tehnologij. Hibridne aplikacije predstavljajo nabor prepoznavnih orodij, kot so Apache Cordova, Adobe PhoneGap in Appcelerator Titanium [59]. Rezultati analize in napovedi organizacije Gartner za leto 2017 kažejo, da lahko pri 268 milijonih prenosov mobilnih aplikacij in pri generiranem dobičku v višini 77 milijard dolarjev [74] govorimo o začetku razcveta industrije mobilnih aplikacij, ki trenutno doživlja svoj vrhunec. Zato je še toliko pomembneje raziskati zmogljivosti posameznih rešitev razvojnih pristopov mobilnih aplikacij.

1.2 Namen in cilj magistrskega dela

Osnovni namen raziskave je usmerjen v analizo mobilnih aplikacij, temelječo na spletnih tehnologijah. Osredotočili se bomo na vprašanja povezana z zmogljivostjo domorodnih in spletnih mobilnih aplikacij. Z analizo želimo izpostaviti prednosti in slabosti domorodnega in spletnega pristopa razvoja. Rezultati analize so identificirani faktorji za določevanje zmogljivosti. Praktičen del magistrskega dela bo zajemala analiza zmogljivosti izbranih mobilnih aplikacij Facebook, Twitter, YouTube in Gmail v nadzorovanem okolju, kjer bomo izvedli eksperiment in preverjali indikatorje zmogljivosti na podlagi raziskovalnega modela.

Cilji v teoretičnem delu:

- Pregled relevantne literature omenjenega področja, kar vključuje predvsem pregled tujih strokovnih člankov s področja mobilnih aplikacij, mobilnega razvoja, domorodnega in spletnega razvoja ter pregled indikatorjev zmogljivosti;
- proučevanje različnih strokovnih analiz različnih avtorjev;
- predstavitev koncepta mobilnih aplikacij, njihovih razvojnih pristopov, razvojnih orodij ter prednosti in slabosti posameznega pristopa.

Cilji v raziskovalnem delu:

- Pridobiti odgovore na zastavljena raziskovalna vprašanja;
- analizirati zmogljivost mobilnih aplikacij razvitih s pomočjo spletnih tehnologij in izpostaviti njihove prednosti in slabosti;
- preveriti definirane raziskovalne hipoteze

Raziskovalna vprašanja, na katera želimo v okviru naloge odgovoriti:

- **RV 1:** Kako vpliva izbira razvojnega pristopa na zmogljivost končnega izdelka?
- **RV 2:** Kaj omogoča mobilnim aplikacijam, razvitim s pomočjo spletnih tehnologij, konkurenčni obstoj vzporedno z domorodnimi različicami?

- **RV 3:** Kakšni so faktorji zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah?
- **RV 4:** Ali izbira mobilne aplikacije, temelječe na spletnih tehnologijah, vpliva na uporabniško izkušnjo in zadovoljstvo?
- **RV 5:** Ali izbira mobilne aplikacije, temelječe na spletnih tehnologijah, vpliva na dojeto zmogljivost, enostavnost uporabe, kakovost in odzivnost?

Raziskovalne hipoteze, ki smo jih na osnovi raziskovalnih hipotez formulirali, so:

- **H1:** Mobilna aplikacija, razvita s pomočjo spletnih tehnologij, izkazuje enako enostavnost uporabe, kakovost, odzivnost in vizualno izdelavo kot njena domorodna različica.
- **H2:** Mobilna aplikacija, razvita s pomočjo spletnih tehnologij, pozitivno vpliva na zadovoljstvo uporabnikov v primerjavi z domorodno različico.
- **H3:** Mobilne aplikacije, temelječe na spletnih tehnologijah, z večjim številom zahtev v seji izkazujejo večji čas nalaganja, kar negativno vpliva na uporabniško izkušnjo.
- **H4:** Obstaja signifikantna razlika v zmogljivosti med testiranimi aplikacijami, temelječimi na spletnih tehnologijah.

1.3 Predpostavke in omejitve

Predpostavljamo, da so podatki in informacije iz literature in spletnih virov točni in dovolj obsežni ter da bomo s pomočjo anketnega vprašalnika, eksperimenta in analize le-teh dobili dovolj točne in ažurne podatke za uspešno potrjevanje ali zanikanje postavljenih raziskovalnih hipotez. Hkrati predpostavljamo, da bodo podatki, pridobljeni iz eksperimenta, preverljivi, legitimni in natančni.

V magistrskem delu se bomo omejili zgolj na dva razvojna pristopa, in sicer na domorodni in spletni način, ki ju bomo vključili tako v teoretični kot tudi v raziskovalni del. Pri raziskovalnem delu se bomo omejili na štiri aplikacije (Facebook, YouTube, GMail, Twitter) ter na odprto-kodno orodje za testiranje in analiziranje podatkov zaradi finančnih omejitev, sodelovali pa bodo lahko moški in ženske stari 18 – 55+ let. Predhodno znanje ali seznanjenost z mobilnimi aplikacijami ni potrebno. V sklopu raziskave in končnih rezultatov ne bomo upoštevali neveljavnih ali nepopolnih podatkov.

1.4 Kratak opis celotnega dela

Prvo poglavje prikazuje trenutno stanje domorodnih, spletnih in hibridnih mobilnih aplikacij na trgu v razmerju z njihovo popularnostjo, uporabnostjo in razširjenostjo. Izpostavimo dejstvo, da so zahteve končnih uporabnikov različne glede na izbrano področje, kar vpliva na izbiro razvojnega pristopa zaradi nabora privzetih funkcionalnosti, ki jih pristop omogoča. V ospredje postavimo spletne tehnologije, specifično označevalni jezik HTML5, saj zaradi svojega obširnega nabora funkcionalnosti in hitre izdelave pripomore k priljubljenosti razvojnega pristopa spletnih mobilnih aplikacij (kot napoveduje organizacija Gartner).

V drugem poglavju predstavimo domorodne mobilne aplikacije ter izpostavimo ključna orodja in komponente za njihov razvoj, kot sta SDK in API vmesnike, ter se dotaknemo materialnega oblikovanja. Med razvojnimi orodji izpostavimo le najbolj priljubljena, in sicer Android SDK, Xcode in Windows Phone SDK. Na koncu poglavja predstavimo prednosti in slabosti domorodnih mobilnih aplikacij, ki lahko vplivajo na zmogljivost ter uporabniško zadovoljstvo in izkušnjo.

V tretjem poglavju se srečamo z mobilnimi aplikacijami, temelječimi na spletnih tehnologijah. Razložimo pojem odzivnih spletnih strani in izpostavimo razliko med spletno mobilno aplikacijo in odzivnimi spletnimi stranmi. Predstavimo trenutno prevladujoča razvojna orodja ter posamezne spletne tehnologije – označevalni jezik HTML5, oblikovalni jezik CSS3 in skriptni jezik JavaScript. Delovanje tovrstnih aplikacij poteka v brskalniku, zaradi česar je pomembno tudi prikazati delovanje vmesnika WebKit. Na koncu predstavimo še prednosti in slabosti mobilnih aplikacij, ki so razvite s pomočjo spletnih tehnologij.

Sledi četrto poglavje, kjer predstavimo opravljeno raziskavo in analizo empiričnih podatkov. Podrobneje opišemo omejitve eksperimenta in njegovo izvedbo ter postopek analiziranja pridobljenih podatkov. Predstavimo pomen HAR datotek in Google DevTools orodja za analiziranje spletne zmogljivosti ter postopke analize posameznih mobilnih aplikacij (Twitter, YouTube, Facebook in GMail). Za konec potrdimo oziroma ovržemo postavljene hipoteze.

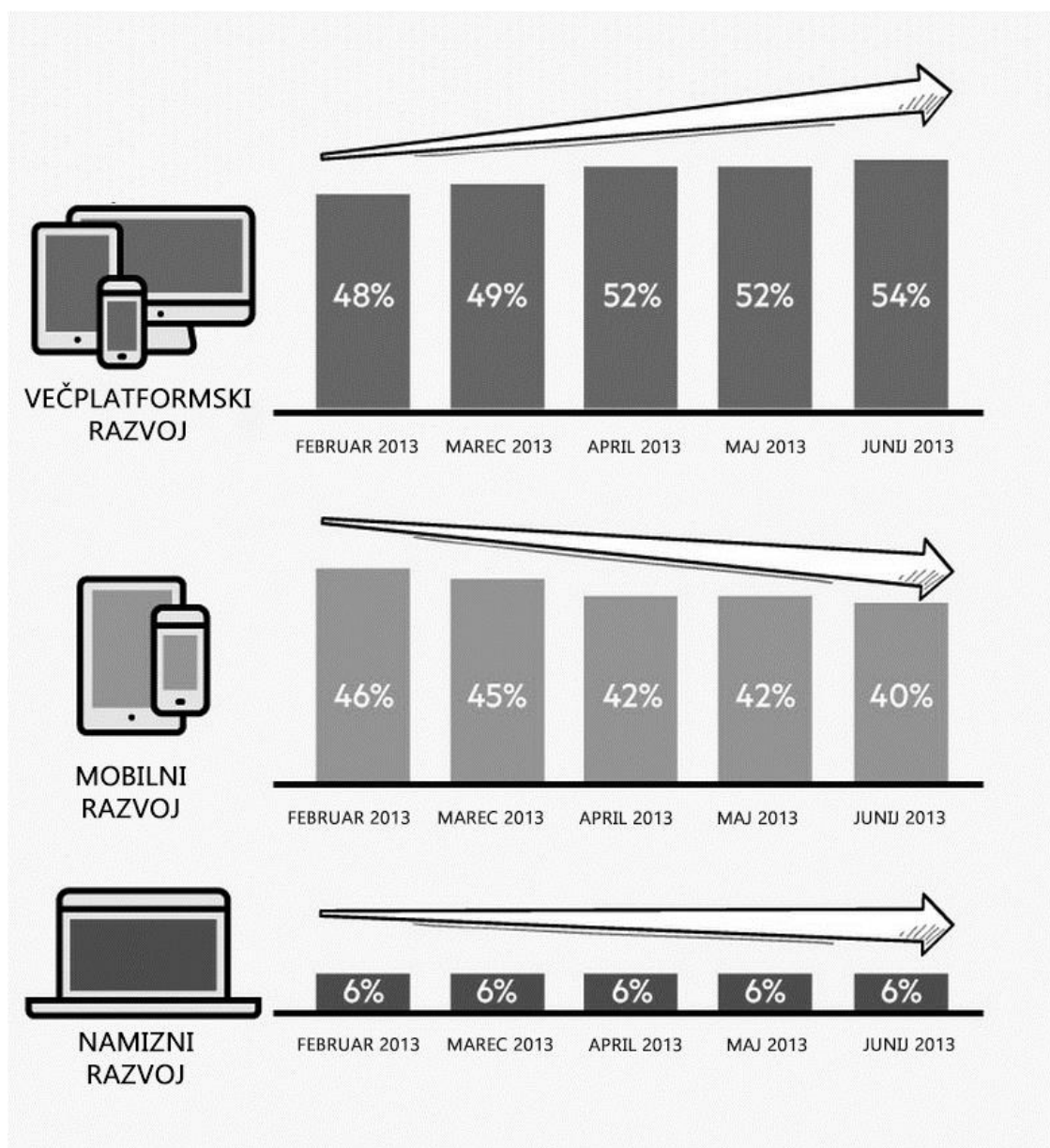
Peto oziroma zadnje poglavje predstavlja sklep, v katerem podamo objektivno oceno rezultatov ter jih povežemo s problemom, ki smo ga postavili v uvodu. Nakažemo morebitne težave in opažanja ter podamo napotke za nadaljnjo raziskovalno delo.

2 STANJE MOBILNIH APLIKACIJ DANES

Glavni del magistrske naloge je odgovarjanje na vprašanje, ali so mobilne aplikacije, temelječe na spletnih tehnologijah, enako zmogljive kot domorodne različice. Prihodnost leži v aplikacijah, ki podpirajo razvoj tako za mobilne aparate, tablice kot tudi osebne računalnike in pametne televizorje. Razvoj je usmerjen predvsem v večplatformske aplikacije. Medplatformski in večplatformski razvoj sta sinonima, s katerima lahko označujemo dvoje, in sicer razvoj rešitve preko različnih tipov naprav (mobilnik, tablica, televizor) ali razvoj rešitve preko različnih platform (Android, Apple, Blueberry, Windows). Raziskava, opravljena pri ComScore [69] iz leta 2013, nakazuje upad in vzpon razvojnih pristopov. Uporaba zgolj računalnikov je začela upadati, saj se je povečala uporaba mobilnih aplikacij.

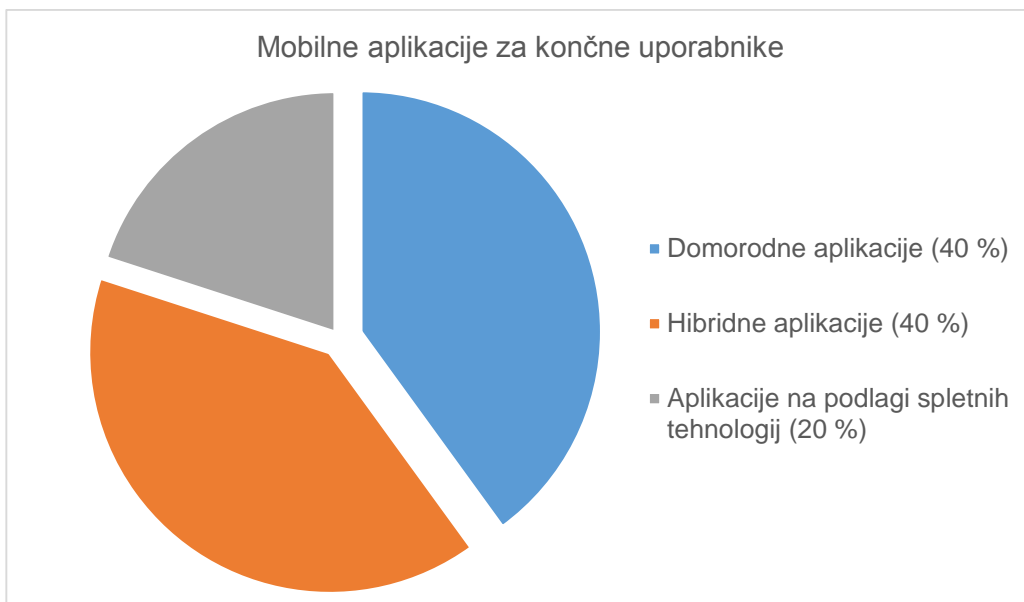
Slika 2.1 potrjuje, da prihodnost leži v medplatformskem razvoju, mobilni razvoj upada, namizni razvoj pa ne kaže padcev ali vzponov v razvoju. 52 % anketiranih razvijalcev soglaša z razvojem označevalnega jezika HTML5 pri razvoju aplikacij, saj podpira medplatformski razvoj rešitev [69]. Med anketiranimi razvijalci jih 52 % soglaša, da uporaba označevalnega jezika HTML5 – torej uporaba spletnih tehnologij – pripomore k učinkovitejšemu razvoju mobilnih aplikacij za različne platforme zaradi obsežnosti podprtih funkcionalnosti, ki vplivajo na zmogljivost rešitve. Drugi so odgovorili, da bi razvoj za različne platforme lahko razrešili s pomočjo razvoja domorodnih aplikacij in jih šele nato razširili na druge platforme (23 %).

Sledijo razvijalci, ki bi razvijali različne rešitve vzporedno (21 %), ter tisti (4 %), ki se strinjajo, da se primarno izbere eno platformo, za katero se ekskluzivno razvija rešitev [69]. Iz enake raziskave pridobimo tudi podatke, ki potrjujejo, da se vedno več razvijalcev (kar 65 % vseh anketirancev) odloča za razvoj mobilnih aplikacij, ki temeljijo na spletnih tehnologijah. Razvijalci so prišli do spoznanja, da so hibridne aplikacije za 30–90 % cenejše v razvoju v primerjavi z domorodnimi aplikacijami [69].

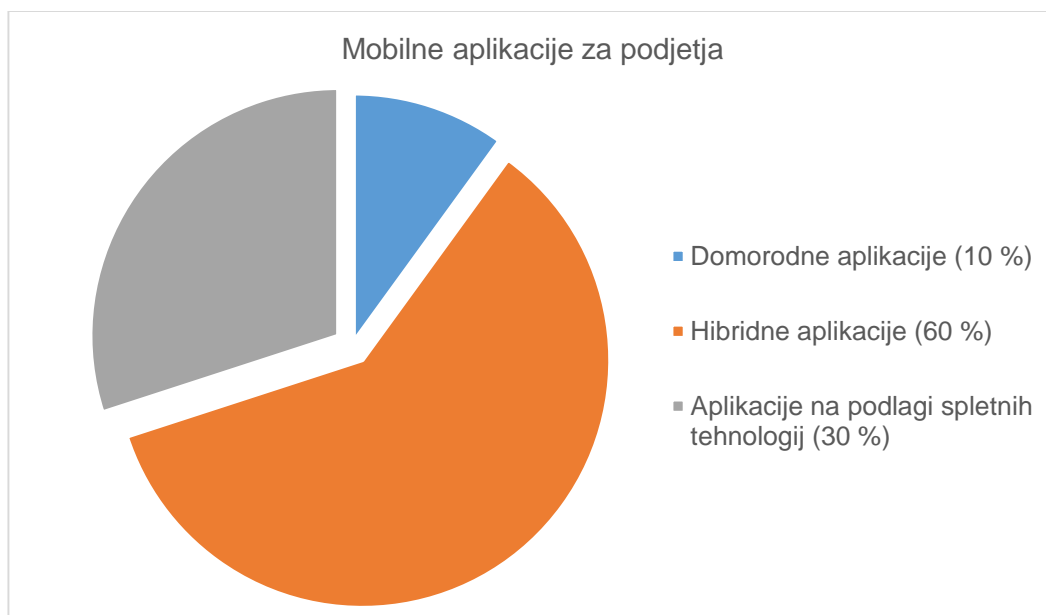


Slika 2.1: Statistična slika razvoja aplikacij za različne platforme [69]

Organizacija Gartner predvideva, da bomo leta 2016 imeli več kot 50 % mobilnih aplikacij, ki bodo zgrajene s pomočjo hibridnega razvojnega načina [73]. Statistično gledano bodo prevladovali spletne in hibridne aplikacije v primerjavi z domorodnimi še posebej, kadar govorimo o razvoju aplikacij za podjetja. Tako je predvsem zaradi enostavnega dostopa spletnih tehnologij (označevalni jezik HTML, oblikovalski jezik CSS in skriptni jezik JavaScript) ter prednosti cenovno ugodnejšega in hitrejšega razvoja v primerjavi z domorodnimi mobilnimi aplikacijami.



Slika 2.2: Napoved razvojnih pristopov za končne uporabnike v letu 2016 [69]



Slika 2.3: Napoved razvojnih pristopov za podjetja v letu 2016 [69]

Slika 2.2 in Slika 2.3 podrobneje prikazujeta napoved mobilnih aplikacij in njihovih smernic, pri čemer sta pri končnem uporabniku konkurenčna razvojna pristopa hibridni in domorodni, pri podjetjih pa hibridni in spletni razvoj. Do teh razlik pride predvsem zaradi hitrejšega in cenovno ugodnejšega razvoja pri spletnih aplikacijah. Priljubljenost domorodnih aplikacij narašča, kar je razvidno iz analize iz leta 2014 [66], kjer je bilo preko trgovine Apple Store prenesenih 75 milijard mobilnih aplikacij. Med letoma 2009 in 2017 obstaja projekcija, ki

zagotavlja povečanje števila prenešenih mobilnih aplikacij na svetovni ravni na 102.062 milijonov [66]. Samo v letu 2017 pričakujemo projicirano število prenesenih mobilnih aplikacij v višini 268.692 milijonov. Dejstvo je, da se v tej industriji obrne ogromno denarja, kar se kaže kot povratni dobiček, ki znaša trenutno 34,99 milijard ameriških dolarjev [66]. Med bolj priljubljene trgovine mobilnih aplikacij prištevamo Google Play, Windows Phone Store in Apple App. Ta razporeditev izhaja iz analize [66], kjer se je Google Play uvrstil na prvo mesto z 1,3 milijona mobilnih aplikacij, Apple App na drugo mesto in Windows Phone Store na tretje mesto z le 300.000 mobilnimi aplikacijami. Med nastajanjem magistrskega dela je Google Play trgovina v vodstvu s skoraj 1,5 milijona mobilnih aplikacij, medtem ko lahko v Apple App trgovini najdemo le 1,25 milijona mobilnih aplikacij [6]. Število mobilnih aplikacij ni edini podatek, ki ga lahko izpostavimo. Pomembna je tudi priljubljenost posameznih mobilnih aplikacij. Število unikatnih obiskovalcev na omrežju Facebook z uporabo mobilnih aplikacij iz iOS platforme znaša samo v juniju 2014 47,25 milijona [66]. Ob tem imamo še podatek, kot je število unikatnih ameriških obiskovalcev v Google Play trgovini v mesecu juniju 2014, ki znaša 72,25 milijona [66]. Priljubljenost Google Playa narašča tudi v letu 2015 še posebej zato, ker narašča priljubljenost Android operacijskega sistema zaradi njegove razširjenosti med končnimi uporabniki in napravami.

Popularnost mobilnih aplikacij iz leta v leto narašča. Kendo UI, oddelek podjetja Telerik, je leta 2013 izvedel raziskavo, v katero je bilo vključenih 5000 razvijalcev iz različno velikih podjetij in z različnih pozicij. Glavno vprašanje raziskave je bilo, ali se označevalni jezik HTML5 in sorodne spletne tehnologije uporabljajo v mobilnem razvoju ter zakaj. Izpostavi se tudi uspešnost tovrstnih spletnih mobilnih aplikacij. Med anketiranci je bilo 72 % razvijalcev, preostali del so bili različni tehnični delavci [7]. Povzetek rezultatov ankete:

- 50 % vseh anketirancev je priznalo, da so v letu 2012 uporabili spletne tehnologije (označevalni jezik HTML5) za razvoj različnih aplikacij. Vsaj 9 od 10 načrtuje uporabo spletnih tehnologij za nadaljnje razvoje [7].
- Izmed vseh razvijalcev jih kar 41 % uporablja kombinacijo spletnih tehnologij (označevalni jezik HTML5 in skriptni jezik JavaScript) za razvoj mobilnih aplikacij, medtem ko jih samo 36 % uporablja striktno domorodni pristop [7].
- Samo 15 % vseh anketirancev je priznalo, da striktno razvija domorodne mobilne aplikacije [7].
- Velikost podjetja nima direktnega vpliva na izbiro razvojnega pristopa. Pri tem se v 36 % primerov uporablja pristop razvoja s pomočjo spletnih tehnologij ter v 32 % primerih hibridni pristop. Ostali del se je strinjal, da je najboljši pristop tisti, kjer se vključujejo različni pristopi glede na zahteve mobilnih aplikacij in končnih uporabnikov [7].

- Glede uporabe označevalnega jezika HTML5 v prihodnosti se kar 87 % vseh anketirancev strinja, da se bo kot primarni način razvoja mobilnih in spletnih aplikacij uporabljal ta pristop, 53 % pa, da bo osredotočenost na razvoju mobilnih spletnih strani – razvoj odzivnih spletnih strani (RWD) [7].

Raziskave, kot je raziskava podjetja Telerink [7], podprejo teorijo, da so spletne mobilne aplikacije enako zmogljive kot sorodne rešitve ter da se v podjetjih vključuje tudi razvojni pristop. Popularnost mobilnih aplikacij, ki so jih razvili s pomočjo spletnih tehnologij, je začela naraščati okrog leta 2012 in je v letu 2015 dosegla svoj vrhunec. Organizacija Gartner je v letu 2014 identificirala 10 mobilnih tehnologij in sprememb za razvojno leto 2015 in 2016 [72]. Med omenjenimi mobilnimi tehnologijami je označevalni jezik HTML5, kjer so izpostavili predvsem jezikovno nezrelost in fragmentacijo, ki je prevladovala v letu 2014 in posledično vplivala na neuporabo razvojnega pristopa. Napoved pojasnjuje še, da bo popularnost spletnih jezikov, kot je HTML5, narasla šele takrat, ko bo označevalni jezik dozorel in s tem omogočil večjo varnost, prenosljivost in zmogljivost [72]. Organizacija Gartner oz. njihova napoved izpostavlja pomembnost spletne tehnologije HTML5 kot vodilne sile med mobilnimi tehnologijami tako za končne uporabnike kot tudi za podjetja, saj v primerjavi z drugimi jeziki podpira medplatformski razvoj [72]. Medplatformski razvoj pomeni, da se aplikacija sicer razvije enkrat, vendar se lahko izvaja na vseh prevladujočih platformah – Android, Apple, Windows in Blackberry. Srečamo se tudi z izrazom »3x3« razvoj, kar pomeni, da se mobilna aplikacija hkrati razvije za Android, Apple in Windows ter je popolnoma prilagojena njihovi infrastrukturi (posledično ponudi ta pristop podporo trinivojski arhitekturi). Trinivojska arhitektura je arhitektura, ki je sestavljena iz treh osnovnih slojev, in sicer iz predstavitvenega, aplikacijskega in podatkovnega sloja [39]. Lahko povzamemo, da razvoj mobilnih aplikacij teži k spletnim in domorodnim rešitvam, s tem da vsak pristop prinaša svoje prednosti in slabosti tako pri podjetjih kot tudi pri končnem uporabniku.

V magistrskem delu se želimo osredotočiti predvsem na primerjavo med domorodnimi in spletnimi mobilnimi aplikacijami. Zato bomo v naslednjih poglavjih podrobneje predstavili njihove principe, prednosti in slabosti ter razvojna orodja.

3 DOMORODNE MOBILNE APLIKACIJE

Domorodne mobilne aplikacije so aplikacije specifično razvite za določene naprave in njihove domorodne platforme [50]. V primerjavi z mobilnimi aplikacijami, ki so razvite s pomočjo spletnih tehnologij, do katerih se enostavno dostopa preko brskalnika, je zanje treba prenesti in inštalirati domorodne različice. Domorodno mobilno aplikacijo prenesemo iz aplikacijskih trgovin, kot so Google Play, Apple Store in Windows Store. Najpogosteje uporabljeni programski jeziki za ustvarjanje domorodnih mobilnih aplikacij so Java, C# in Objektni C ter C++. Domorodne mobilne aplikacije so v bistvu binarne izvršilne datoteke, ki jih prenesemo iz različnih aplikacijskih trgovin neposredno na naš mobilnik ter jih lokalno tudi shranimo, namestimo in izvajamo [34]. Do nameščene domorodne aplikacije dostopamo preko bližnjice na mobilniku. Ob inicializaciji domorodna mobilna aplikacija direktno vzpostavi komunikacijski dialog z operacijskih sistemov na napravi brez dodatnih vmesnikov. Ta komunikacija je lahko vzpostavljena med domorodno aplikacijo in operacijskim sistemom s pomočjo API vmesnikov, čeprav je izbor le-teh močno odvisen od izbranega operacijskega sistema [34]. Načeloma sta domorodnim aplikacijam omogočena dostop in uporaba vseh API vmesnikov operacijskega sistema, kar močno vpliva na njihovo zmogljivost v primerjavi s spletnimi aplikacijami, kjer smo glede uporabe API vmesnikov omejeni.

Razvoj domorodnih aplikacij poteka tako, da najprej sprogramiramo kodo s pomočjo izbranega programskega jezika (Java, C#, C++ ali Objektni C) – to kodo imenujemo izvorna koda. Pri razvoju si pomagamo z različnimi razvojnimi orodji, s pomočjo katerih izvorno kodo preoblikujemo in predelamo v izvršilno obliko [34]. Razvoj domorodnih aplikacij ostaja med osrednjimi razvojnimi pristopi, kar potrjuje tudi analiza podjetja Flurry Analytics [25], v okviru katere so prikazali, da je uporaba mobilnih aplikacij narasla z 80 % na 86 %, medtem ko je uporaba spleta in spletnih rešitev upadla z 20 % na 14 %. Razvojna orodja za domorodne aplikacije imenujemo SDK oziroma Software Development Kit.

3.1 Razvoj s pomočjo SDK in API

Vsaka mobilna platforma ima svoj SDK, ki predstavlja domorodno razvojno ogrodje za ustvarjanje mobilnih aplikacij izbrane platforme. Čeprav je način razvoja med posameznimi platformami dokaj podoben, imajo te vseeno specifična ogrodja za razvoj mobilnih aplikacij ter specifične smernice in pravila. Izpostavili smo operacijske sisteme iOS, Android, Blackberry OS in Windows, ki smo jih med seboj primerjali v Tabela 3.1. v sledečih kategorijah: programski jezik, ime ogrodja, format končne aplikacije ter ime aplikacijske trgovine, kjer lahko končni produkt prodajajo oziroma ga lahko tam snamemo [34].

Tabela 3.1: Primerjava razvojnih orodij za domorodne mobilne aplikacije [34]

	Apple iOS	Android	Blackberry OS	Windows Phone
Programski jezik	Objektni C, C, C++	Java (občasno C, C++)	Java	C#, VB.NET, C++
Ogrodje	Xcode	Android SDK	BB Java Eclipse vtičnik	Visual Studio, Windows Phone SDK
Format	.app	.apk	.cod	.xap
Trgovina	Apple App Store (iTunes)	Play Store	Blackberry App World	Windows Phone Marketplace

Razčlenjenost razvojnih orodij in programerskih jezikov prinaša vodilno slabost domorodnih mobilnih aplikacij, namreč programska koda je napisana v enem okolju, a je neizvedljiva v drugem. Takšen razvoj in princip pripomoreta, da traja razvoj dlje in predstavlja velik denarni zalogaj ter zahteva večjo skupino razvijalcev, da se pokrijejo posamezni operacijski sistemi. Vsak operacijski sistem ima tudi unikatni zalogaj API vmesnikov. Čeprav stane razvoj domorodnih aplikacij v povprečju več kot razvoj hibridnih in spletnih, se še vedno uporablja zaradi nabora API vmesnikov, do katerih spletne in hibridne rešitve nimajo popolnega dostopa. API vmesniki skrbijo za komunikacijo z funkcionalnostmi, kot so pospeševalnik, kamera, koledar, budilka, opozorila, SMS sporočila in več. Delimo jih na visokonivojske in nizkonivojske API vmesnike. Trenutno obstaja 13.344 različnih kategoriziranih API vmesnikov, ki jih lahko uporabljamo v domorodnih aplikacijah; mednje prištevamo

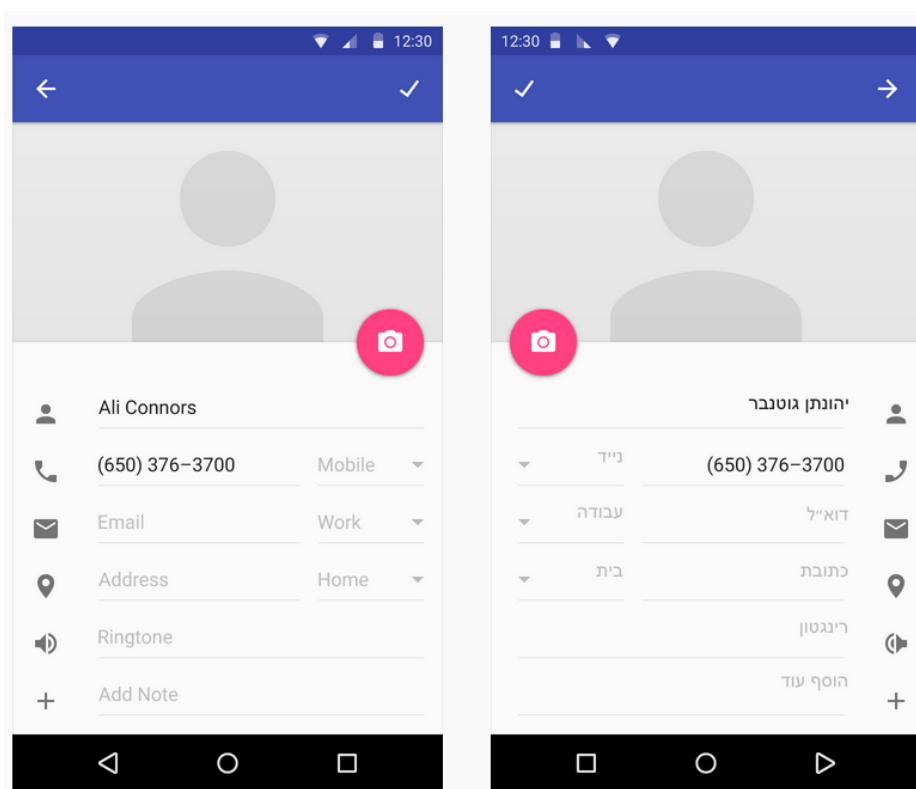
kategorije, kot so vmesniki za mapiranje, družbeni vmesniki, iskalni vmesniki, vmesniki za sporočanje, slike, finance, potovanja in delov v oblaku [56].

Poznamo nizkonivojske vmesnike (omogočajo direktno komunikacijo z zasloni na dotik ali s tipkovnico, prikazovanje grafičnih elementov ter povezavo s spletom in razno komunikacijo) in visokonivojske vmesnike (vključujejo vse funkcionalnosti nizkonivojskih vmesnikov z dostopom do višjenivojskih funkcionalnosti, ki zahtevajo več pooblastil) [34].

Vmesniki skrbijo za komunikacijo med strojno in programsko opremo operacijskega sistema. Njihova funkcionalnost se izkaže pri uporabi aplikacij, kot so Dropbox (komuniciranje in prenos podatkov v oblak), Facebook (dvosmerna komunikacija med uporabniki) ter razne notifikacije aplikacij. Vsak operacijski sistem ima svoje smernice glede grafičnih vmesnikov. Grafične uporabniške vmesnike imenujemo tudi GUI. Navodila za grafične uporabniške vmesnike vsebujejo definicije za različne komponente, ki predstavljajo osrednji del mobilnih aplikacij: razporeditev elementov, videz različnih akcijskih gumbov, vnosnih polj, menijev in navigacij, dialogov, iskalnikov ter definicije posameznih stilov in tem [36] [5] [53]. Domorodne mobilne aplikacije, ki sledijo tem grafičnim smernicam, zagotovijo večjo uporabniško zadovoljstvo in izkušnjo. Rezultat specifičnih grafičnih smernic postanejo posamezne aplikacije, ki so unikatne na operacijski sistem ter omogočajo uporabniku, da jih takoj prepozna.

Med bolj popularne smernice prištevamo materialno obliko od Googla [4]. Lahko jih tudi kategoriziramo kot animacije, stile, razporeditve, komponente, vzorce in vire. Velik poudarek je na uporabniški prijaznosti in uporabnosti. Google je prepričan, da se izboljšata tudi uporabniška izkušnja in zadovoljstvo, če se poenotijo smernice za mobilne aplikacije, saj je uporabnik bolj seznanjen z aplikacijo in njenim delovanjem. Aplikacija je uporabna, kadar lahko končni uporabnik navigira skozi aplikacijo brez predhodne uporabe in razume delovanje le-te [31]. Pomembnost navigacije je velika, saj omogoča končnemu uporabniku, da se lažje znajde, da so informacije dostopne na principu dotika in da se med preklopom zaslonov ne izgubijo podatki ter uporabnik. Drugi faktor je berljivost. Za slabovidne ali starejše uporabnike moramo zagotoviti brezhibno delovanje tudi pri uporabi večje pisave, močnejšega kontrasta in z manj vpadljivimi barvami za barvno slepe. Uporabnost aplikacije poveča tudi bidirektivnost. Bidirektivnost se definira kot funkcionalnost aplikacije, s pomočjo katere lahko prikažemo lokalizirano vsebino v angleškem ter maternem jeziku končnega uporabnika [31]. Android razrešuje bidirektivnost s tako imenovanimi bidirektivnimi skriptami. Bidirektivnost ne vpliva samo na sam kontekst in postavitev komponent, ampak tudi na uporabniške elemente in ikonografijo. Slika 3.1 prikazuje bidirektivnost, kjer desna stran predstavlja aplikacijo za angleške končne uporabnika in leva stran aplikacijo,

lokalizirano za arabske končne uporabnike. Vizualni vtis aplikacije kot tudi njena uporabnost se ne spremenita, prilagodi se le princip uporabe aplikacije za izbranega uporabnika.



Slika 3.1: Pomembnost bidirektivnosti v mobilnih aplikacijah [31]

Ideja Google rešitve je omogočiti razvoj domorodnih mobilnih aplikacij, ki ne zahtevajo predhodne izkušnje za razumevanje delovanja aplikacije. Poenotenje grafičnih smernic pozitivno vpliva na uporabniško izkušnjo in na uporabnost mobilne aplikacije, saj je Google učinkovitost materialnega oblikovanja dokazal z razsežnostjo svojih izdelkov in z njihovo popularnostjo (Google Docs, Google Forms, Google Search).

3.2 Razvojna orodja

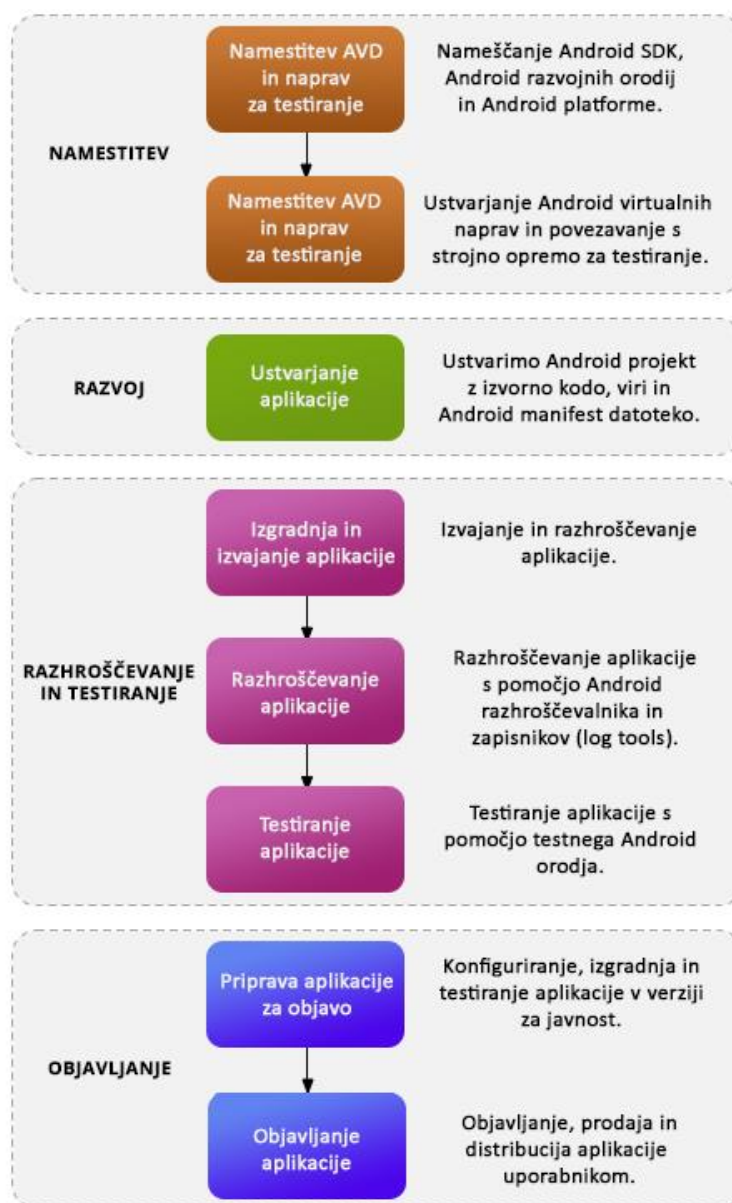
V poglavju razvojna orodja za domorodne mobilne aplikacije bomo predstavili tri glavna orodja, in sicer Android SDK, Xcode in Windows Phone SDK. Predstavili bomo vsak pristop posebej, funkcionalnosti in osnovni nabor orodij, ki jih SDK orodja prinašajo.

3.2.1 Android SDK

Android SDK je orodje, s katerim lahko razvijamo domorodne mobilne aplikacije za Android operacijski sistem. Glavni faktorji razvojnega orodja so inteligentni urejevalnik kode, možnost integracije kodnih delov z javnega repozitorija GitHub, razvoj aplikacij na več ekranih in emulator mobilnih kot tudi na ostalih napravah. V zadnjem letu so Android SDK orodju dodali tudi novosti, kot sta fleksibilen Gradle (sistem za avtomatizacijo testiranja in izgradnje aplikacij) ter ProGuard (dodana varnost aplikacijam).

Razvojna orodja so orodja, s katerimi lahko razvijemo in testiramo mobilne aplikacije. Celoten proces izgradnje domorodne Android mobilne aplikacije je prikazan tudi na Slika 3.2 [3]:

- **Postavitev razvojnega okolja:** faza, kjer vzpostavimo razvojno okolje ter ustvarimo Android virtualne naprave (AVD) in ju povežemo z razvojnim okoljem naprave, na kateri želimo testirati in uporabljati aplikacijo.
- **Vzpostavitev projekta in razvoj:** faza, kjer ustvarimo nov projekt znotraj razvojnega okolja in izberemo aplikacijske module, ki vsebujejo izvorno kodo naše aplikacije.
- **Razvijanje, preverjanje in testiranje aplikacije:** v tej fazi ustvarimo datoteko s končnico .apk, ki jo lahko sedaj inštaliramo tudi na naši testirani napravi, ter testiramo obnašanje aplikacije v realnem okolju v realnem času. Faza razvoja in preverjanja je zelo pomembna, saj služi razvoju temeljnih funkcionalnosti ter preverjanju in testiranju vseh funkcionalnosti in pričakovanega obnašanja aplikacije.
- **Objavljanje aplikacije:** faza, v kateri konfiguriramo mobilno aplikacijo ter poskrbimo, da je pripravljena za objavo v Google Play Store ter da odgovarja vsem podanim grafičnim ter funkcionalnim smernicam izbrane platforme.



Slika 3.2: Faze razvoja domorodne Android mobilne aplikacije [3]

3.2.2 Xcode

Razvoj Apple domorodnih aplikacij poteka s pomočjo Xcode razvojnega ogrodja. Za razvoj tovrstnih aplikacij moramo izpolniti tri zadeve. Potrebujemo Mac računalnik z operacijskim sistemom OS X ali starejšo verzijo, Xcode programsko ogrodje ter iOS SDK [35]. Razvoj domorodnih mobilnih aplikacij je podoben, neodvisno od izbranega operacijskega sistema. Pri strukturiranju mobilne aplikacije je pomembno, da definiramo koncept, oblikujemo uporabniški vmesnik ter definiramo vse interakcije, do katerih lahko pride med končnim

uporabnikom in našo mobilno aplikacijo. Kadar želimo definirati koncept za mobilno aplikacijo, je treba odgovoriti na vprašanja:

- **Kakšen je namen aplikacije?** Mobilna aplikacija potrebuje jasno definirane cilje in namen.
- **Kateri problem želimo rešiti s pomočjo aplikacije?** Odlične mobilne aplikacije vedno razrešujejo jasno definirane probleme z jasno definiranimi cilji in definicijami.
- **Katero vsebino želimo vključiti v aplikacijo?** Z raziskavo je dokazano, kako pomembna je vsebina, kadar razvijamo mobilne aplikacije neodvisno od izbrane platforme. Izbira vsebine vpliva na oblikovanje uporabniškega vmesnika ter na način prikazovanja podatkov končnemu uporabniku [35].

Rdeča nit, ki povezuje aktivnosti mobilnih aplikacij med seboj, se pri Android rešitvah imenuje »delovni tok« (ang. workflow), pri Apple rešitvah pa »zgodbe« (ang. storyboards). Do razlik med načinom izvajanja aktivnosti prihaja zaradi razlik v operacijskih sistemih samih. Na Slika 3.3 je prikazan način izdelave Apple domorodne aplikacije. V primerjavi z Android domorodno aplikacijo (Slika 3.2) na prvi pogled opazimo, da si aktivnosti izgradnje Android aplikacije sledijo linearno, medtem ko so pri Apple aplikaciji paralelne aktivnosti. Vendar je princip enak: uporabnik se registrira, naloži razvojno orodje, paralelno ustvarja aplikacijo ter dokumentacijo, uporabniški vmesnik in razne tehnologije. Procesu sledi testiranje oziroma razhroščevanje in če je to uspešno, se lahko aplikacija objavi v Apple trgovini.



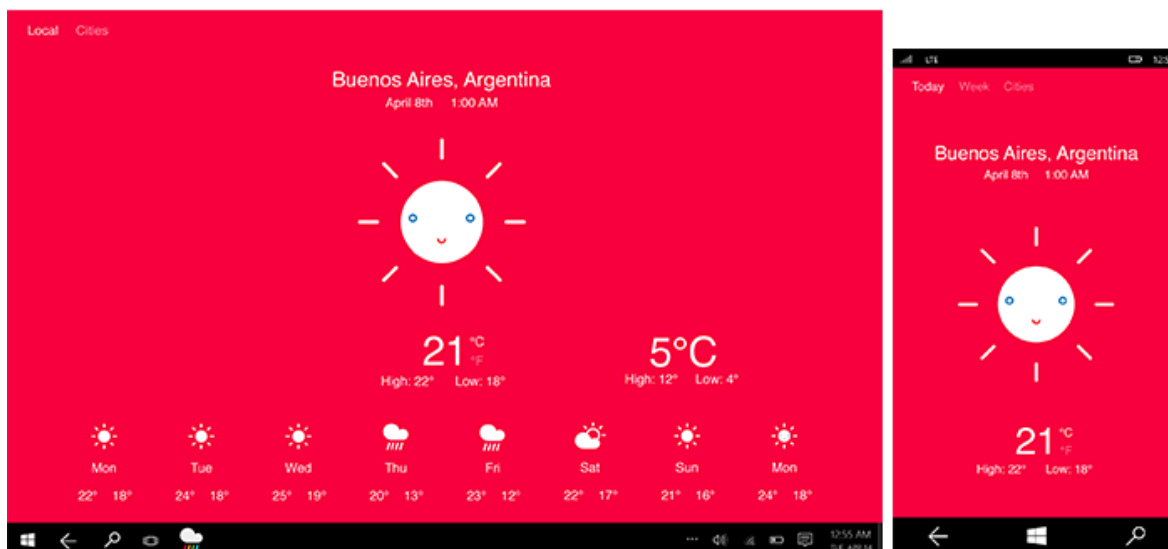
Slika 3.3: Razvoj domorodne aplikacije za Apple platformo [35]

Ali se razvojna orodja razlikujejo po funkcionalnosti? Ne. Razvojna orodja ponujajo enak nabor funkcionalnosti. Edina razlika so grafične smernice za razvoj uporabniškega vmesnika ter nabor dodatnih orodij za testiranje aplikacije in preverjanje kode – avtomatizacija. Do razlik prihaja tudi v načinu verzioniranja aplikacij in naših projektov.

3.2.3 Windows Phone SDK

Windows je prevzel hibridni princip razvoja mobilnih aplikacij, saj domorodne Windows Phone aplikacije kombinira z označevalnim jezikom HTML5, oblikovalskim jezikom CSS3 in celo skriptnim jezikom JavaScript. Primarno se uporabljajo programski jeziki, kot so C#, C++/C in XAML. Razvojno orodje tovrstnih aplikacij je Visual Studio, kjer se nahajajo tudi dodatna orodja za testiranje, preverjanje kode in oblikovanje uporabniškega vmesnika. V Visual Studio orodju pričnemo z razvojem domorodne Windows aplikacije tako, da ustvarimo nov projekt, ki je prilagojen za aplikacije. V primerjavi z Android in Apple ima Windows svoje grafične smernice, ki jih imenuje kar UWP oziroma Universal Windows Platform smernice. Te smernice omogočajo paralelni razvoj za vse Windows naprave, ki imajo nameščen Windows 10 operacijski sistem. Tako kot pri Android in Apple aplikacijah podaja tudi Windows osnovne grafične smernice, ki vsebujejo vse od osnov, navigacije kot tudi napotke za izdelavo uporabniškega vmesnika [37].

V primerjavi z Android in Apple razvojem se v programski kodi pojavijo tudi spletne tehnologije, ki pripomorejo, da se lahko aplikacije avtomatično prilagajajo napravi (Slika 3.4). UWP smernice med drugim določijo tudi pravila glede oblikovalskega jezika CSS3, s pomočjo katerega se avtomatično prilagodita razporeditev elementov ter resolucija naprave.



Slika 3.4: Prilagajanje Windows aplikacij na različnih napravah [37]

Princip razvoja Windows domorodnih aplikacij je zagotavljanje delovanja med različnimi napravami, dokler te naprave delujejo na operacijskem sistemu Windows 10. Prednost takšnega razvoja je, da lahko uporabniški vmesnik razvijemo le enkrat ter ga avtomatično sprogramiramo, da se samodejno prilagodi napravi [37]. Druga prednost je, da kot razvijalec nismo prisiljeni v večplatformski razvoj, temveč lahko selektivno izberemo, katere so naše ciljne naprave (mobilniki, tablice, prenosniki računalniki, osebni računalniki ali televizijski sprejemniki) [37].

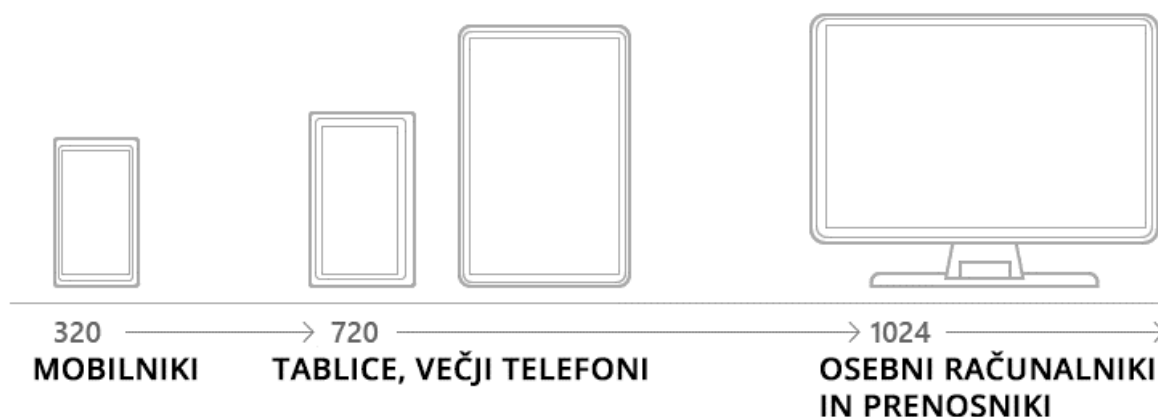
Odzivno oblikovanje Windows domorodnih aplikacij nosi enake vrednote kot odzivno oblikovanje spletnih strani – princip, s katerim se pogosto srečamo tudi pri spletnih mobilnih aplikacijah. Faktorji, ki vplivajo na odzivno oblikovanje, so sprotno spreminjanje velikosti (razporeditev UI-elementov popolnoma glede na velikost resolucije naprave), prelivanje (spreminjanje toka UI-elementov za optimizirano prikazovanje vsebine), razkrivanje (razkrijemo samo relevantne UI-elemente glede na resolucijo naprave), preoblikovanje (prerazporeditev arhitekture same aplikacije) in nadomeščanje (končni uporabnik lahko določi resolucijo in orientacijo, na kateri želi pregledovati informacije iz aplikacije) [38].

Tehniko nadomeščanja vidimo na Slika 3.5. Princip je povzet po praksi oblikovanja spletnih strani po mrežnem sistemu, kjer lahko s pomočjo posameznih definiranih stolpcev ustvarimo prelivanje in nadomeščanje vsebine. Tehnika nadomeščanja stori ravno to, kjer poskrbi za prikazovanje UI-elementov glede na podano resolucijo in orientacijo končne naprave.



Slika 3.5: Tehnika nadomeščanja UI-elementov [37]

Velik del odzivnega oblikovanja leži predvsem v prelomnih točkah. Kot prikazuje Slika 3.6, se v praksi razvoja Windows mobilnih aplikacij definirajo tri prelomne točke: 320 pikslov, 720 pikslov in 1024 pikslov. Prevladujoča misel v razvoju Windows aplikacij je, da razvijemo aplikacijo enkrat, uporabljamo pa jo lahko na različnih napravah. Zato moramo biti skrbni pri zasnovi aplikacij, da bodo informacije in podatki enako dosegljivi in berljivi ne glede na izbrano napravo in njeno resolucijo.



Slika 3.6: Prelomne točke v Windows domorodni aplikaciji [38]

3.3 Prednosti in slabosti domorodnih mobilnih aplikacij

Prednosti in slabosti domorodnih mobilnih aplikacij vplivajo na izbiro tega razvojnega pristopa. Med prednosti domorodnih mobilnih aplikacij prištevamo karakteristike, kot so [50] bogatejši in prijaznejši uporabniški vmesnik (uporaba strojne in programske opreme je omogočena zaradi dostopa preko API vmesnikov), sposobnost delovanja brez omrežja (razbremenitev baterije naprave in pomnilnika), trajnostni razvoj in posodabljanje (povečanje interesa in interakcije s končnim uporabnikom), možnost monetizacije, popolna podpora s strani aplikacijskih trgovin (uporabniki lažje najdejo domorodne aplikacije), varnost (distribucija preko aplikacijskih trgovin je možna samo s predhodno verifikacijo in validacijo). Seveda imajo domorodne aplikacije tudi svoje slabosti, in sicer [50] veliki stroški razvoja in dolg proces validacije pred objavo v aplikacijskih trgovinah, nekompatibilnost s starejšimi različicami operacijskega sistema (fragmentacija predvsem Android operacijskega sistema), neprestano posodabljanje zaradi novih funkcionalnosti, neprestani razvoj zaradi izpolnjevanja potreb trga in končnih uporabnikov, problematika, kako najti aplikacijo v nasičenem trgu in zagotoviti uspešnost le-te pri končnih uporabnikih [75].

Poglavitni pristopi razvoja so razvoji Android, Apple in Windows mobilnih aplikacije, ki v primerjavi z Windows aplikacijo upoštevajo enake principe razvoja in oblikovanja. Windows domorodne aplikacije so edine, ki vključujejo spletne tehnologije, kot so označevalni jezik HTML5, oblikovalski jezik CSS3 in skriptni jezik JavaScript, za ustvarjanje odzivne rešitve za različne naprave in resolucije. Med drugim so prednosti domorodnih aplikacij v primerjavi s spletnimi uporaba API vmesnikov, s pomočjo katerih lahko enostavno dostopamo do strojne in programske opreme operacijskega sistema. Možnost objavljanja aplikacij v aplikacijskih trgovinah predstavlja monetizacijo, preko katere dobimo povračilo stroškov. Pri izboru razvojnega pristopa je primerno analizirati, katere funkcionalnosti potrebujejo in zahtevajo naši končni uporabniki. Domorodne mobilne aplikacije so prava odločitev, kadar potrebujemo hitre, učinkovite, tesno povezane z operacijskim sistemom in grafično sposobne rešitve v kombinaciji z zadovoljivim uporabniškim vmesnikom.

4 MOBILNE APLIKACIJE, RAZVITE S SPLETNIMI TEHNOLOGIJAMI

Mobilne aplikacije, temelječe na spletnih tehnologijah, so sestavljene iz treh različnih tehnologij, in sicer iz označevalnega jezika HTML5, oblikovalnega jezika CSS3 in skriptnega jezika JavaScript. Označevalni jezik HTML5 predstavlja peto različico spletnega označevalnega jezika in tudi prvo verzijo, ki podpira multimedijske vsebine brez dodatnih vtičnikov [60]. Standard HTML5 je bil razvit z idejo, da lahko spletne aplikacije delujejo brez neprestane internetne povezave – brezomrežno delovanje (ang. offline). Sodeč po raziskavah Johansson D. in Andersson K., lahko trdimo, da bodo spletne tehnologije v bližnji prihodnosti izkazale enako, če ne celo boljšo zmogljivost kot domorodne in hibridne rešitve [40]. Prednosti HTML5 označevalnega jezika vključujejo napredne komponente uporabniškega vmesnika, dostop do bogatih medijskih tipov, storitve za geolociranje ter možnost brezomrežnega delovanja [55].

Dovzetnost do spletnih mobilnih aplikacij v primerjavi z domorodnimi se skriva ravno v aplikacijski prenosljivosti. Aplikacijska prenosljivost je zagotovljena, kadar je omogočeno predvajanje aplikacije v mobilnih brskalnikih ter kadar je omogočena medplatformska podpora, ki pripelje do velikega finančnega prihranka tako za razvijalce kot tudi za podjetja [8]. Na trgu se pojavlja vedno več razvojnih orodij za razvoj spletnih in hibridnih mobilnih aplikacij. Razvojni orodji, kot sta Adobe PhoneGap in Apache Cordova, se uveljavljata kot orodji, s pomočjo katerih lahko ustvarimo zmogljive hibridne ter spletne mobilne aplikacije. Poglavitna prednost razvoja spletnih mobilnih aplikacij pa je v dejstvu, da končnemu uporabniku ni treba neprestano posodobljati aplikacije ali je namestiti lokalno na svoj mobilnik, saj je le-ta dosegljiva preko spleta [8]. Značilna za spletne aplikacije je njihova trinivojska arhitektura [39], sestavljena iz predstavitvenega nivoja (uporabniški vmesnik), logičnega nivoja (strežniški del) ter podatkovnega nivoja (podatkovne baze). Mnenja in podatki glede zmogljivosti spletnih mobilnih aplikacij so mešani. Razlike so očitne predvsem v naboru funkcionalnosti, saj spletne mobilne aplikacije grafično niso enako sposobne kot domorodne različice.

Aplikacijska mobilnost je dodaten atribut, ki ga lahko upoštevamo kot prednost tovrstnih aplikacij. Kriteriji, ki jih mora aplikacija izpolniti, da ji pripišemo aplikacijsko mobilnost, so [40]:

- **Distribucija aplikacije in identifikacija:** pomembna je tesna integracija distribucije aplikacije s sistemom. Aplikacija naj nosi identiteto avtorja ter pravila o uporabnosti, tj., kdo jo lahko uporablja in pod katerimi pogoji.
- **Zavedanje konteksta in kakovost konteksta:** pogoj aplikacijske mobilnosti je medplatformsko delovanje, ki aplikacijo neposredno prisili v zavedanje okolja, in na podlagi tega prikazovanje relevantnega, lokalnega konteksta.
- **Nemoteno nadaljnje izvajanje:** kadar pride do migracije aplikacije z ene na drugo napravo, je izpostavljena pomembnost hitrosti migracije. Migracija naj bo čim bolj nemotena in neopazna, tako da ne vpliva direktno na uporabniško izkušnjo in njen kontekst.
- **Heterogenost:** fragmentacija števila mobilnikov in njihovih operacijskih sistemov posledično povzroči veliko potrebo po migraciji aplikacij, kar neposredno zahteva tudi oblikovanje in izpolnjevanje vseh naštetih zahtev.
- **Uporabnost:** čeprav uporabnost ni splošno gledano zahteva za aplikacijsko mobilnost, pa je vendar ena glavnih smernic pri razvoju aplikacij. Aplikacija mora biti ne glede na to, ali je mobilna ali ne, enostavna za uporabo, intuitivna ter robustna.

Tabela 4.1 prikazuje rešitve za posamezne zahteve glede aplikacijske mobilnosti. Aplikacijska mobilnost je del razvoja spletnih mobilnih aplikacij. Večina rešitev zajema učinkovito uporabo spletnih tehnologij, predvsem novosti, s katerimi se je predstavil standard HTML5 (web workers – spletni delavci, web sockets – spletna vtičnica) in oblikovalski jezik CSS3.

Tabela 4.1: Zahteve in njihove arhitekturne rešitve za aplikacijsko mobilnost [40]

Zahteva	Arhitekturna rešitev
Distribucija aplikacije in identifikacija	URI, spletna varnost, lokalna podatkovna baza, predpomnilnik, Web Workers, Web Sockets
Zavedanje konteksta in kakovost konteksta	Lokalna podatkovna baza, notifikacije, lokalno osnovane storitve, XML privolitev

Zahteva	Arhitekturna rešitev
Nemoteno nadaljevanje	Brskalnikove funkcionalnosti, Web Workers
Heterogenost	CSS3, detekcija funkcionalnosti
Uporabnost	Poznanost HTML, uporabniški vmesnika brskalnika, prepoznavanje funkcionalnosti, podpora multimedijskih vsebin

4.1 Spletne mobilne aplikacije in odzivne spletne strani

Spletne mobilne aplikacije niso sinonim za odzivne spletne strani, čeprav ni nič nenavadnega, če omenjena koncepta ne razlikujemo popolnoma. Seznanjenost z brskanjem po spletu preko mobilnika danes ni tuja. V prvi vrstise srečamo s spletnimi stranmi, ki so optimizirane za mobilnike, imenovane tudi odzivne spletne strani, ki so del prakse odzivnega oblikovanja, imenovanega Responsive Web Design (RWD). Odzivne spletne strani avtomatično prepoznajo napravo, s katero dostopamo do spletne strani, ter prilagodijo prikazovanje vsebine njeni resoluciji. Velikokrat se vključi tudi rešitev za naprave na dotik. Uporaba skriptnih rešitev na podlagi JavaScript (dojox.mobile, Sencha Touch in JQuery Mobile) ni nič nenavadnega, kadar želimo uporabniku omogočiti enako uporabniško izkušnjo na dotik, kot so je uporabniki navajeni na domorodnih rešitvah [34]. Primerjava med spletnimi aplikacijami in spletnimi stranmi je v celoti vidna v Tabela 4.2. Primerjali smo ta dva pristopa v kategorijah orodja in znanja, izvedba, uporabniška izkušnja in zmogljivost.

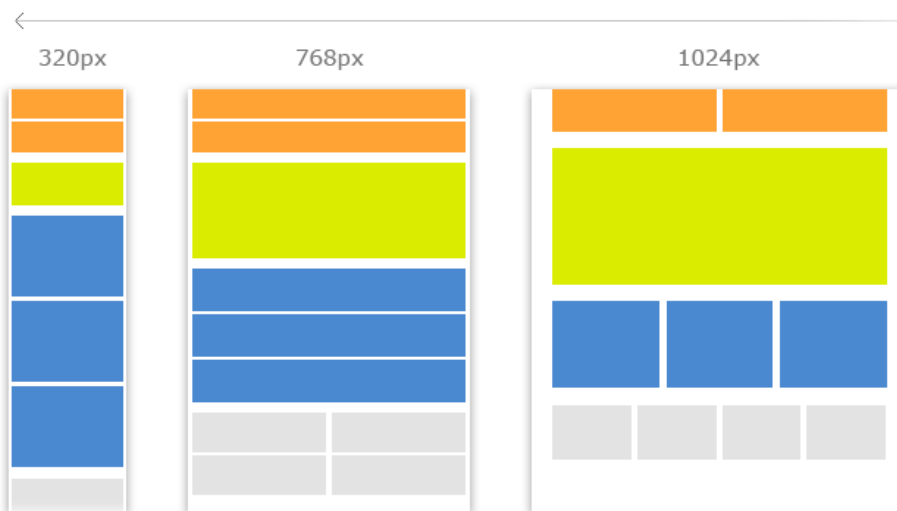
Tabela 4.2: Razlika med spletnimi mobilnimi aplikacijami in stranmi [34]

Lastnosti	Mobilne spletne aplikacije	Mobilne spletne strani
Orodja in znanja	HTML, CSS in JavaScript	HTML, CSS in JavaScript
Izvedba	Namestimo bližnjico, ki omogoča zagon kot domorodna mobilna aplikacija.	Dostopamo s pomočjo navigiranja do specifične spletne strani preko URL.

Lastnosti	Mobilne spletne aplikacije	Mobilne spletne strani
Uporabniška izkušnja	Prijazno na dotik, interaktiven uporabniški vmesnik.	Navigacijski uporabniški vmesnik med spletnimi stranmi za prikazovanje statičnih podatkov.
Zmogljivost	Uporabniški vmesnik je shranjen lokalno, kar ustvari odzivno aplikacijo, dostopno tudi brez interneta.	Vsa koda se izvede iz strežnika, kar povzroči odvisnost od povezave z internetom.

Pomembno je omeniti WebKit brskalnike, ki so v bistvu Google Chrome in Apple Safari – oba brskalnika sta odprtokodna in popolnoma podpirata funkcionalnosti označevalnega jezika HTML5 [34]. Ker so spletne mobilne aplikacije napisane v spletnih jezikih, kot so označevalni jezik HTML5, oblikovalski jezik CSS3 in skriptni jezik JavaScript, pomeni, da so avtomatično bolj dovzetni za medplatformsko kompatibilnost. Na osnovi 6.000 anketiranih razvijalcev v anketi, izvedeni leta 2015, je bilo dokazano, da zanimanje za razvoj spletnih mobilnih aplikacij ter z njimi povezane tehnologije in orodji narašča. Število je naraslo s 23 % na 30 % v letu 2015 [22]. V primerjavi z domorodnimi mobilnimi aplikacijami, ki jih lahko direktno zaženemo z naših mobilnikov in se tudi lokalno namestijo, se spletne mobilne aplikacije izvajajo znotraj brskalnikov. Brskalniki predstavljajo domorodne mobilne aplikacije, ki imajo popoln dostop do API vmesnikov operacijskega vmesnika, vendar filtrirajo dostopanje do specifičnih funkcionalnosti kot preventiva pred zlorabo [34].

Odzivne spletne strani avtomatično prepoznajo resolucijo in zmogljivosti naprave, s katero končni uporabnik dostopa do spletne strani, ter prilagodijo videz spletne strani glede na to napravo. Odzivnost tega obsega dosežemo z integracijo medijskih predpisov (ang. media queries), ki predstavljajo poglobljeno značilnost oblikovalskega jezika CSS3, in sicer možnost prilagoditve strani napravi. Praksa je dokaj podobna kot pri Windows Phone aplikacijah iz poglavja 3.2.3. Spletni strani dodamo tri glavne prelomne točke, in sicer za mobilnike (320 pikslov), za vmesne naprave, kot so tablice in dlančniki (768 pikslov), ter prenosnike in osebne računalnike (1024 pikslov). Slika 4.1 nazorno prikazuje, kako se vsebina porazdeli glede na resolucijo in orientacijo naprave. To pa nikakor ne pomeni, da ni mogoče definirati posebnih pravil za vse naprave znotraj in zunaj tega spektra.



Slika 4.1: Primerjava odzivnosti spletne strani glede na napravo [58]

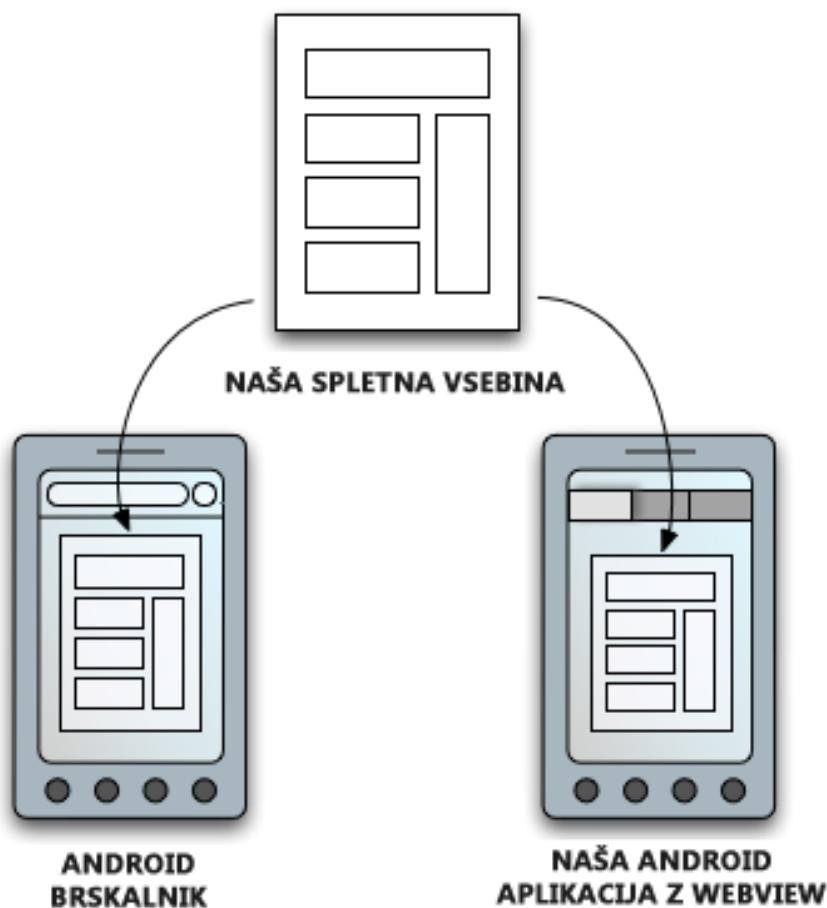
Princip odzivnost je tudi vključiti in prilagoditi vsebino spletnih strani ter navigacije tako, da je prijazna za naprave na dotik. Ob večjih rešitvah, kot so orodja za razvoj odzivnih spletnih strani, si lahko pomagamo z ogrodji, kot so Foundation 3, Bootstrap ter podobnimi rešitvami, kot so Skeleton, CardinalCSS, ConciseCSS, PowertoCSS, Furtive, BassCSS. To so ogrodja, kjer lahko spletne strani oblikujemo po principu stolpcev in mrežnega sistema, ki se nato manjšajo ali večajo glede na podano resolucijo. Večina ogrodij, kot so Foundation 3, Bootstrap in tudi Skeleton, predstavlja rešitve, ki pridejo z vnaprej definiranimi komponentami uporabniškega vmesnika (npr.: oblikovane notifikacije, gumbi, različni obrazci, ikone ter sezname in celotne definicije tipografije). Kadar uporabljamo vnaprej pripravljen mrežni sistem, se preureditev komponent uporabniškega vmesnika zgodi avtomatično glede na podane procentne vrednosti posameznih stolpcev pri določeni prelomni točki.

4.2 Spletne mobilne aplikacije, WebView in WebKit

Mobilno aplikacijo lahko razvijamo na tri načine: kot domorodno mobilno aplikacijo s pomočjo SDK orodja, kot spletno mobilno aplikacijo s pomočjo spletnih tehnologij in CSS ogrodji ali kot hibridno mobilno aplikacijo, kjer spletno aplikacijo ovijemo v domorodno ogrodje. Kot smo v tretjem poglavju že obrazložili, se domorodne aplikacije izvajajo preko

namestitve na napravo, do spletnih aplikacij pa dostopamo preko brskalnika, kjer se vsebina aplikacije prikaže glede na velikost naprave in predčasna namestitvev ni potrebna.

Android omogoča dva poglavitna načina, kako aplikacijo dostaviti končnemu uporabniku: kot domorodno mobilno aplikacijo (razvijemo aplikacijo s pomočjo SDK orodja) ali kot spletno mobilno aplikacijo (uporabnik dostopa do nje preko brskalnika) [2]. S pomočjo uporabe Android brskalnika v kombinaciji z WebView ogrodjem lahko kot razvijalec določimo vidno točko (ang. viewport) ter stil spletne strani optimiziramo za napravo. Uporaba vidne točke je dodaten način, kako lahko povečamo ali zmanjšamo našo spletno stran na izbrano resolucijo in orientacijo naprave. V bistvu je WebView ogrodje, s pomočjo katerega lahko ovijemo našo spletno vsebino in jo prikažemo kot aplikacijo končnemu uporabniku. Slika 4.2 prikazuje kako lahko vsebino spletne strani dostavimo končnemu uporabniku: ali jo prikažemo kot tradicionalno spletno stran v brskalniku ali kot ovito aplikacijo znotraj WebView ogrodja.



Slika 4.2: Prikazovanje spletne mobilne aplikacije v Android brskalniku [2]

Testiranje spletnih mobilnih aplikacij poteka na enak način, kot bi testirali spletne strani. Uporabljamo orodja, kot sta Chrome Developer Tools od podjetja Google ali FireBug rešitev od Firefox. Čeprav obstaja rešitev tudi za Apple operacijski sistem se v sklopu magistrskega dela osredotočamo zgolj na Windows rešitve, specifično na Chrome Developer Tools, ker je rešitev integrirana v sam Chrome brskalnik. Zajemanje podatkov med eksperimentom in testiranjem spletne mobilne aplikacije je potekalo s pomočjo principa oddaljenega razhroščevanja (ang. remote debugging), ki je podprt v Android operacijskem sistemu od verzije 4.4 naprej [1]. Več o tem v šestem poglavju, kjer se osredotočamo na sam eksperiment in analizo empiričnih podatkov eksperimenta in ankete.

4.2.1 Chrome Developer Tools

Orodje Chrome Developer Tools se uporablja za preverjanje ozkih grl spletnih strani, problemov glede časa nalaganja, zmogljivosti, preverjanja uporabnosti virov ter pregledovanja strukture spletne strani [12]. Zmogljivost smo predhodno definirali kot efektivnost sistema, da lahko obdela določeno količino podatkov v določenem času ter pri tem poskrbi za optimizirano odzivnost ter stabilnost sistema. V našem primeru je sistem zbirka spletnih strani [62]. Orodje DevTools ponuja možnost natančnega analiziranja delovanja brskalnika in vsebine v njem. Tako lahko pregledamo kakršnekoli DOM probleme, ugotovimo, ali nastajajo zamude zaradi skriptnega jezika JavaScript, in dobimo še dodatne optimizacijske nasvete glede kode in celotne vsebine spletne strani [12]. Osnovne skupine, v katerih lahko analiziramo obnašanje, zmogljivost in odzivnost spletne strani oziroma aplikacije, so:

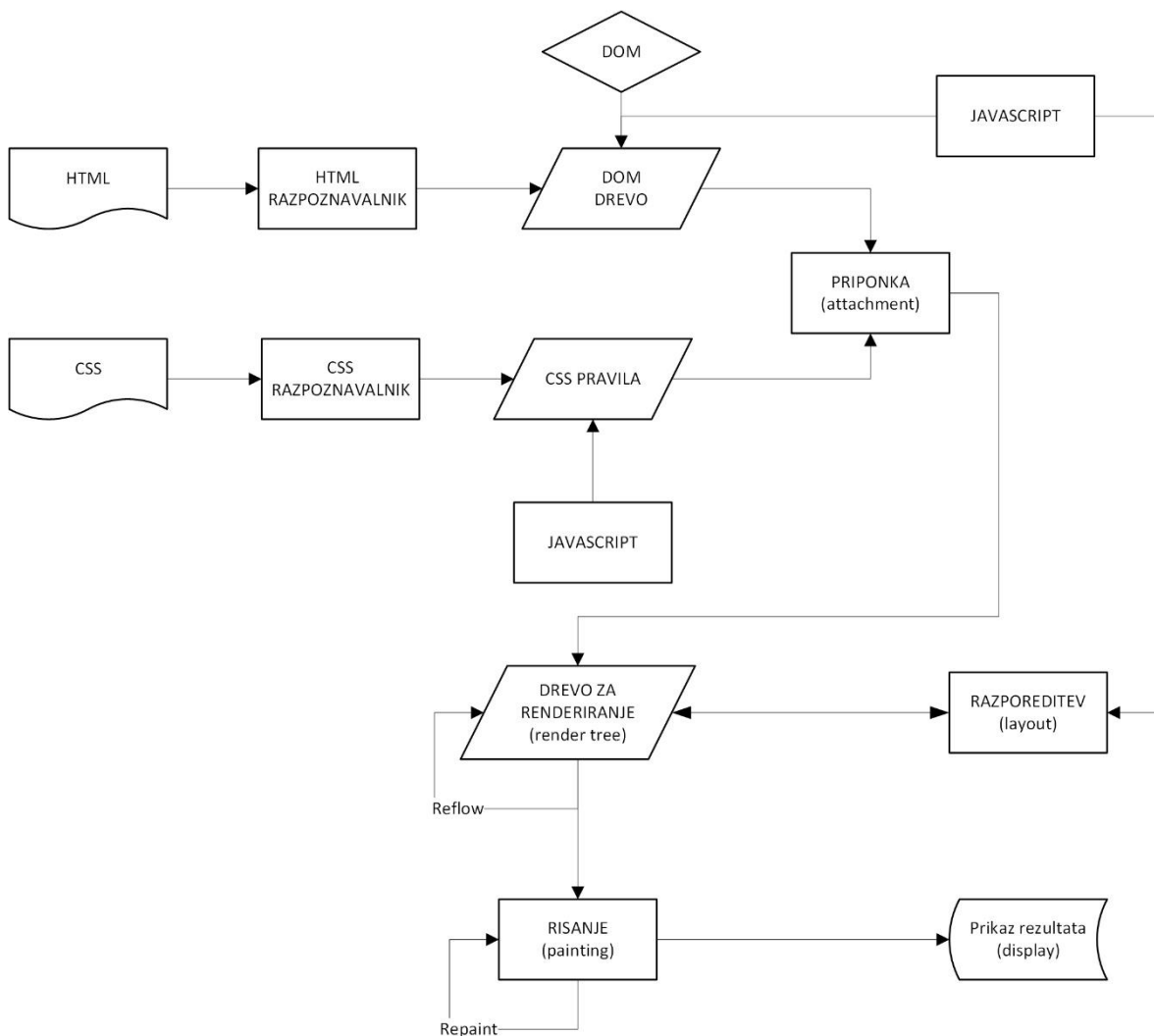
- **Elementi:** kategorija za pregledovanje DOM elementov, manipuliranje z njimi, dodajanje ali odstranjevanje označevalnega jezika HTML in oblikovalskega jezika CSS ter popravljanje razporeditev elementov v realnem času [13].
- **Sredstva:** kategorija, s pomočjo katere lahko pregledamo lokalno stanje naše aplikacije in njenih podatkovnih baz (IndexedDB, WebSQL) ter lokalne in strežniške podatkovne seje, piškotke in aplikacijska sredstva predpomnilnika (ang. cache) [15].
- **Omrežje:** kategorija, ki igra največjo vlogo, kadar želimo definirati zmogljivost naše spletne aplikacije, saj vsebuje natančno analizo vsakega prenosa, HTTP zahtev in odgovorov, odzivnost piškotkov, WebSocket podatkov in več. S pomočjo te kategorije lahko odgovorimo na vprašanja glede virov, ki potrebujejo največ časa za nalaganje, zakaj je naša spletna aplikacija tako počasna in kako se lahko znebimo nepotrebne obremenjenosti virov [14].

- **Viri:** kategorija, kjer lahko posamezne vire, pridobljene s strani strežnika pregledamo s pomočjo razhroščevanja, in opazujemo njihovo obnašanje.
- **Časovnica:** je vizualna predstavitev vseh aktivnosti, ki se odvijajo od časa, ko zaženemo našo aplikacijo, do trenutka, ko prenehamo z opazovanjem. Je najboljše orodje za določevanje ozkih grl v naši aplikaciji, problemov z zmogljivostjo in slabo odzivnostjo. Opazujejo se predvsem štiri osnovne aktivnosti aplikacije, in sicer nalaganje dogodkov, nalaganje skriptnih delov, predelava podatkov (ang. rendering) in izris [16].
- **Profili:** omogočajo kategoriziranje izvajalnega časa procesorja ter pregled uporabljenega pomnilnika za izvajanje spletne aplikacije. Poznamo dva aktivna profila ter CSS profil za selektorje. Procesni profil (ang. CPU profile) prikazuje izvajalni čas, potreben za nalaganje JavaScript skript, medtem ko profil skladišča (ang. heap profile) prikazuje distribucijo pomnilnika za nalaganje JavaScript objektov kot tudi DOM povezanih vozlišč [17].
- **Audits:** predstavlja nabor pravil, s katerimi lahko našo spletno aplikacijo primerjamo. Med temi pravili lahko testiramo zmogljivost spletne strani, njeno berljivost, izkoriščenost omrežja ter stabilnost aplikacije.
- **Konzola** je zadnja kategorija in predstavlja dejansko konzolo, kjer lahko preverjamo vrednosti JavaScript izrazov ali pa izvajamo določene ukaze preko konzole.

4.2.2 WebView in WebKit

Android rešitev WebView skrbi za procesiranje spletnih strani znotraj brskalnika in kot končni rezultat ustvari spletno stran, ki je videti kot aplikacija. Enako rešitev lahko najdemo pri Apple operacijskem sistem z imenom WKWebView za iOS 8 – princip ostaja enak, le ime se spreminja odvisno od operacijskega sistema. WebKit je ogrodje za predelovanje spletnih strani, na podlagi česar se tudi HTML elementi pravilno izrišejo in s pomočjo oblikovalskega jezika CSS pravilno oblikujejo [76]. Vsak brskalnik ima svojo domorodno rešitev glede ogrodja za predelavo podatkov in zato prihaja tudi do težav med platformami glede kompatibilnosti. WebKit je v bistvu rešitev, katere pobudnika sta bila Google in Apple, čeprav je Google leta 2013 zapustil odprtokodni projekt WebKit in namesto njega ustvaril Blink, ki se danes uporablja v zadnjih različicah Opera brskalnika in Chrome [79]. Tudi Microsoft ima svoje rešitve na tem področju, imenovane EdgeHTML za brskalnik Edge, Tasman za Microsoft Entourage in Trident za Internet Explorer [79].

WebKit so torej orodja, s pomočjo katerih predelamo podatke, ki jih pridobimo kot strukturo spletne strani in oblike (označevalni jezik HTML poskrbi za strukturo, medtem ko spletno stran oblikujemo s pomočjo oblikovalskega jezika CSS) v končne, spletne strani za uporabnika. Delovni tok tovrstnega orodja prikazuje Slika 4.3. Najprej se predelajo HTML in CSS pravila ter procesirajo naprej v obliki priponke do drevesa za obdelavo, kjer ta poskrbi za pravilno razporeditev elementov glede na pridobljena DOM pravila. Ta proces se ponavlja, dokler se ne razporedijo vsi elementi na spletni strani, nato pa se izriše končna slika uporabniku v brskalnik.



Slika 4.3: Delovni tok delovanja WebKit ogrodja [27]

Obdelava in procesiranje podatkov torej poteka s pomočjo orodja WebKit. V naslednjem poglavju si bomo ogledali še razvojna orodja za spletne mobilne aplikacije.

4.3 Razvojna orodja

Orodja, kot so PhoneGap, Titanium in Apache Gordova, so namenjena razvoju hibridnih rešitev. Ker se želimo osredotočiti na spletne mobilne aplikacije, bomo v tem poglavju pregledati popularne rešitve, kot so JQuery Mobile, Codiqa in Sencha Touch.

4.3.1 JQuery Mobile in Codiqa

JQuery Mobile je ogrodje na osnovi označevalnega jezika HTML5, kjer lahko ustvarimo odzivne spletne strani, ki so avtomatično prilagojene na dotik. Orodje je sestavljeno iz treh komponent, in sicer iz skripte JQuery, JQuery uporabniških komponent in Ajax, ki poskrbi za animirano navigacijo med posameznimi stranmi in dogodki na dotik [41]. Druge funkcionalnosti vključujejo tudi uporabo funkcije ThemeRoller, kjer lahko definiramo globalno temo oziroma videz naše aplikacije. Dokaj podobna rešitev JQuery Mobile je Codiqa, ki deluje na princip »povleci in spusti« v kombinaciji z označevalnim jezikom HTML5 ter zagotavlja popolnoma funkcionalno medplatformsko rešitev [18].



Slika 4.4: Primer spletne mobilne aplikacije ustvarjenje s pomočjo JQuery Mobile [41]

Na Slika 4.4 vidimo spletno mobilno aplikacijo, ustvarjeno s pomočjo JQuery Mobile, kjer so razvidne posamezne uporabniškega vmesnika (navigacija in gumbi).

Spletne strani ustvarjene z JQuery Mobile dajejo končnemu uporabniku vtis, da ta uporablja aplikacijo in ne zgolj spletne strani – integrirana so pravila za simuliranje dogodkov na dotik.

Prednosti uporabe JQuery Mobile, Codiqa in njemu podobnih ogrodij so [19]:

- **Enostavna uporabnost:** v primerjavi z domorodnimi mobilnimi aplikacijami pri JQuery Mobile ni potrebno predhodno znanje programiranja, saj je ustvarjanje enostavno in intuitivno.
- **Izogibanje odobritvi aplikacijskih trgovin in razhroščevanju:** enostavno se lahko izognemo razhroščevanju ter odobritvi distribucije aplikacije, saj jo posredujemo končnemu uporabniku preko spletnega naslova.
- **Avtomatična medplatformska podpora:** spletne mobilne aplikacije in spletne strani ustvarjene s pomočjo JQuery Mobile avtomatično delujejo na vseh operacijskih sistemih ter na različnih napravah. S tem se izognemo resni fragmentaciji naprav in optimizaciji aplikacije glede na resolucijo mobilnika in verziji operacijskega sistema.

Vsaka rešitev pa ima tudi slabosti, in sicer [19]:

- **Spletna aplikacije je počasnejša kot domorodna:** JQuery Mobile aplikacije so občutno počasnejše od domorodnih.
- **Večja omejitev glede komponent uporabniškega vmesnika in zahtevnejše oblikovanje:** orodja »povleci in spusti« so omejena glede nabora opcij, ki jih imamo kot končni uporabnik pri oblikovanju.
- **Ni primerna rešitev za razvoj mobilnih igralnih aplikacij:** JQuery Mobile je primerno predvsem za enostavne aplikacije in ne za grafično zahtevnejšo obdelavo podatkov.
- **Omejitev glede dostopanja vseh domorodnih funkcionalnosti naprave:** JQuery Mobile ne omogoča dostopa do vseh funkcionalnosti strojne in programske opreme preko API vmesnikov.

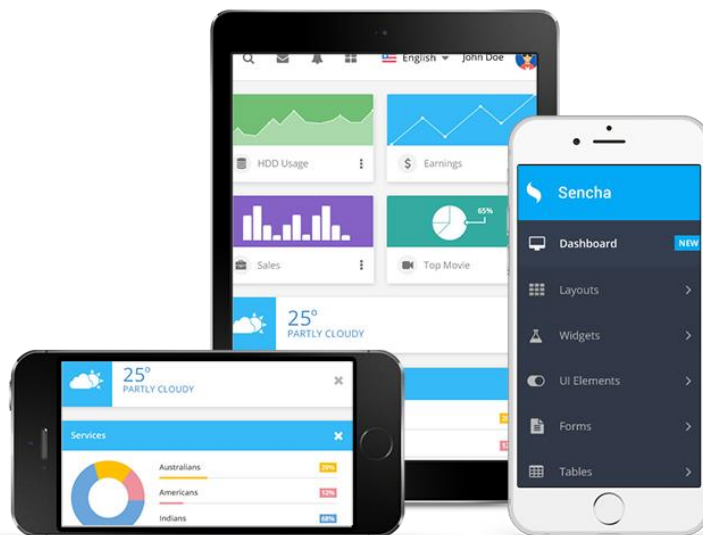
4.3.2 Sencha Touch

Sencha Touch je v mnogo pogledih podobna rešitev kot jQuery Mobile, le da se primarno uporablja kot orodje za izdelavo hibridnih mobilnih aplikacij [65]. V tem poglavju omenimo ogrodje zgolj zaradi njegove popularnosti in podobnosti z jQuery Mobile. Omogoča razvoj s spletnimi tehnologijami, kot so označevalni jezik HTML5, oblikovalski jezik CSS3 in skriptni jezik JavaScript. Zagotavlja rapidno oblikovanje, kodiranje ter upravljanje aplikacij ki omogočajo medplatformsko podporo. V primerjavi z ostalimi ogrodji omogoča tudi razvoj

aplikacij na osnovi MVC vzorca [65]. Ogrodje med drugimi zagotavlja funkcionalnosti in izpolnitev zahtev, kot so [65]:

- visoka zmogljivost, domorodno podobni UI-pripomočki,
- prilagoditveni razpored, animacije ter gladke transformacije,
- diagnostični podatkovni paket,
- napredni paket za ustvarjanje grafov in prikazovanje podatkov,
- funkcionalnosti naprave ter domorodno pakiranje.

Zadnja različica orodja Sencha Touch 2.0 na žalost ni odprtokodne narave, si pa prizadeva poenotiti spletne in hibridne mobilne aplikacije glede stabilnosti, hitrosti in vizualne dodelanosti, kar so večkrat omenjene pomanjkljivosti tovrstnih spletnih aplikacij [63]. Med drugim podpira tudi REST ter SOAP API vmesnike, MVC arhitekturo in velik nabor privzetih komponent uporabniškega vmesnika. Slabosti samega ogrodja so njegova kompleksnost, problematika, povezana z razvojem spletnih aplikacij na osnovi JavaScript (počasne animacije zaradi izvajanja skriptnega jezika JavaScript namesto CSS datotek) ter dejstvo, da ogrodje ni odprtokodno. Primer zmogljivosti Sencha Touch ogrodja in integracija rešitve v obliki medplatformskega razvoja so vidne na spodnji sliki, Slika 4.5.



Slika 4.5: Primer spletne mobilne aplikacije, ustvarjene s pomočjo Sencha Touch 2 [65]

Pomembno je izpostaviti dejstvo, da obstaja ogromno orodij za razvoj spletnih aplikacij, med drugim tudi Kendo UI, ki je optimiziran za hiter in lahkoten razvoj aplikacij, saj ima popolnoma integriran AngularJS ter podporo za Bootstrap CSS ogrodje [43]. Predstavili

smo JQuery Mobile in Sencha Touch, ker sta ti dve ogrodji vodilni na svojem področju razvoja spletnih in hibridnih mobilnih aplikacij.

4.4 Kratka predstavitev spletnih tehnologij

Spletne tehnologije so vse tehnologije, s pomočjo katerih lahko ustvarjamo in povezujemo različne podatke na spletu. Srečamo se s terminologijo, kot je Splet 1.0 in Splet 2.0. Verzija 1.0 predstavlja predvsem statičen splet, medtem ko so se zadeve začele razvijati v bolj organski, dinamičen splet v verziji 2.0. V tem podpoglavju bodo predstavljene osnovne spletne tehnologije, in sicer označevalni jezik HTML5, oblikovalski jezik CSS3 in skriptni jezik JavaScript.

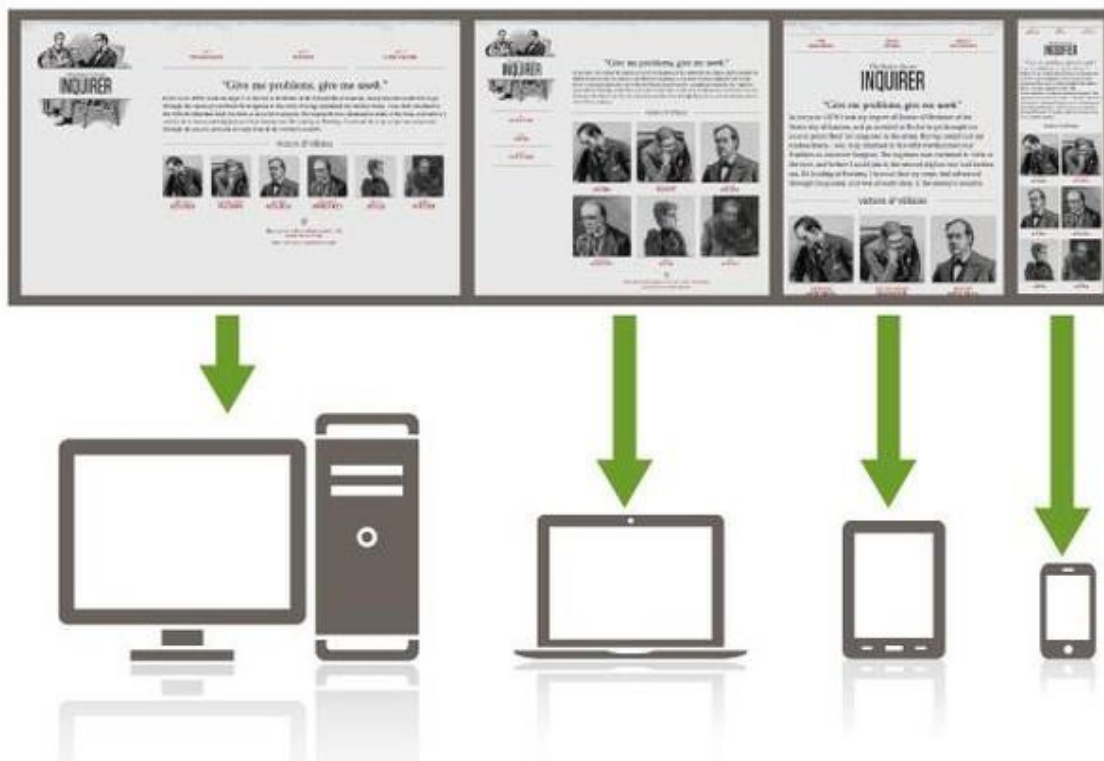
4.4.1 HyperText Markup Language (HTML 5)

Označevalni jezik HTML je osnovna tehnologija in standard, na podlagi katerega so zgrajene strukture spletnih strani. Njegov obstoj sega v leto 1993, medtem ko trenutna verzija 5, imenovana tudi HTML5, predstavlja standardiziran označevalni jezik s strani W3C [81]. V osnovi je označevalni jezik sestavljen iz posameznih HTML značk, lastnosti in raznih specifikacij ter pravil, s pomočjo katerih dostopamo do posameznih elementov znotraj DOM drevesa in z njimi strukturiramo našo spletno stran. Označevalni jezik HTML služi tudi kot orodje za odkrivanje kompatibilnosti posameznih brskalnikov [55]. Njegova enostavnost in nadgradljivost pripomore k njegovi razširjenosti. Verzija 5 je temeljito izboljšala zmogljivost, kompatibilnost ter odzivnost spletnih strani, saj je v celoti tudi kompatibilna s starejšo verzijo jezika. Delovanje standarda HTML5 je avtomatizirano, saj se večina njegovih novosti avtomatično izvaja znotraj brskalnika brez vednosti končnega uporabnika. Trenutno je spletni standard najbolj podprt s strani Google Chrome. Test podpore standarda je pokazal, da je največja podpora pri Google Chrome brskalniku z 501 točko, Operi s 497 točkami in Firefox s 449 točkami [48]. Med novosti standarda prištevamo tudi spletne delavce in spletne vtičnike – spletni delavci pripomorejo k hitrejšem nalaganju in obdelavi zahtev, saj izvajajo storitve neodvisno od drugih aktivnosti v ozadju brskalnika [83].

4.4.2 Cascading Style Sheet (CSS3)

Cascading Style Sheet predstavlja oblikovalski jezik, s pomočjo katerega definiramo videz posameznega HTML strukturnega elementa. Prva verzija jezika je izšla leta 1997, trenutna, tj. tretja, verzija pa je aktivna že od leta 2012. Novosti zadnje verzije se raztezajo vse od definiranja barv (HSLA, RGBA), novih selektorjev, definiranja odzivnosti, fleksibilne razporeditve do testne integracije z Ruby, transformacije in tranzicije ter uporabe MathML

[28]. Asociacija oblikovalskega jezika CSS je predvsem v zmožnosti dodajanja odzivnosti spletne strani, ki smo se je okvirno dotaknili pri razvoju Windows Phone aplikacij (poglavje 3.2.3 Windows Phone SDK) ter v poglavju 4.1. Transformacija spletne strani glede na napravo končnega uporabnika je vidna na Slika 4.6.



Slika 4.6: Odzivnost spletne strani [61]

4.4.3 JavaScript

JavaScript in Java nista enaka jezika – JavaScript je dinamični skriptni jezik, kjer lahko definiramo funkcije in uporabljamo principe objektno orientiranega razvoja, medtem ko je Java programski jezik. Najpogosteje se uporablja skriptni jezik JavaScript v spletnih straneh in aplikacijah, kjer potrebujemo aktivne skriptne za dinamično obdelavo podatkov (komunikacija ali spreminjanje vsebine HTML dokumenta – DHTML oziroma dinamični označevalni jezik HTML). Prvič se je skriptni jezik JavaScript pojavil leta 1995 pod okriljem Netscape [82]. Od takrat se je JavaScript zelo spremenil in leta 1999 postal vodilni skriptni jezik. Njegova popularnost je podprta z imperativnim razvojem kot tudi s strukturirano kodo, z dinamičnim razvojem in z objektno orientiranim programiranjem s pomočjo funkcij. Skriptni

jezik JavaScript se uporablja tudi pri preprečevanju varnostnih vdorov ter izboljšanju uporabniške izkušnje slabovidnih.

Njegove prednosti so predvsem v dejstvu, da se izvaja na strani uporabnika in s tem razbremeni strežnik, osnovan je na enostavni sintaksi, skripte se izvajajo hitro in dinamično ter omogočajo dodatne funkcionalnosti spletnim stranem in aplikacijam [42]. Slabosti skriptnega jezika JavaScript zajemajo predvsem težave, povezane z varnostjo in inkonsistenco pri obdelavi podatkov, kar lahko privede do nepričakovanega obnašanja spletne strani [42]. Skriptni jezik se uporablja tudi v obliki jQuery (kot tudi jQuery Mobile) in je razlog za obstoj številčnih popularnih JavaScript knjižnic, kot so Hammer.js, Bspoke.js, Boba.js, EaseJS in druge.

4.5 Prednosti in slabosti mobilnih aplikacij razvitih s pomočjo spletnih tehnologij

Prednosti mobilnih aplikacij razvitih s pomočjo spletnih tehnologij so predvsem medplatformska podpora, relativno enostaven razvoj, cenejše vzdrževanje v primerjavi z domorodnimi rešitvami, enostaven in univerzalen dostop do aplikacije preko spletnega naslova v brskalniku, dostopanje preko brskalnika omogoča tudi uporabo določenega nabora API vmesnikov, ni potrebe po publikaciji in validaciji, kar pomeni, da se spletne mobilne aplikacije dostavijo končnim uporabnikom hitreje kot npr. domorodne aplikacije [50] [75]. Slabosti tega razvojnega pristopa so povezane z omejitvami dostopanja in uporabe tako strojne kot tudi programske opreme, okrnjenost funkcionalnosti kot posledica uporabe spletnih tehnologij, težje je nadzorovati število končnih uporabnikov in obdržati pregled nad njimi, za končne uporabnike lahko iskanje spletnih mobilnih aplikacij predstavlja izziv, primanjkovanje regulativnega kakovostnega nadzora lahko predstavlja resne varnostne luknje v aplikacijah [50] [75]. Kdaj se torej odločimo za spletni pristop razvoja mobilnih aplikacij? Karakteristike aplikacije, ki so značilne za ta tip razvoja, so [34] direktna distribucija (direktna objava na spletu preko spletnega naslova), pilotna aplikacija (ustvarimo pilotno aplikacijo, s katero najprej testiramo trg in na podlagi podatkov ustvarimo boljšo, dodelano verzijo aplikacije), vidljivost (avtomatično dostopna, ko ji dodelimo spletni naslov v primerjavi z domorodnimi aplikacijami, kjer končni uporabnik lahko dostopa do njih samo preko aplikacijskih trgovin).

Pozitivna lastnost spletnih mobilnih aplikacij je enostavna implementacija in brezskrbna distribucija med končne uporabnike. Omenjene spletne tehnologije, kot so označevalni jezik

HTML, oblikovalski jezik CSS in skriptni jezik JavaScript se konstantno dopolnjujejo in spreminjajo zaradi vedno novih zahtev po zmogljivosti in konkurenčnosti. Implementacija tovrstnih rešitev si prizadeva dohiteti domorodne rešitve v vseh faktorjih, kot so zmogljivost, uporabniška izkušnja in zadovoljstvo ter odzivnost in splošne funkcionalne sposobnosti. V sklopu magistrskega dela se ne bomo dodatno osredotočali na hibridne mobilne aplikacije, temveč v ospredje postavljamo zgolj domorodne in spletne mobilne aplikacije ter njune zmogljivosti, uporabniške izkušnje in zadovoljstvo.

5 RAZISKAVA

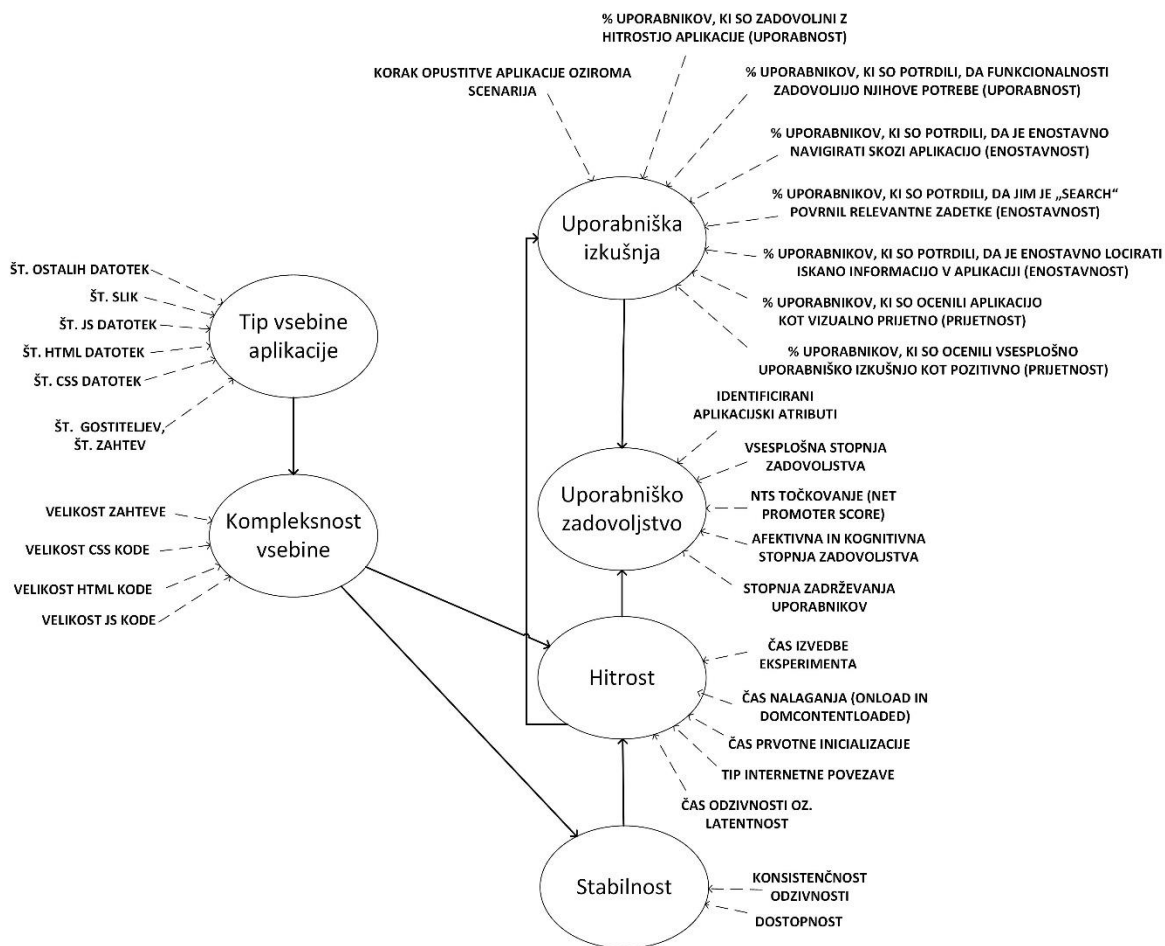
Raziskava je bila osredotočena na analizo zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah, kot so označevalni jezik HTML, oblikovalski jezik CSS in skriptni jeziki JavaScript. Zanimalo nas je predvsem, ali uporabniki občutijo razliko v zmogljivosti – torej v hitrosti, odzivnosti in stabilnosti ter uporabniški izkušnji in zadovoljstvu – med spletno in domorodno različico enake mobilne aplikacije. Z raziskavo smo želeli dokazati, da prihaja do razlik v zmogljivosti med spletno in domorodno različico mobilne aplikacije predvsem s strani uporabnikov v dojeti uporabniški izkušnji in zadovoljstvu. Pri raziskovanju smo najprej analizirali javno dostopno literaturo, na podlagi katere smo oblikovali teoretični model, ki je služil kot osnova za izvedbo eksperimenta in ankete, pri kateri je sodelovalo 50 udeležencev.

Na podlagi pregleda literature v različnih podatkovnih bazah (Google Scholar, IEEE Explorer in ScienceDirect) smo pridobili karakteristike zmogljivosti mobilnih aplikacij. Prišli smo do spoznanja, da raziskava, kjer bi se neposredno primerjale štiri aplikacije (Facebook, GMail, YouTube in Twitter) tako v spletni kot tudi v domorodni različici, še ni bila izpeljana. Aplikacije za eksperiment smo izbrali na podlagi Global Web Index strani [29], kjer so naredili raziskavo glede potrošniških navad v družbenih medijih v letu 2015. Izpostavili so 20 družbenih omrežjih glede na njihovo priljubljenost. V analizi so si našteje aplikacije sledile po naslednjem vrstnem redu: Facebook (80 %), YouTube (60 %), Twitter (50 %) in Google+ (60 %) [29]. Po nadaljnjem študiju literature in posvetu z mentorjem smo prišli do zaključka, da je bolje zamenjati Google+ z GMail storitvijo predvsem zaradi njene prepoznavnosti in uporabnosti. Na podlagi analize [29] smo nato sestavili seznam aktivnosti, ki so jih udeleženci izvedli v okvirju eksperimenta (Priloga A) – povzeli smo najpogostejše aktivnosti za posamezne aplikacije (Facebook, Twitter, YouTube in GMail). Udeleženci so ob pričetku eksperimenta prejeli navodila (seznam z navodili in opisom korakov aktivnosti), ki so jih izvedli pod nadzorom avtorja magistrskega dela.

Pregledu literature je sledilo oblikovanje teoretičnega modela raziskave, kot je prikazan na Slika 5.1, ki je nastal kot rezultat prebiranja člankov [49] [8] [46] [23] [67],

pregleda literature ter prebiranja spletnih objav in spletnih strani, kjer lahko testiramo zmogljivost spletnih strani (DynaTrace, YSlow, PageSpeed, GTMetrix). Definirali smo glavne karakteristike za zmogljivost: tip vsebine aplikacije, kompleksnost vsebine, uporabniško izkušnjo, uporabniško zadovoljstvo, hitrost in stabilnost. Tip vsebine aplikacije smo razširili z opisnimi spremenljivkami, ki definirajo količino dokumentov, ki se prenesejo v posamezni zahtevi in seji udeleženca med uporabo aplikacije (število zahtev, gostiteljev, CSS datotek, HTML datotek, JS datotek, slik in ostalih datotek). Vključili smo tudi število gostiteljev in zahtev, saj so se ti podatki konstantno pojavljali na spletnih testnih orodjih (PageSpeed, YSlow, GTMetrix). Kot kompleksnost vsebine smo definirali posamezne velikosti v bitih oziroma kilobitih. Opisne spremenljivke so velikost zahteve, CSS kode, HTML kode in JS kode. Osrednja kakovost aplikacije sta njena stabilnost in hitrost. Kot stabilnost smo definirali konsistentnost odzivnosti (ang. response time) in dostopnost. Konsistentnost odzivnosti predstavlja čas, ki ga strežnik potrebuje za odgovor, medtem ko dostopnost predstavlja vse pozitivne (200) in negativne (404, 500 in 301) HTTP odgovore. Hitrost smo definirali kot zbirko spremenljivk – čas odzivnosti, tip internetne povezave, čas prvotne inicializacije, čas nalaganja (tako onLoad kot tudi DOMContentLoaded dogodka) ter čas izvedbe eksperimenta. Uporabniško zadovoljstvo in uporabniško izkušnjo smo definirali na podlagi afektivne in kognitivne izkušnje [84]. Podatke o uporabniški izkušnji in uporabniškem zadovoljstvu smo pridobili iz ankete, kjer so udeleženci ocenili svoje izkušnje in zadovoljstvo s pomočjo 5-stopenjske lestvice (se popolnoma ne strinjam – se popolnoma strinjam) ali z ocenami (zelo slabo – odlično).

Nastali teoretičen model je služil kot sredstvo, s katerim smo ocenili zmogljivosti domorodnih in spletnih mobilnih aplikacij. Pomembno je poudariti, da smo se osredotočili na mobilne aplikacije, temelječe na spletnih tehnologijah, in ne na domorodne mobilne aplikacije, predvsem zaradi omejitev testne programske opreme pri izvajanju eksperimenta in zajemanju podatkov domorodnih mobilnih aplikacij. Ker so mobilne aplikacije, ki temeljijo na spletnih tehnologijah, v bistvu spletne strani, ki so optimizirane za mobilne aparate, igrajo njihova stabilnost, kompleksnost vsebine, tip vsebine in hitrost pomembno vlogo pri času nalaganja. Tako kot pri spletnih straneh se izpostavi pomembnost hitrosti časa nalaganja, čas prvotne inicializacije ter latentnost podatkov. Mobilne aplikacije so v vlogi odjemalca in so v našem primeru konstantno povezane s strežnikom, zato smo izpostavili tudi faktorje, kot sta stabilnost (konsistentnost odzivnosti strežnika in dostopnost le-tega) in kompleksnost vsebine.



Slika 5.1: Teoretični model analize

Na osnovi raziskovalnega modela smo formulirali hipoteze, predstavljane na začetku magistrskega dela. Pri izvedbi eksperimenta smo predhodno izpolnili dokument z naslovom Proces izvedbe eksperimenta (Priloga B). Definirali smo motivacijo, objekt opazovanja, namen raziskave, vidike kakovosti in načrt samega eksperimenta. Orodji, ki smo ju potrebovali za izvedbo eksperimenta, sta bila prenosnik z internetno povezavo ter Google Chrome, saj smo s pomočjo Chrome DevTools zajeli posamezno sejo udeleženca eksperimenta ter zabeležili vse podatke, ki smo jih predhodno omenili v teoretičnem modelu. Od sodelujočih se ni pričakovalo predhodnega znanja posameznih mobilnih aplikacij. Je pa bilo glede na njihovo priljubljenost neizbežno predhodno poznanstvo in familiarnost. Eksperiment je bil razdeljen na dva dela; v prvem delu smo izvedli korake, predstavljene v prilogi (Priloga A), drugi del je predstavljala dejanska anketa. Anketo smo pripravili za vsako aplikacijo posebej (Facebook, Twitter, YouTube in GMail) ter za vsako različico posebej (npr. anketa za domorodno aplikacijo Facebook in anketa za spletno aplikacijo Facebook). Za beleženje in osnovno analizo anket in podatkov smo uporabili

Google Forms (Slika 5.2 in Slika 5.3), nadaljnjo analizo pa smo opravili v programu Excel in SPSS.



Analiza zmogljivosti mobilnih aplikacij temelječih na spletnih tehnologijah

Pozdravljeni!

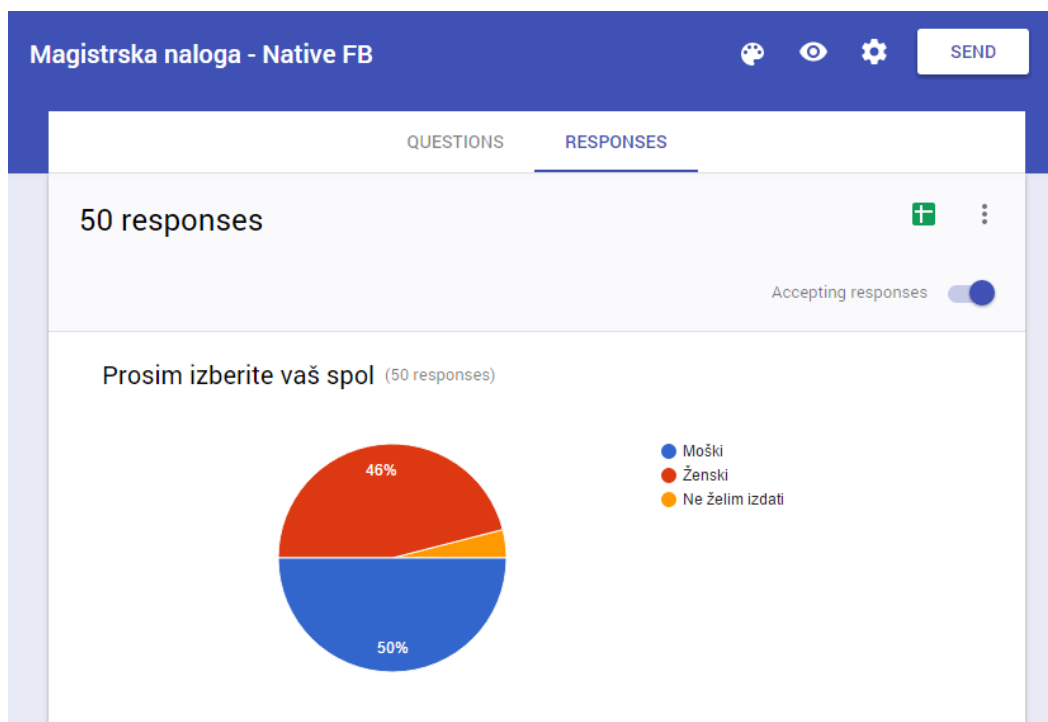
Sem Andreja Fajfar, študentka 2.letnika magistrskega programa informatike in tehnologij komuniciranja na FER-I-ju. V sklopu svojega raziskovalnega dela magistrske naloge Analiza zmogljivosti mobilnih aplikacij temelječih na spletnih tehnologijah izvajam anketo, s pomočjo katere želim pridobiti podatke o uporabniški izkušnji in zadovoljstvu ob uporabi tovrstnih rešitev.

Sodelovanje v anketi je anonimno, zbrani rezultati pa bodo uporabljeni samo za namen magistrske naloge. Reševanje vam bo vzelo le nekaj minut!

Za sodelovanje se vam najlepše zahvaljujem!

NEXT 25% complete

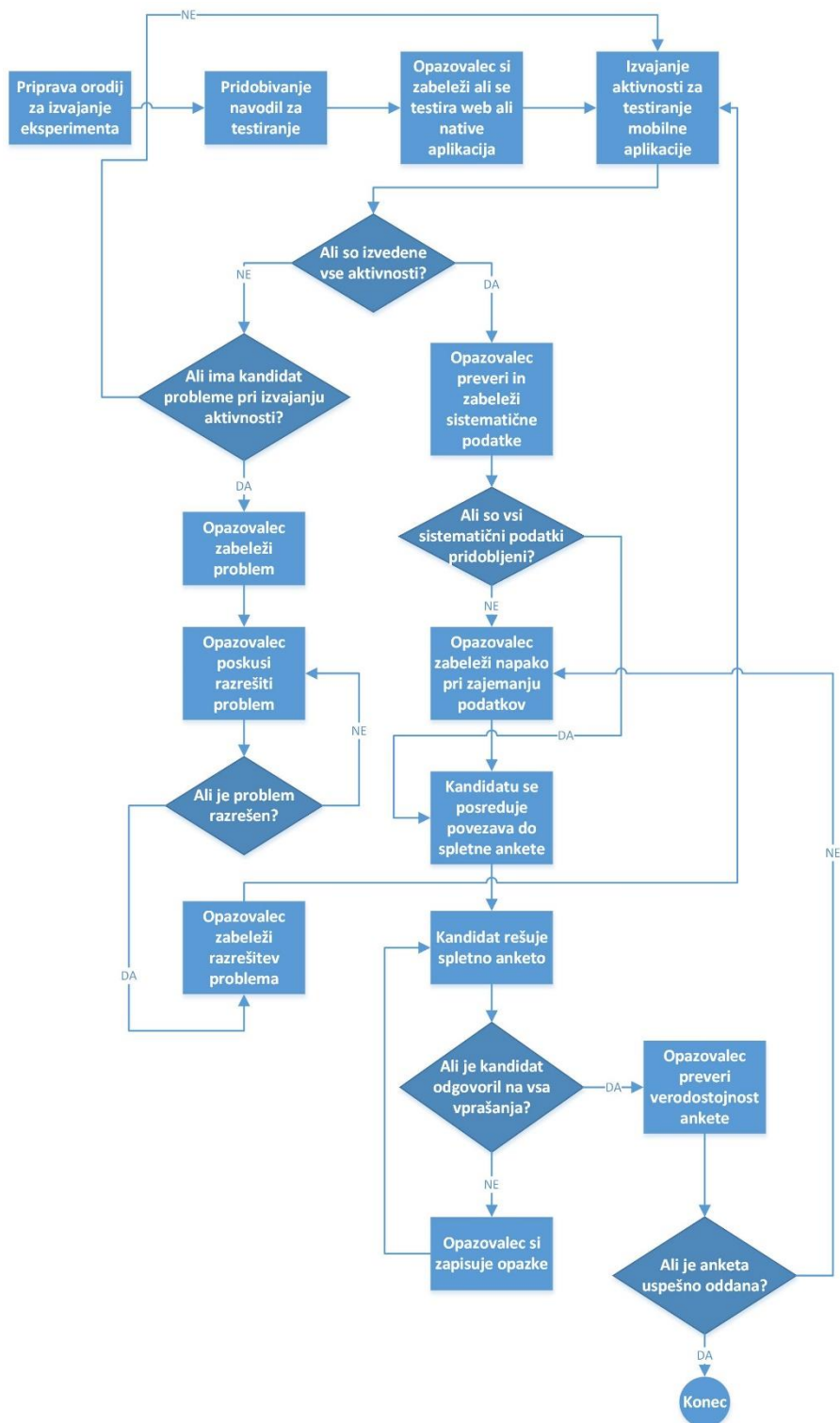
Slika 5.2: Uvodni del ankete za eksperiment



Slika 5.3: Analiza ankete po končanem eksperimentu

5.1 Izvedba eksperimenta

Eksperiment smo izvajali od 12. 10. 2015 do 29. 11. 2015. V tem času smo zabeležili 50 udeležencev, ki so uspešno izvedli predvidene aktivnosti v aplikacijah ter izpolnili ankete. Zaradi finančne omejitve smo uporabljali zgolj odprtokodne rešitve, zaradi česar nismo imeli orodja za zajemanje enakopravnih podatkov med spletnimi in domorodnimi aplikacijami. Ker smo lahko eksperiment zabeležili samo na spletnih aplikacijah, smo se odločili, da bomo zmogljivost med spletno in domorodno različico določili predvsem na podlagi uporabniškega zadovoljstva in izkušnje. Posamezno sejo udeleženca smo po zaključenem testiranju zabeležili v excelovo tabelo. Potek eksperimenta je razviden iz Slika 5.4.



Slika 5.4: Potek eksperimenta

Za vsakega udeleženca se je pripravila programska oprema (odjava iz aplikacije in spletni strani ter čiščenje nastalih piškotkov). Po pripravi so sledila navodila, ki jih je udeleženec

prejel v obliki seznama aktivnosti (Priloga A). Aktivnosti so se izvajale, skladno s seznamom, dokler udeleženec ni izvedel vseh ali pa je prišlo do nepopravljive napake v aplikaciji. V nobenem primeru kot opazovalec nismo sodelovali v eksperimentu z nasveti ali namigi. Kakršnekoli težave udeleženca je opazovalec zabeležil v excelovo tabelo. Po uspešno opravljenem prvem koraku eksperimenta se je udeleženca preusmerilo k anketi, medtem ko je opazovalec poskrbel za shranjevanje HAR in Timeline datotek iz Chrome DevTools orodja. Izvajanje eksperimenta in reševanje ankete se je ponovilo tako za domorodno kot tudi spletno različico. Na vsakih pet kandidatov se je vrstni red testiranja aplikacij zamenjal, če bi ocene spletne in domorodne različice bile odvisne od vrstnega reda izvajanja eksperimenta (ocene aplikacij so pokazale, da ni tako, saj so bile spletne aplikacije ocenjene slabše kot domorodne aplikacije ne glede na vrstni red izvajanja eksperimenta). Eksperiment je bil standardiziran, saj je udeleženec izvedel eksperiment ter šele nato anketo in nato ponovil proces za vse štiri aplikacije (Facebook, Twitter, YouTube in Gmail) za domorodno in spletno različico.

5.1.1 Omejitve eksperimenta

Eksperiment se je izvajal izključno na osebni prenosniku s konstantno internetno povezavo preko WI-FI protokola. Kot je prikazano v Tabela 5.1, se je pri tem uporabljal osebni prenosnik v kombinaciji z mobilnim telefonom. Udeleženci so testirali mobilne aplikacije direktno na Nexus 4 telefonu, medtem ko se je njihova seja shranjevala preko oddaljenega razhroščevanja v Chrome DevTools na prenosniku.

Tabela 5.1: Uporabljeni strojna in programska oprema pri eksperimentu

Prenosnik	Intel Core i7-3610QM @ 2.30Ghz 8 GB RAM Intel HD Graphics 4000, NVIDIA GeForce GT 650M 700GB HDD Intel Centrino Wireless-N 2230
Mobilni telefon	Nexus 4 Android version 5.1.1 Qualcomm APQ8064 Snapdragon S4 Pro Quad-core 1.5 GHz Krait Adreno 320 2GB RAM

Chrome brskalnik	Verzija 46.0.2490.71 - 46.0.2490.86
-------------------------	-------------------------------------

Za beleženje seje smo uporabljali zgolj Chrome DevTools, saj smo bili omejeni na odprtokodne rešitve. Posamezna seja je tako vsebovala HAR datoteko in časovnico aktivnosti udeleženca. Prav zato imamo podatke eksperimentov shranjene samo pri testiranju spletnih različic in ne domorodnih. Te omejitve so bile predčasno odobrene tudi s strani mentorja. Vsak udeleženec je lahko sodeloval pri eksperimentu samo enkrat, ponavljanje eksperimenta ni bilo mogoče. Pri tem smo omejili starost udeležencev, in sicer od 18. leta naprej.

5.1.2 Analiza veljavnosti in zanesljivosti rezultatov

Del vsake empirične raziskave je opredelitev omejitev in tveganj, ki so povezana z veljavnostjo rezultatov raziskave. Rezultate raziskave lahko posplošimo na celotno populacijo, saj smo uporabili naključen vzorec udeležencev (specifično nismo iskali udeležencev z izkušnjami, znanji, specifičnega spola ali starostne kategorije). Čeprav je vzorec eksperimenta majhen, menimo, da rezultati niso zanemarljivi in jih lahko upoštevamo. Vprašalnik je bil sestavljen iz vprašanj, ki so bila ovrednotena s pomočjo Likertove lestvice (se popolnoma ne strinjam – se popolnoma strinjam, zelo slabo –odlično). Za zagotavljanje večje veljavnosti konstruktov smo konstrukte operacionalizirali na podlagi indikatorjev. Podatke, pridobljene iz eksperimenta, lahko posplošimo na katerokoli vrsto spletnih mobilnih aplikacij, ni pa mogoče enakih podatkov izmeriti pri domorodnih mobilnih aplikacijah.

Za veljavnost konstruktov latentne spremenljivke smo poskrbeli s pomočjo analize z metriko Cronbach α , ki pove, kako dobro sodijo skupaj posamezni indikatorji za posamezno latentno spremenljivko. Metrika in analiza se uporabljata predvsem v primerih analize anket, kjer so odgovori ovrednoteni po Likertovi lestvici [51]. Pri določevanju notranje zanesljivosti smo upoštevali lestvico [80]:

- $\alpha \geq 0.9$: odlična notranja zanesljivost
- $0.9 > \alpha \geq 0.8$: dobra notranja zanesljivost
- $0.8 > \alpha \geq 0.7$: sprejemljiva notranja zanesljivost
- $0.7 > \alpha \geq 0.6$: vprašljiva notranja zanesljivost
- $0.6 > \alpha \geq 0.5$: slaba notranja zanesljivost
- $0.5 > \alpha$: nesprejemljiva notranja zanesljivost

Na podlagi predlagane lestvic so vrednosti nad 0.7 sprejemljive. Na podlagi zanesljivosti smo preverili štiri glavne latentne spremenljivke, in sicer kognitivno in afektivno zadovoljstvo ter uporabniško izkušnjo in zadovoljstvo. Rezultati SPSS izračunov so navedeni v tabelah Tabela 8.1, Tabela 8.2, Tabela 8.3, Tabela 8.4 in Tabela 8.5. Povzetek izračunov se nahaja v Tabela 5.2 in Tabela 5.3.

Tabela 5.2: Zanesljivost podatkov po Cronbach merilu

	Cronbach Alfa (α) vrednost	Stopnja notranje zanesljivosti	N- elementov	Tabele izračunov
Afektivno zadovoljstvo	.542	slaba	5	Tabela 8.4, Tabela 8.5
Kognitivno zadovoljstvo	.909	odlična	5	Tabela 8.1, Tabela 8.2
Uporabniška izkušnja	.900	odlična	7	Tabela 8.3
Uporabniško zadovoljstvo	.805	dobra	23	-

Tabela 5.3: Izračun zanesljivosti podatkov za afektivno zadovoljstvo

	Cronbach Alfa (α) vrednost	α v primeru brisanja indikatorja	Stopnja	Izbrisani indikatorji
Afektivno zadovoljstvo	.542	.703	sprejemljiva	Veliko število funkcionalnosti, dosegljivost in stabilnost
Uporabniško zadovoljstvo	.805	.819	dobra	

Po izračunih se je spremenljivka afektivno zadovoljstvo izkazala za spremenljivko s slabo notranjo zanesljivostjo, zato smo po opravljeni Cronbach analizi izbrisali določena indikatorja ter s tem povečali njeno notranjo zanesljivost (Tabela 5.3) – izračuni se nahajajo v Tabela 8.6 in Tabela 8.7. Spremembe vplivajo zgolj na izboljšanje notranje zanesljivosti afektivnega in uporabniškega zadovoljstva. Tabele nismo vključevali v primeru

uporabniškega zadovoljstva zgolj zaradi njihove obsežnosti, saj so vsebovali kar 21 posameznih konstruktov. Obrazložitev konstruktov uporabniškega zadovoljstva se nahaja v poglavju 5.3.9.

5.1.3 Vzorec eksperimenta

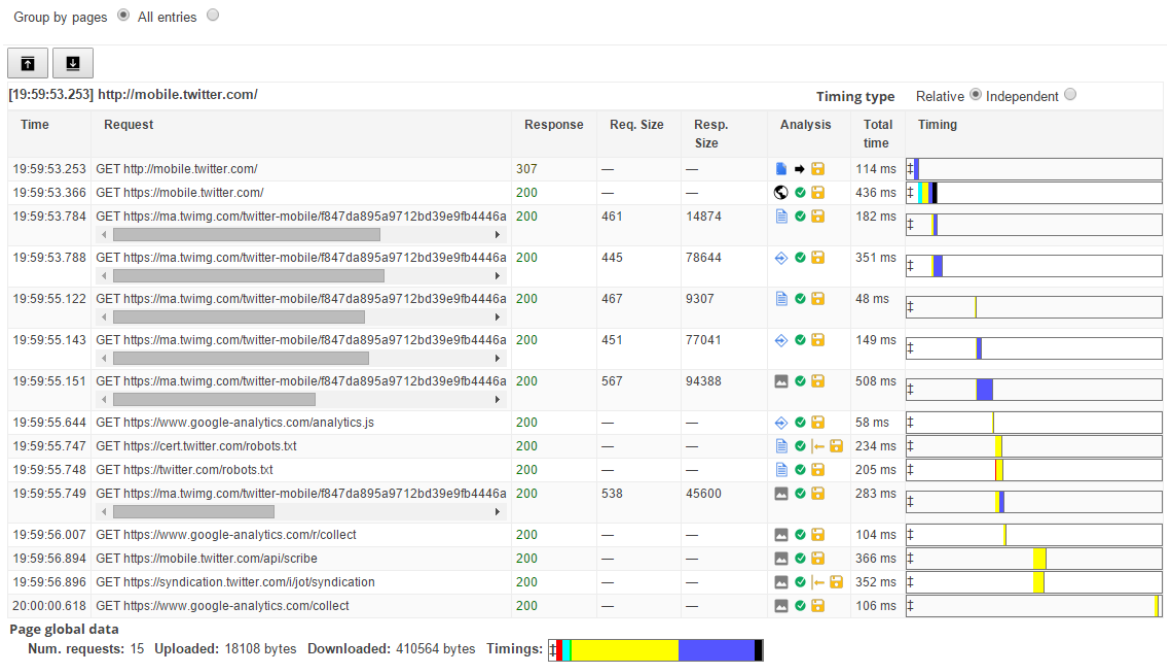
V eksperimentu je sodelovalo 50 udeležencev, od tega 26 moških (52 %) in 22 žensk (44 %) ter 2 udeleženca, ki nista želela podati svojega spola. Starost udeležence je bila kategorizirana v različne stopnje, in sicer 18–25 let (14 %), 26–35 let (58 %), 36–45 let (20 %), 46–55 let (6 %) in več kot 55 let (2 %). Vzorčenje je potekalo naključno, saj je lahko sodeloval kdorkoli ne glede na predhodno znanje uporabe ali poznavanje aplikacije. Edina omejitev je bila starost, torej udeleženec je moral biti starejši od 18 let. Pri posamezni aplikaciji smo udeležencem postavili vprašanje, koliko časa že uporabljajo testirano aplikacijo in kako pogosto jo uporabljajo. YouTube domorodno aplikacijo uporablja 76 % anketirancev več kot 3 leta, uporabljajo pa jo večkrat dnevno, tedensko in dnevno. YouTube spletno aplikacijo je 94 % anketirancev uporabljalo prvič, sicer uporabljajo aplikacijo večkrat dnevno in tedensko. Twitter domorodno aplikacijo uporablja 31,4 % anketirancev 1–3 leta, in sicer mesečno ter tedensko. Twitter spletno aplikacijo je 88 % anketirancev uporabljalo prvič, sicer pa jo uporabljajo mesečno in tedensko. Facebook domorodno aplikacijo uporablja 76 % več kot 3 leta večkrat dnevno, Facebook spletno aplikacijo pa je 96 % uporabljalo prvič, sicer pa jo uporabljajo večkrat dnevno. Gmail domorodno aplikacijo 84 % anketirancev uporablja že več kot 3 leta večkrat dnevno, Gmail spletno aplikacijo pa je 100 % anketirancev, torej vsi, uporabljalo sploh prvič.

Podatke, ki smo jih prejeli na podlagi eksperimenta, so bili razdeljeni v dve kategoriji (spletne in domorodne aplikacije), pri čemer smo prejeli 100 podatkov za vsako aplikacijo posebej (Facebook, Twitter, YouTube in Gmail), torej skupaj 400 podatkov.

5.2 Analiziranje HAR in Timeline datotek

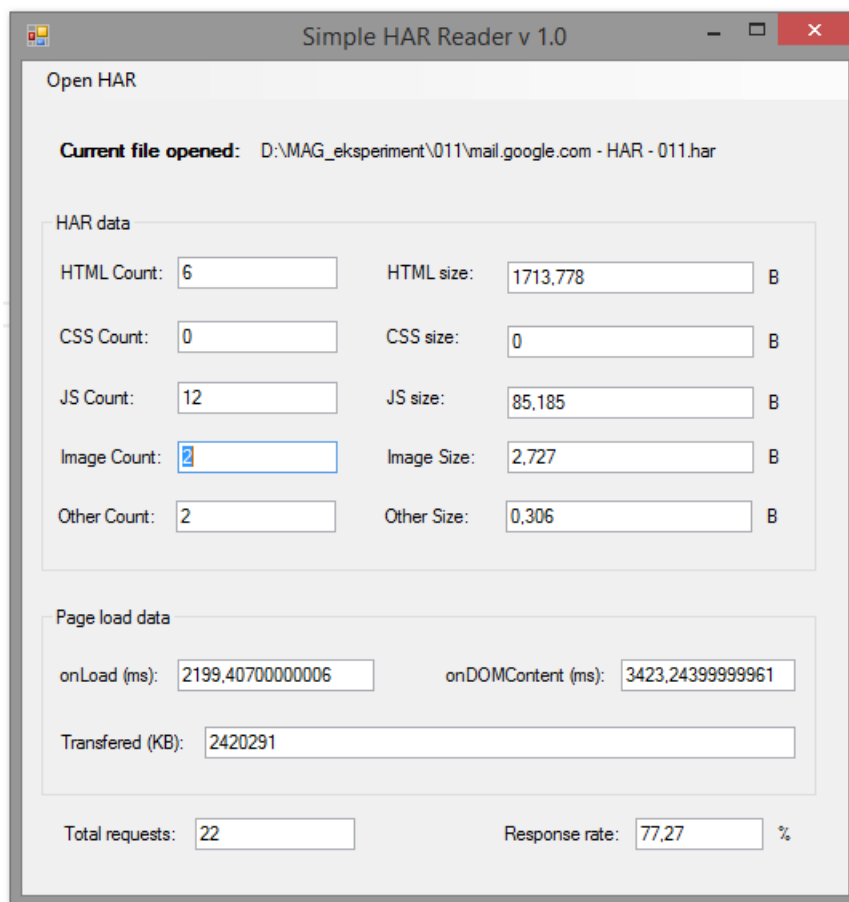
Z izvajanjem eksperimenta smo pridobili podatke iz praktičnega dela ter iz ankete. Med posamezno sejo testiranja smo le-to zabeležili s pomočjo Chrome DevTools ter shranili v HAR in Timeline datoteko. HAR oziroma HTTP Archive format datoteke so v bistvu JSON formatirane arhivske datoteke, ki shranjujejo interakcije med spletnim brskalnikom in stranjo [24]. V teh interakcijah lahko najdemo posamezne objekte, s pomočjo katerih določimo čas nalaganja, latentnost, čas prvotne inicializacije ter konsistentnost odzivnosti in dostopnost do strežnika. HAR datoteke povedo, točno kolikrat je bila uspešno vzpostavljena povezava med odjemalcem in strežnikom, koliko zahtev je bilo potrebnih, da se je spletna stran v

celoti naložila, ter čas odzivnosti oziroma hitrost spletne strani – v našem primeru smo pridobili podatke za mobilno aplikacijo, ki temelji na spletnih tehnologijah. HAR datoteke smo nato analizirali s pomočjo Google HAR Analyzer [30] v kombinaciji z orodjem, imenovanim Telerik Fiddler 4 [70].



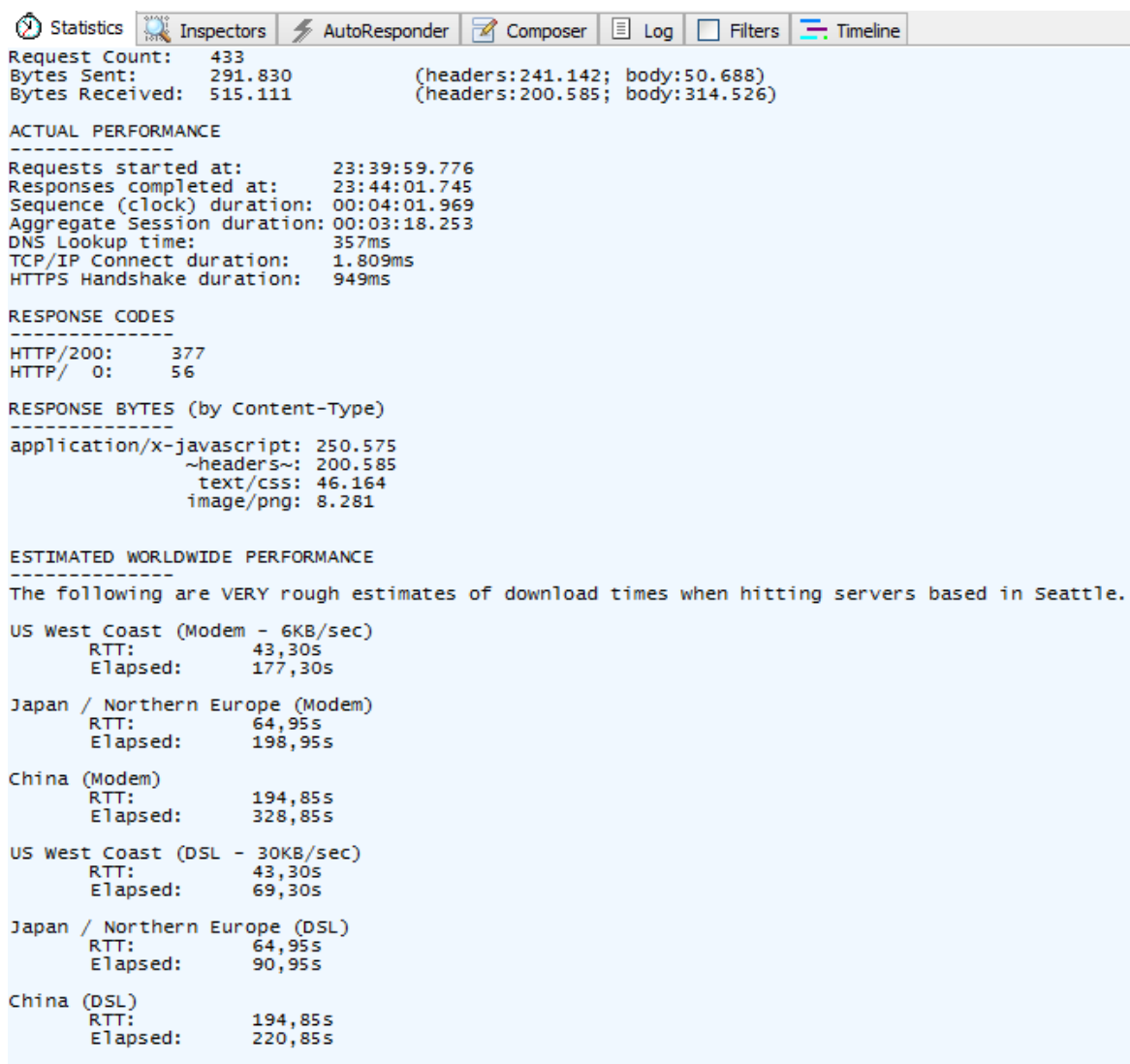
Slika 5.5: Analiza HAR datoteke eksperimenta

Iz Slika 5.5 je razvidno, da se informacije iz HAR datoteke v večini prikažejo kot padajoči slap. Posamezna zahteva je zabeležena z odzivno številko (v našem primeru smo upoštevali samo odzivno številko 200 kot uspešno zahtevo), velikost zahteve in odgovora ter posamezna analiza tipa zahteve in celoten čas, potreben za izvedbo. Iz tega prikaza smo dobili idejo, koliko časa je bilo potrebno za prvotno inicializacijo oziroma first byte, kot ga imenujemo. Pridobili smo tudi število in velikost zahteve ter število posameznih datotek, ki so se prenesle v naši seji. Zaradi obširnosti HAR datotek smo kmalu ugotovili, da potrebujemo orodja za razbiranje podatkov. Ustvarili smo preprost HAR bralec na osnovi C# (Slika 5.6). Ideja programa je, da prebere HAR datoteko ter kot izhodne podatke vrne število datotek (HTML, CSS, JS, slike in ostalo), velikost teh datotek, potreben čas za nalaganje strani in vseh elementov strani ter velikost prenesenih podatkov, število zahtev in uspešnost odzivnosti strežnika. Izpisovali smo samo podatke, ki smo jih pozneje dejansko potrebovali za SPSS statistično obdelavo.



Slika 5.6: Primer obdelave HAR datoteke z avtorskim programom

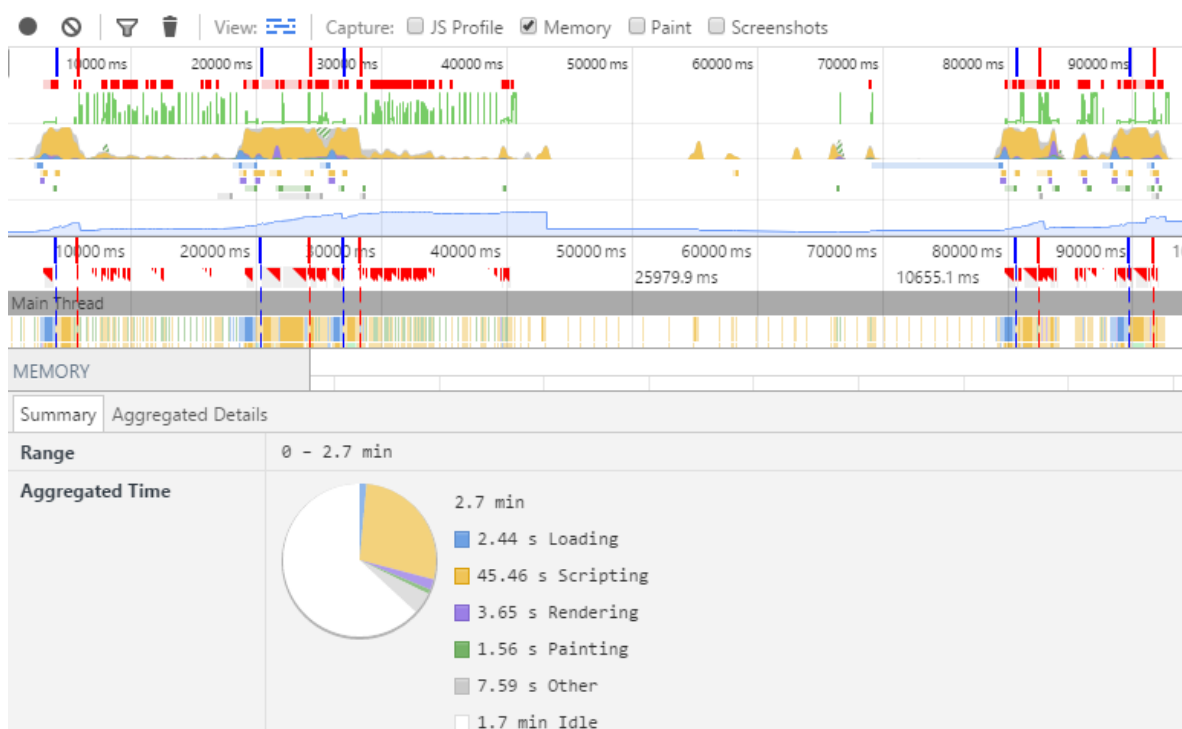
V kombinaciji z našim programom smo uporabljali še program Fiddler, ki je odlično orodje za prebiranje HAR datotek, iz katerih smo nadknadno dobili število odzivov in različne manjkajoče čase. Tako smo s pomočjo Fiddler orodja natančno videli, koliko bitov smo poslali in prejeli, kako aktivna je bila posamezna seja (dejanski prenos podatkov med odjemalcem in strežniškim delom – *aggregate session duration*) in kako dolga je bila posamezna seja skupaj z neaktivnimi in aktivnimi deli (*sequence clock duration*). Upoštevali smo drug podatek, saj je bila aplikacija aktivna, čeprav se med uporabnikom in strežnikom v ozadju trenutno ni nič izvajalo oziroma pošiljajo. Slika 5.7 prikazuje prebrane podatke iz testne HAR datoteke, kjer je razvidno število zahtev, odzivni časi, število odgovorov ter aktivni in neaktivni deli seje.



Slika 5.7: Pridobivanje podatkov s pomočjo Fiddler orodja

Drugi del shranjenih podatkov iz eksperimenta smo lahko pregledovali s pomočjo časovnice znotraj Chrome DevTools orodja. Časovnica (ang. timeline) predstavlja posnetek seje, kjer lahko analiziramo vse aktivnosti, ki se izvajajo, medtem ko uporabljamo aplikacijo [16]. V našem primeru je to pomenilo, da smo snemali posamezne seje udeležencev, medtem ko so le-ti izvajali v naprej določene korake v mobilnih aplikacijah. Med snemanjem se je vsak dogodek, do katerega je prišlo, zapisal in vsak tak posnetek dogodka se lahko klasificira v štiri osnovne kategorije: nalaganje dogodkov (ang. loading events), izvajanje skripte (ang. scripting), obdelava podatkov (ang. rendering) in prikazovanje podatkov oz. izrisovanje (ang. painting). Ob tem smo upoštevali še čas za izvajanje ostalih aktivnosti (ang. other) in koliko časa aplikacija ni izvajala dogodkov (ang. idle time). Vse omenjene čase smo združili v kategorijo, s katero smo merili čas latentnosti aplikacije. S pomočjo časovnice lahko

natančneje analiziramo, kje v naši aplikaciji prihaja do ozkih grl, kje se določene skripte izvajajo predolgo in katere datoteke potrebujejo preveč časa za uspešno prikazovanje (Slika 5.8).



Slika 5.8: Pregled časovnice v Chrome DevTools orodju

Vsaka seja se vizualno prikaže najprej kot graf (Slika 5.8) ter kot celotna aktivnost, razdeljena na podaktivnosti (Slika 5.8). Grafi prikazujejo vizualne konice v delovanju aplikacije, medtem ko spodnje informacije natančno sporočijo uporabniku, koliko časa je bilo porabljenega za posamezno aktivnost znotraj obdelave zahteve.

5.3 Analiza podatkov

Statistično proučevanje podatkov zajema vse podatke, ki smo jih pridobili na podlagi opravljenega eksperimenta in analize. Pridobljene podatke smo v orodju SPSS testirali zoper normalne porazdeljenosti. Ker podatki, ki jih potrebujemo za dokazovanje hipotez, niso izkazali normalne porazdeljenosti, smo se odločili za izvedbo neparametričnih testov, tj. za Mann-Whitneyjev U-test [52]. Preverili smo, ali posamezne spremenljivke izpolnjujejo štiri poglobitve predpogoje za izvedbo Mann-Whitneyjevega testa: ali so spremenljivke ovrednotene s pomočjo Likertove lestvice, ali neodvisna spremenljivka sestoji iz dveh kategorij (spletna in domorodna aplikacija), ali smo v eksperimentu sodelovali zgolj kot

opazovalec ter ali so podatki normalno porazdeljeni [52]. Določene spremenljivke smo tudi transformirali iz niza v številčne kategorije in obratno. Končni rezultat združevanja spremenljivk iz eksperimenta in ankete je bila SPSS datoteka s 73 spremenljivkami.

5.3.1 Analiza podatkov mobilne aplikacije Twitter

V povprečju so udeleženci za izvedbo eksperimenta na spletni različici aplikacije Twitter potrebovali 2,6 minute in 2,2 minuti na domorodni. Pogosto zabeležena težava je povezana z uporabniškim vmesnikom, ki je otežil izpis iz aplikacije tako iz spletne kot tudi iz domorodne različice. O drugih težavah udeleženci niso poročali. Grafi so zaradi svoje obširnosti priloženi v prilogi (Priloga C). Podatki, povzeti iz analiz, so prikazani v Tabela 5.4. Iz 50 zabeleženih primerov izvajanja eksperimenta je jasno razvidno (Slika 8.1), da je posamezna zahteva sestavljena iz določenega števila HTML, CSS, JS, slik in ostalih dokumentov, pri čemer očitno prevladuje število slik na posamezno zahtevo. V povprečju znašajo slike kar 75 % celotne zahteve (Slika 8.2). Optimiziranost zahtev vodi v boljšo zmogljivost same aplikacije, vendar nastane težava, kadar obstaja več kot očitni preskok v velikosti med slikami in ostalimi dokumenti v zahtevi (Slika 8.3). Večja, kot je zahteva, več časa potrebuje aplikacija za njeno obdelavo. V povprečju se je izvedlo 200–300 posamičnih zahtev (izračunana povprečna vrednost znaša 263) na sejo udeleženca. Povprečna velikost tovrstne seje je 0,105 MB (Slika 8.4).

Iz Slika 8.5 in Slika 8.6 je razvidno, da je inicializacijski čas mobilne aplikacije Twitter, temelječe na spletnih tehnologijah, sprejemljiv. Pri tem smo uporabili lestvico, kjer je veljalo, da je inicializacijski čas < 1 sekunde odličen, 1–2,5 sekunde sprejemljiv in kar je > 2,5 sekunde slab. Twitter spletna aplikacija ima torej sprejemljiv inicializacijski čas, kar pomeni, da je v večini primerov izkazala primerno odzivnost na zahtevo.

Aplikacija ni optimizirana, saj se veliko časa (kar 78 %) porabi za čakanje (Slika 8.7). Druge aktivnosti vključujejo nalaganje (1 %, ang. loading), čakanje (78 %, ang. idle), izrisovanje (1 %, ang. painting), obdelavo (2 %, ang. rendering), izvajanje skripte (13 %, ang. scripting) in ostalo (5 %, ang. other). V kriterijih dostopnosti (Slika 8.8) in konsistentnosti odziva (Slika 8.9) se je Twitter odrezal odlično. V povprečju je bila aplikacija dostopna kar 98,96 % in pri tem izkazala konsistenco v odzivnosti povprečne vrednosti 3 minute s standardno deviacijo le 1 minuto in 11 sekund.

Kriterij dostopnosti se je meril na podlagi uspešnih zahtev s strani uporabnika (torej zahteve s statusom 200). Kakršnekoli druge zahteve (404, 500, 301) so se štete za neuspešno obdelane zahteve s strani aplikacije. Ker je hitrost aplikacije odvisna od časa nalaganja in

od časa, ki je potreben za obdelavo posameznega segmenta zahteve, (scripting, loading, itd.), ne smemo ignorirati onLoad in DOMContentLoaded izmerjenega časa (Slika 8.10). Oba časa sta namreč tesno povezana z JavaScript obdelavo – onLoad dogodek se sproži, kadar so uspešno naloženi vsi elementi strani, medtem ko se DOMContentLoaded dogodek sproži, ko se DOM hierarhija uspešno strukturira. Priporočljivi časi za tovrstne JavaScript dogodke so: odlično < 2 sekundi, sprejemljivo 2–5 sekund in slabo pri > 5 sekund. Povprečne vrednosti onLoad dogodka (5,788 s) in DOMContentLoaded (6,27 s) nakazujejo na počasno nalaganje spletne aplikacije.

Na podlagi ankete smo razbrali afektivno in kognitivno zadovoljstvo (Slika 8.11). Obe skupini sta pripomogli, da smo Twitter mobilni aplikaciji, temelječi na spletnih tehnologijah, določili uporabniško izkušnjo v višini 44 % (nezadostno) in uporabniško zadovoljstvo le pičlih 41,73 % (nezadostno). Ob tem so enaki subjekti izkazali večje zadovoljstvo (78,60 %) in boljšo izkušnjo (87,78 %) pri domorodni različici. Udeleženci so domorodno aplikacijo opisali kot uporabno (84,3 %), visoke kakovosti (52,9 %) in zanesljivo (45,1 %). Pri spletni različici prevladujejo opisi, kot so uporabna (62 %), enostavna (60 %) in slabe kakovosti (24 %). Spletne različice 96 % udeležencev ne bi priporočalo naprej. Domorodno različico pa bi priporočalo 74,5 % udeležencev.

Tabela 5.4: Analiza podatkov Twitter aplikacije

Tip vsebine		Sestavljenost zahteve		Kompleksnost vsebine (B) - povprečja		Čas inicializacije	
Št. HTML	11	HTML	4 %	Velikost HTML kode	177	2,02 s	
Št. CSS	7	CSS	3 %	Velikost CSS kode	395	Kategorizirani čas inicializacije	
Št. JavaScript	10	JS	4 %	Velikost JS kode	1843	Odličen	22 %
Št. slik	197	Slike	75 %	Velikost slik	4824	Sprejemljiv	50 %
Št. ostalo	38	Ostalo	14 %	Povprečno št. zahtev	263	Slab	28 %

Povprečen čas aktivnosti za obdelavo zahteve		Dostopnost aplikacije	Čas nalaganja	
Nalaganje	1 %	98,96 %	onLoad	DOMContentLoaded
Čakanje	78 %	Konsistentnost odzivnosti	5,787 s	6,268 s
Izrisovanje	1 %	Povprečna vrednost	Uporabniško zadovoljstvo	
Obdelovanje	2 %	3 min 0 s	Domorodna različica	Spletna različica
Izvajanje skripte	13 %	Stand. deviacija	78,60 %	41,73 %
Ostalo	5 %	1 min 11 s	Uporabniška izkušnja	
Čas izvedbe eksperimenta			Domorodna različica	Spletna različica
Domorodna različica		Spletna različica	87,78 %	44 %
2,2 min		2,6 min		

5.3.2 Analiza podatkov mobilne aplikacije Facebook

V povprečju so udeleženci potrebovali 3,2 minuti za izvedbo eksperimenta na spletni različici aplikacije Facebook in 2,9 minut na domorodni. Pogosto zabeleženi težavi sta povezani z neodzivno tipkovnico v domorodni različici ter s prenehanjem delovanja spletne aplikacije ob izbiri počutja in lokacije. O drugih težavah udeleženci niso poročali. Grafi so priloženi v prilogi (Priloga C). Tabela 5.5 predstavlja povzetek vseh omenjenih podatkov.

Tabela 5.5: Analiza podatkov Facebook aplikacije

Tip vsebine		Sestavljenost zahteve		Kompleksnost vsebine (B) - povprečja		Čas inicializacije	
Št. HTML	13	HTML	2 %	Velikost HTML kode	602	1,09 s	
Št. CSS	102	CSS	22 %	Velikost CSS kode	1767	Kategorizirani čas inicializacije	
Št. JavaScript	129	JS	28 %	Velikost JS kode	7052	Odličen	56 %

Tip vsebine		Sestavljenost zahteve		Kompleksnost vsebine (B) - povprečja		Kategorizirani čas inicializacije	
Št. slik	203	Slike	44 %	Velikost slik	4037	Sprejemljiv	44 %
Št. ostalo	17	Ostalo	4 %	Povprečno št. zahtev	464	Slab	0 %
Povprečen čas aktivnosti za obdelavo zahteve		Dostopnost aplikacije		Čas nalaganja			
Nalaganje	1 %	98,08 %		onLoad		DOMContentLoaded	
Čakanje	67 %	Konsistentnost odzivnosti		30,014 s		16,118 s	
Izrisovanje	2 %	Povprečna vrednost		Uporabniško zadovoljstvo			
Obdelovanje	3 %	4 min 24 s		Domorodna različica		Spletna različica	
Izvajanje skripte	22 %	Stand. deviacija		90,4 %		18,4 %	
Ostalo	5 %	1 min 23 s		Uporabniška izkušnja			
Čas izvedbe eksperimenta				Domorodna različica		Spletna različica	
Domorodna različica		Spletna različica		92 %		33 %	
2,9 min		3,2 min					

Iz 50 zabeleženih primerov izvajanja eksperimenta je razvidno (Slika 8.12), da je posamezna zahteva sestavljena iz HTML, CSS, JS, slik in ostalih dokumentov, pri čemer prevladujejo slike glede na število dokumentov pri posamezni zahtevi. V povprečju slike predstavljajo 44 % sestavnega dela vsake zahteve (Slika 8.13). Kompleksnost vsebine je odvisna od velikosti posameznega tipa dokumenta znotraj zahteve (HTML, CSS, JS in slike). Čeprav slike prevladujejo po številu, je JavaScript glavni faktor, ki vpliva na velikost zahteve in posamezne strani (Slika 8.14). V povprečju se je izvedlo 400–500 posameznih zahtev (izračunano povprečje je 463) na sejo udeleženca. Povprečna velikost seje je znašala 0,454 MB (Slika 8.15).

Slika 8.16 in Slika 8.17 prikazujeta inicializacijski čas Facebook aplikacije, ki je odličen. Pri tem smo ponovno uporabili lestvico, kjer je veljalo, da je inicializacijski čas < 1 sekunda odličen, 1–2,5 sekunde sprejemljiv in kar je > 2,5 sekundi slab. Aplikacija ni optimizirana,

saj se 67 % časa porabi za čakanje dogodkov in izvajanje funkcij (Slika 8.18). V kriterijih dostopnosti (Slika 8.19) in konsistentnosti odziva (Slika 8.20) se je aplikacija Facebook odzvala odlično. V povprečju je bila aplikacija dostopna 98,08 % in pri tem izkazala konsistenco v odzivnosti povprečne vrednosti 4 minute s standardno deviacijo le 1 minuto in 23 sekund.

Kriterij dostopnosti se je meril na podlagi uspešnih zahtev s strani uporabnika. Uspešne zahteve so bile zahteve s statusom 200, preostale so se štete za neuspešne (404, 500 in 301). Ker je hitrost aplikacije odvisna od časa nalaganja in od časa, ki je potreben za obdelavo posameznega segmenta zahteve, ne smemo ignorirati onLoad in DOMContentLoaded časa (Slika 8.22). onLoad JavaScript dogodek se sproži, ko so uspešno naloženi vsi elementi strani, kar pojasnuje tudi večji čas nalaganja v primerjavi z DOMContentLoaded dogodkom. Priporočljivi čas za tovrstne JavaScript dogodke je manj kot 5 sekund. Povprečne vrednosti onLoad dogodka (30,01 s) in DOMContentLoaded (16,11 s) nakazujejo na počasno nalaganje Facebook aplikacije, do česar pride zlasti zaradi obširne obdelave JavaScript skript in slik v zahtevah.

Na podlagi ankete smo lahko razbrali afektivno in kognitivno zadovoljstvo (Slika 8.21). Obe skupini sta pripomogli, da smo Facebook spletni aplikaciji določili uporabniško izkušnjo v višini 33 % (nezadostno) in uporabniško zadovoljstvo le pičlih 18,4 % (nezadostno). Kognitivno zadovoljstvo predstavlja mišljenje udeleženca, afektivno prikaže čustveni odziv. Ob tem so enaki udeleženci izkazali večje zadovoljstvo (90,4 %) in boljšo uporabniško izkušnjo (92 %) pri domorodni različici aplikacije.

Udeleženci so domorodno aplikacijo opisali kot uporabno (80 %), visoke kakovosti (72 %) in zanesljivo (44 %). Pri spletni različici prevladujejo opisi nepraktična (56 %), slabe kakovosti (40%) in enostavna (30 %) ter uporabna (30 %). Kar 64 % udeležencev spletne različice ne bi priporočalo sodelavcem in znancem, medtem ko domorodno bi (78 %).

5.3.3 Analiza podatkov mobilne aplikacije YouTube

50 udeležencev je za izvedbo eksperimenta na spletni različici YouTube aplikacije v povprečju potrebovalo 2,27 minut in 2,28 minut na domorodni. Pogosto zabeleženi težavi sta povezani z izpisom iz aplikacije ter z neodzivnostjo spletne različice aplikacije. O drugih težavah udeleženci niso poročali. Grafi so priloženi v prilogi (Priloga C), vsi omenjeni analizirani podatki pa se nahajajo v Tabela 5.6.

Slika 8.23 prikazuje, da je posamezna zahteva sestavljena iz določenega števila HTML, CSS, JS, slik in ostalih dokumentov, pri čemer prevladujejo slike s 34 % in ostali dokumenti s 30 % (Slika 8.24). Pogost rezultat poskusa optimizacije aplikacije je razviden na Slika 8.25, kjer v velikosti posamezne zahteve prevladujejo JavaScript skriptne datoteke. V povprečju se je izvedlo 100–200 posamičnih zahtev (171,52) na sejo udeleženca. Povprečna velikost tovrstne seje je 0,077 MB (Slika 8.26).

Slika 8.27 in Slika 8.28 prikazujeta, da je inicializacijski čas mobilne aplikacije YouTube, temelječe na spletnih tehnologijah, sprejemljiv. Pri tem smo uporabili lestvico, kjer velja, da je inicializacijski čas < 1 sekundo odličen, 1–2,5 sekunde sprejemljiv in kar je > 2,5 sekunde slab. Sprejemljiv inicializacijski čas v povprečju nakazuje na primerno odzivnost aplikacije. Aplikacija ni optimizirana, saj se 69,99 % časa porabi za čakanje (Slika 8.29). V kriterijih dostopnosti (Slika 8.30) in konsistentnosti odziva (Slika 8.31) se je YouTube odrezal sprejemljivo. V povprečju je bila aplikacija dostopna 80 % in pri tem izkazala konsistenco v odzivnosti povprečne vrednosti 2 minut s standardno deviacijo 1 minuta in dve sekundi.

Tabela 5.6: Analiza podatkov YouTube aplikacije

Tip vsebine		Sestavljenost zahteve		Kompleksnost vsebine (B) – povprečja		Čas inicializacije	
Št. HTML	39	HTML	20 %	Velikost HTML kode	745	1,031 s	
Št. CSS	7	CSS	3 %	Velikost CSS kode	168	Kategorizirani čas inicializacije	
Št. JavaScript	24	JS	13 %	Velikost JS kode	1959	Odličen	54 %
Št. slik	68	Slike	34 %	Velikost slik	921	Sprejemljiv	46 %
Št. ostalo	58	Ostalo	30 %	Povprečno št. zahtev	171	Slab	0 %

Povprečen čas aktivnosti za obdelavo zahteve		Dostopnost aplikacije	Čas nalaganja	
Nalaganje	0,6 %	80 %	onLoad	DOMContentLoaded
Čakanje	70 %	Konsistentnost odzivnosti	3,857 s	3,478 s
Izrisovanje	3,5 %	Povprečna vrednost	Uporabniško zadovoljstvo	
Obdelovanje	3,6 %	2 min 30 s	Domorodna različica	Spletna različica
Izvajanje skripte	13,6 %	Stand. deviacija	95,6 %	34,2 %
Ostalo	8,8 %	1 min 2 s	Uporabniška izkušnja	
Čas izvedbe eksperimenta			Domorodna različica	Spletna različica
Domorodna različica		Spletna različica	92 %	55 %
2,28 min		2,7 min		

Kriterij dostopnosti se je meril na podlagi uspešnih in neuspešnih zahtev. Čas nalaganja (Slika 8.32) je odziv zmogljivosti aplikacije. Zato se izpostavita onLoad in DOMContentLoaded dogodka. Ob tem smo izmerili povprečno vrednost onLoad dogodka 3,85 sekund in DOMContentLoaded dogodka 3,48 sekund. Podana lestvica, kjer so izmerjeni časi 2–5 sekund kategorizirani kot sprejemljivi, označi YouTube spletno aplikacijo kot sprejemljivo hitro glede na čas nalaganja.

Na podlagi kognitivnega in afektivnega zadovoljstva (Slika 8.33) smo določili uporabniško izkušnjo in uporabniško zadovoljstvo. YouTube spletna aplikacija se je izkazala z uporabniško izkušnjo v višini 55 % (nezadostno) in uporabniškim zadovoljstvom 34,2 % (nezadostno). Domorodna različica je izkazala večje zadovoljstvo (95,6 %) in uporabniško izkušnjo (92 %). Udeleženci so domorodno aplikacijo opisali kot uporabno (84 %), visoke kakovosti (68 %) in zanesljivo (50 %). Pri spletni različici prevladujejo opisi, kot so uporabna (62 %), enostavna (42 %) in zanesljiva (22 %). Spletne različice udeleženci ne bi priporočali naprej (76 %), domorodno bi (88 % udeležencev).

5.3.4 Analiza podatkov mobilne aplikacije Gmail

V povprečju so udeleženci za izvedbo eksperimenta na spletni različici aplikacije Gmail potrebovali 1,9 minut in le 1,7 minut na domorodni. Pogosto zabeležena težava je povezana

s funkcionalnostjo domorodne aplikacije, kjer so udeleženci pogrešali možnost direktnega dodajanja slik s pomočjo kamere. O drugih težavah udeleženci niso poročali. Grafi so priloženi v prilogi (Priloga C), vsi omenjeni analizirani podatki so v Tabela 5.7.

Tabela 5.7: Analiza podatkov GMail aplikacije

Tip vsebine		Sestavljenost zahteve		Kompleksnost vsebine (B)		Čas inicializacije	
Št. HTML	6	HTML	27 %	Velikost HTML kode	1817	1,267 s	
Št. CSS	0	CSS	0 %	Velikost CSS kode	0	Kategorizirani čas inicializacije	
Št. JavaScript	13	JS	53 %	Velikost JS kode	91	Odličen	44 %
Št. slik	3	Slike	11 %	Velikost slik	6	Sprejemljiv	56 %
Št. ostalo	2	Ostalo	9 %	Povprečno št. zahtev	21	Slab	0 %
Povprečen čas aktivnosti za obdelavo zahteve		Dostopnost aplikacije		Čas nalaganja			
Nalaganje	0,3 %	77,43%		onLoad		DOMContentLoaded	
Čakanje	20,9 %	Konsistentnost odzivnosti		2,651 s		3,394 s	
Izrisovanje	74,4 %	Povprečna vrednost		Uporabniško zadovoljstvo			
Obdelovanje	0,68 %	1 min 51 s		Domorodna različica		Spletna različica	
Izvajanje skripte	2,54 %	Stand. deviacija		95,6 %		24,2 %	
Ostalo	1 %	1 min 10 s		Uporabniška izkušnja			
Čas izvedbe eksperimenta				Domorodna različica		Spletna različica	
Domorodna različica		Spletna različica		98 %		30 %	
1,7 min		1,9 min					

Slika 8.34 in Slika 8.35 prikazujeta, da zahteve ne vsebujejo CSS datotek. Prevladujejo JavaScript skripte s kar 53 %. Večji del zahteve je sestavljena iz HTML dokumentov, kar

prikazuje Slika 8.36. V povprečju se je izvedlo 20–30 posamičnih zahtev (izračunano povprečje je 24) na sejo udeleženca. Povprečna velikost seje je 2,41 MB (Slika 8.37).

Iz Slika 8.38 in Slika 8.39 je razvidno, da je inicializacijski čas spletne mobilne aplikacije Gmail sprejemljiv. Pri tem smo uporabili lestvico, kjer velja, da je inicializacijski čas < 1 sekunda odličen, 1–2.5 sekund sprejemljiv in kar je > 2.5 sekund slab.

Iz vseh testiranih aplikacij (Facebook, Twitter, YouTube in Gmail) izkazuje spletna različica Gmail najboljšo optimizacijo, saj se večino časa porabi za izrisovanje rezultatov obdelave (74,44 %) in ne za čakanje (20,96 %), kar je razvidno na Slika 8.40. V kriterijih dostopnosti (Slika 8.41) in konsistentnosti odziva (Slika 8.42) se je Gmail odrezal sprejemljivo. V povprečju je bila aplikacija dostopna v 77,43 % primerih in pri tem izkazala konsistenco v odzivnosti povprečne vrednosti 1 minute in 51 sekund s standardno deviacijo le 1 minuto in 10 sekund.

Kriterij dostopnosti se je meril na podlagi uspešnih in neuspešnih zahtev s strani uporabnika. Hitrost aplikacije je odvisna od časa nalaganja ter od časa, ki je potreben za obdelavo posameznega segmenta zahteve (Slika 8.43). Pri času nalaganja izpostavimo dogodka onLoad in DOMContentLoaded. Na podlagi priporočljivih časov (odličen, če je pod dvema sekundama, sprejemljiv, kadar je čas izvedbe od dve do pet sekund, ter počasen, če je več kot 5 sekund) lahko določimo sprejemljiv čas nalaganja za spletno aplikacijo Gmail – onLoad dogodek v povprečju potrebuje 2,65 sekundi in DOMContentLoaded dogodek 3,39 sekundi.

Analiza ankete (Slika 8.44) je pokazala, da so udeleženci ocenili spletno mobilno aplikacijo Gmail z uporabniško izkušnjo v višini 30 % (nezadostno) in uporabniško zadovoljstvo le s 24,2 % (nezadostno). Ob tem so enaki udeleženci izkazali večje zadovoljstvo (95,6 %) in izkušnjo (98 %). Udeleženci so domorodno aplikacijo opisali kot uporabno (84 %), visoke kakovosti (76 %) in zanesljivo (62 %). Pri spletni različici prevladujejo opisi, kot so nepraktična (56 %), slabe kakovosti (46 %) in enostavna (36 %). Spletne različice udeleženci ne bi priporočali sorodnikom in sodelavcem (94 %), domorodno različico bi (100 %).

5.3.5 Analiza podatkov za dokazovanje enostavnosti uporabe, kakovosti, odzivnosti in vizualne izdelave aplikacije

Za dokazovanje omenjenih kriterijev smo uporabili spremenljivke iz ankete ter njihove vrednosti primerjali med spletno različico in domorodno različico posameznih aplikacij.

Tabela 5.8: Deskriptivna statistika za aplikacijo Facebook

	N	Mean	Median	Stand. deviacija	Minimum	Maksimum
Kakovost aplikacije	100	3,36	3,00	1,307	1	5
Odzivnost aplikacije	100	3,26	3,00	1,276	1	5
Enostavnost uporabe aplikacije	100	3,09	3,00	1,326	1	5
Vizualna izdelava aplikacije	100	3,01	3,00	1,605	1	5

Prvi rezultat deskriptivne analize iz Tabela 5.8 prikaže število podatkov znotraj posamezne spremenljivke (N = 100) ter sredinsko vrednost in standardno deviacijo za kakovost (3,36), odzivnost (3,26), enostavnost uporabe (3,09) in vizualno izdelavo (3,01).

Tabela 5.9: Mann-Whitneyjev U-test za aplikacijo Facebook

	Tip aplikacije	N	Mean rank	Vsota rankov
Kakovost aplikacije	Spletna aplikacija	50	27,38	1369,00
	Domorodna aplikacija	50	73,62	3681,00
Odzivnost aplikacije	Spletna aplikacija	50	26,96	1348,00
	Domorodna aplikacija	50	74,04	3702,00
Enostavnost uporabe aplikacije	Spletna aplikacija	50	26,86	1343,00
	Domorodna aplikacija	50	74,14	3707,00
Vizualna izdelava aplikacije	Spletna aplikacija	50	26,29	1314,50
	Domorodna aplikacija	50	74,71	3735,50

Tabela 5.10: Statistični rezultati za aplikacijo Facebook

	Kakovost aplikacije	Odzivnost aplikacije	Enostavnost uporabe aplikacije	Vizualna izdelava aplikacije
Mann-Whitneyjev U-test	94,000	73,000	68,000	39,500

	Kakovost aplikacije	Odzivnost aplikacije	Enostavnost uporabe aplikacije	Vizualna izdelava aplikacije
Wilcoxon W	1369,000	1348,000	1343,000	1314,500
Z	-8,228	-8,340	-8,373	-8,578
Asymp. Sig. (2-tailed)	,000	,000	,000	,000
r	-0,823	-0,834	-0,837	-0,858

Primarno je iz Tabela 5.9 razvidno, da prihaja do očitnih razlik med spletno in domorodno različico Facebook aplikacije, pri čemer so ocene za spletno različico nižje kot za domorodno. Statistični rezultati (Tabela 5.10) nakazujejo, da je signifikanca prisotna pri vseh štirih testiranih spremenljivkah, saj je manjša kot 0,05. Uporabimo enačbo (5.1) [26]:

$$r = z/\sqrt{N} \quad (5.1)$$

kjer velja:

z – z vrednost iz tabele Mann-Whitneyjevega statističnega izračuna,

N – število elementov.

Na podlagi enačbe dobimo nove vrednosti efekta za spremenljivke kakovost aplikacije (-0,822796273), odzivnost aplikacije (-0,834033085), enostavnost uporabe (-0,837255047) in vizualna izdelava (-0,85775518), kjer s pomočjo Cohen lestvice [20] določimo njihovo efektivno velikost: $-0,1 < 0,3$ je vrednost mala, $0,3 < 0,5$ je srednje velika in vse nad 0,5 je velika vrednost. Efektivna velikost pove, da bo poljubna vrednost, vzeta iz kategorije »domorodna aplikacija«, verjetno večja kot poljubna vrednost iz kategorije »spletna aplikacija«. To samo potrди, da so vrednosti za domorodno aplikacijo večje, torej so bolje ocenjene.

Tabela 5.11: Deskriptivna statistika testa za aplikacijo GMail

	N	Mean	Median	Stand. deviacija	Minimum	Maksimum
Kakovost aplikacije	100	3,50	4,00	1,467	1	5
Odzivnost aplikacije	100	3,36	4,00	1,425	1	5
Enostavnost uporabe aplikacije	100	3,30	4,00	1,481	1	5
Vizualna izdelava aplikacije	100	3,15	3,50	1,666	1	5

Povprečne vrednosti in standardna deviacija za posamezne spremenljivke v kategoriji GMail aplikacije v Tabela 5.11 se ne razlikujejo med seboj oz. v primerjavi z ostalimi aplikacijami (Facebook, Twitter in YouTube). Posamezne vrednosti so višje v primerjavi z Facebook aplikacijo: kakovost (3,5), odzivnost (3,36), enostavnost uporabe (3,3) in vizualna izdelava (3,15). Tabela 5.12 natančno prikazuje, da je tudi pri primerjavi domorodne in spletne aplikacije GMail boljše ocene prejela domorodna aplikacija.

Tabela 5.12: Mann-Whitneyjev U-test za aplikacijo GMail

	Tip aplikacije	N	Mean rank	Vsota rankov
Kakovost aplikacije	Spletna aplikacija	50	26,05	1302,50
	Domorodna aplikacija	50	74,95	3747,50
Odzivnost aplikacije	Spletna aplikacija	50	26,71	1335,50
	Domorodna aplikacija	50	74,29	3714,50
Enostavnost uporabe aplikacije	Spletna aplikacija	50	26,95	1347,50
	Domorodna aplikacija	50	74,05	3702,50
Vizualna izdelava aplikacije	Spletna aplikacija	50	26,00	1300,00
	Domorodna aplikacija	50	75,00	3750,00

Tabela 5.13: Statistični rezultati za aplikacijo GMail

	Kakovost aplikacije	Odzivnost aplikacije	Enostavnost uporabe aplikacije	Vizualna izdelava aplikacije
Mann-Whitney U	27,500	60,500	72,500	25,000
Wilcoxon W	1302,500	1335,500	1347,500	1300,000
Z	-8,772	-8,427	-8,352	-8,748
Asymp. Sig. (2-tailed)	,000	,000	,000	,000
r	-0,877	-0,843	-0,835	-0,875

Tabela 5.13 nakazuje, da je signifikanca med spremeljivkami prisotna ($p < 0.05$). S pomočjo Cohen lestvice [20] lahko določimo velikost efekta, in sicer izračunane vrednosti so kakovost aplikacije (-0,877232952), odzivnost aplikacije (-0,842701055), enostavnost uporabe (-0,835187329) in vizualna izdelava (-0,874845031). Posledično te številke pomenijo, da bo poljubna vrednost, vzeta iz kategorije »domorodna aplikacija«, verjetno večja kot poljubna vrednost iz kategorije »spletna aplikacija«. Navedeno potrdi, da so vrednosti za domorodno aplikacijo večje, torej so bolje ocenjene.

Tabela 5.14: Deskriptivna statistika testa za aplikacijo Twitter

	N	Mean	Median	Stand. deviacija	Minimum	Maksimum
Kakovost aplikacije	100	3,52	4,00	,979	1	5
Odzivnost aplikacije	100	3,55	4,00	,968	1	5
Enostavnost uporabe aplikacije	100	3,30	3,00	1,059	1	5
Vizualna izdelava aplikacije	100	3,38	3,00	1,170	1	5

Podatki iz Tabela 5.14 natančno nakazujejo povprečno vrednost in standardno deviacijo posameznih spremenljivk: kakovost aplikacije (3,52), odzivnost (3,55), enostavnost uporabe (3,3) in vizualna izdelava (3,38).

Tabela 5.15: Mann-Whitneyjev U-test za aplikacijo Twitter

	Tip aplikacije	N	Mean rank	Vsota rankov
Kakovost aplikacije	Spletna aplikacija	50	33,72	1686,00
	Domorodna aplikacija	50	67,28	3364,00
Odzivnost aplikacije	Spletna aplikacija	50	33,34	1667,00
	Domorodna aplikacija	50	67,66	3383,00
Enostavnost uporabe aplikacije	Spletna aplikacija	50	36,16	1808,00
	Domorodna aplikacija	50	64,84	3242,00
Vizualna izdelava aplikacije	Spletna aplikacija	50	32,03	1601,50
	Domorodna aplikacija	50	68,97	3448,50

Tabela 5.16: Statistični rezultati za aplikacijo Twitter.

	Kakovost aplikacije	Odzivnost aplikacije	Enostavnost uporabe aplikacije	Vizualna izdelava aplikacije
Mann-Whitney U	411,000	392,000	533,000	326,500
Wilcoxon W	1686,000	1667,000	1808,000	1601,500
Z	-6,086	-6,231	-5,143	-6,565
Asymp. Sig. (2-tailed)	,000	,000	,000	,000
r	-0,609	-0,623	-0,514	-0,656

Iz Tabela 5.15 je razvidno, da so ocene nižje v primerjavi spletne različice in domorodne. Statistični rezultati iz Tabela 5.16 nakazujejo na signifikanco med spremenljivkami (< 0.05). Na podlagi teh podatkov lahko izračunamo tudi efekt velikosti za posamezne spremenljivke: kakovost aplikacije (-0,608573477), odzivnost aplikacije (-0,623117368), enostavnost

uporabe (-0,514279192) in vizualna izdelava (-0,656472437). V vsakem primeru je efekt velikosti velik (torej večji od 0,5).

Tabela 5.17: Deskriptivna statistika testa za aplikacijo YouTube

	N	Mean	Median	Stand. deviacija	Minimum	Maksimum
Kakovost aplikacije	100	3,72	4,00	1,111	1	5
Odzivnost aplikacije	100	3,64	4,00	1,133	1	5
Enostavnost uporabe aplikacije	100	3,54	4,00	1,086	1	5
Vizualna izdelava aplikacije	100	3,75	4,00	1,048	1	5

Tabela 5.17 prikazuje povprečne in standardno deviacijo posameznih spremenljivk: kakovost aplikacije (3,72), odzivnost (3,64), enostavnost uporabe (3,54) in vizualna izdelava (3,75).

Tabela 5.18: Mann-Whitneyjev U-test za aplikacijo YouTube

	Tip aplikacije	N	Mean rank	Vsota rankov
Kakovost aplikacije	Spletna aplikacija	50	30,00	1500,00
	Domorodna aplikacija	50	71,00	3550,00
Odzivnost aplikacije	Spletna aplikacija	50	31,19	1559,50
	Domorodna aplikacija	50	69,81	3490,50
Enostavnost uporabe aplikacije	Spletna aplikacija	50	31,22	1561,00
	Domorodna aplikacija	50	69,78	3489,00
Vizualna izdelava aplikacije	Spletna aplikacija	50	32,34	1617,00
	Domorodna aplikacija	50	68,66	3433,00

Mann-Whitneyjev U-test nakazuje, da prihaja do razlik v ocenah med posameznimi spremenljivkami, kar je razvidno iz Tabela 5.18. S pomočjo enačbe za velikost efekta in

Tabela 5.19 izračunamo posamezne vrednosti efekta: kakovost aplikacije (-0,737398796), odzivnost aplikacije (-0,692936094), enostavnost uporabe (-0,693895574) in vizualna izdelava (-0,651562608). Velikost efekta je velika, kar podpre dejstvo, da so spremenljivke ocenjene boljše oziroma višje v primeru domorodne aplikacije kot v primeru spletne različice za YouTube aplikacijo.

Tabela 5.19: Statistični rezultati za aplikacijo YouTube

	Kakovost aplikacije	Odzivnost aplikacije	Enostavnost uporabe aplikacije	Vizualna izdelava aplikacije
Mann-Whitney U	225,000	284,500	286,000	342,000
Wilcoxon W	1500,000	1559,500	1561,000	1617,000
Z	-7,374	-6,929	-6,939	-6,516
Asymp. Sig. (2-tailed)	,000	,000	,000	,000
r	-0,737	-0,693	-0,694	-0,652

5.3.6 Analiza uporabniškega zadovoljstva

V SPSS smo izračunali uporabniško zadovoljstvo ob uporabi posamezne aplikacije (Facebook, Twitter, YouTube in Gmail) tako za spletno kot tudi za domorodno različico.

Tabela 5.20: Izračun povprečnih vrednosti uporabniškega zadovoljstva

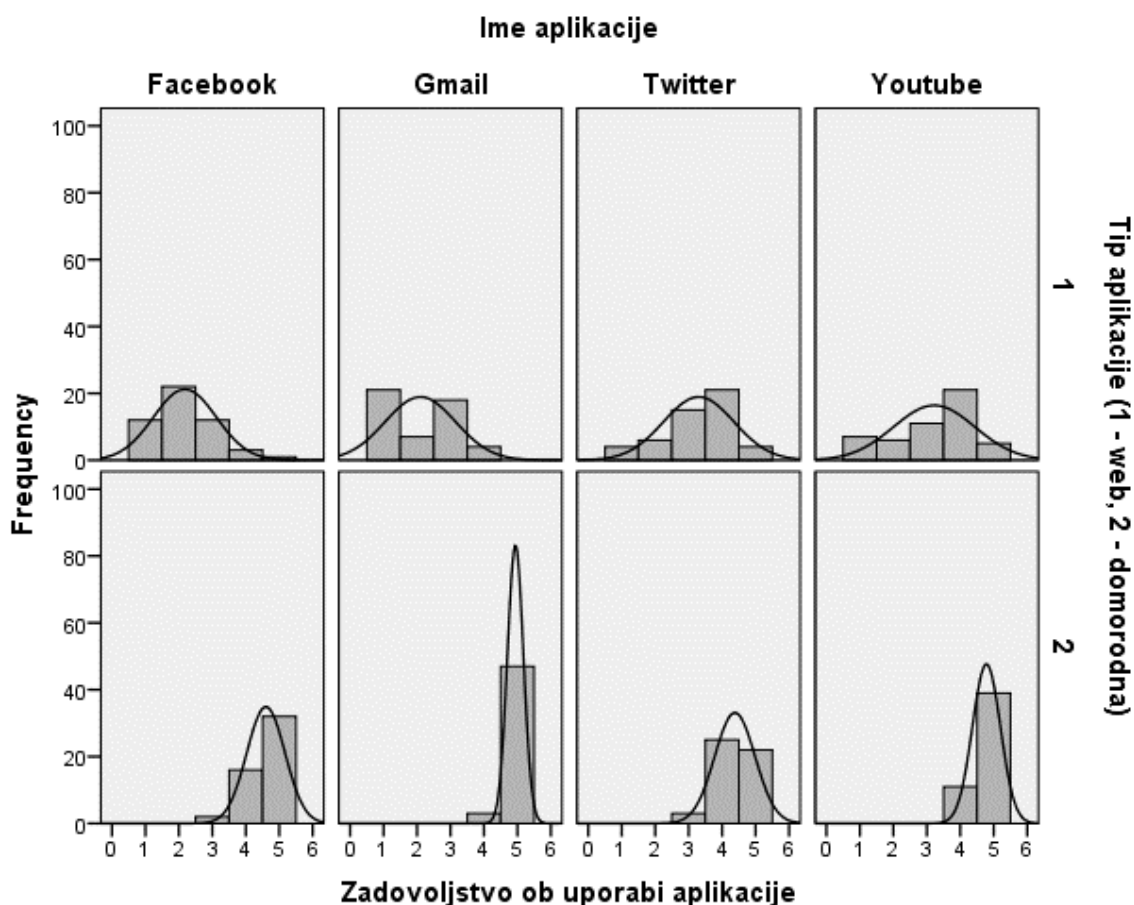
	Tip aplikacije	N	Mean rank	Seštevek rankov
Zadovoljstvo ob uporabi aplikacije	Spletna	200	115,66	23132,50
	Domorodna	200	285,34	57067,50

Tabela 5.21: Statistični izračuni za uporabniško zadovoljstvo

Statistični izračuni	
	Zadovoljstvo ob uporabi aplikacije
Mann-Whitney U	3032,500
Wilcoxon W	23132,500

Z	-15,270
Asymp. Sig. (2-tailed)	,000

Čeprav smo uporabniško zadovoljstvo določili predhodno na podlagi kognitivnih in afektivnih vprašanj iz ankete, lahko te podatke potrdimo tudi s pomočjo SPSS. Osredotočili smo se na spremenljivko »zadovoljstvo ob uporabi aplikacije«, kjer so udeleženci ocenili svojo zadovoljstvo na podlagi 1–5 Likertove lestvice. Iz statističnih rezultatov iz Tabela 5.20 in Tabela 5.21 lahko razberemo standardno deviacijo in povprečno vrednost ocene uporabniškega zadovoljstva za spletno in domorodno različico.



Slika 5.9: Graf prikazovanja uporabniškega zadovoljstva med posameznimi tipi aplikacij

Slika 5.9 prikazuje ocene uporabniškega zadovoljstva ob uporabi posamezne aplikacije (Facebook, Gmail, Twitter in YouTube) v domorodni in spletni različici. Ocene in s tem tudi uporabniško zadovoljstvo so višji v primeru domorodnih aplikacij, kjer so odgovori pri spletnem tipu aplikacije s poudarkom na manjši oceni skorajda normalno porazdeljeni.

5.3.7 Analiza korelacije med številom zahtev in časom nalaganja (onLoad in DOMContentLoaded) pri spletnih mobilnih aplikacijah

Za dokazovanje korelacije med spremenljivkama število zahtev in čas nalaganja smo uporabili Pearson korelacijski test [44]. Korelacijo spremenljivke število zahtev smo primerjali z onLoad in DOMContentLoaded časom nalaganja.

Tabela 5.22: Korelacija med številom zahtev in onLoad časom nalaganja

		Čas nalaganja: onLoad	Število zahtev
Čas nalaganja: onLoad	Pearson Correlation	1	,813**
	Sig. (2-tailed)		,000
	N	200	200
Število zahtev	Pearson Correlation	,813**	1
	Sig. (2-tailed)	,000	
	N	200	200

Tabela 5.23: Korelacija med številom zahtev in DOMContentLoaded časom nalaganja

		Število zahtev	Čas nalaganja: DOMContentLoaded
Število zahtev	Pearson Correlation	1	,837**
	Sig. (2-tailed)		,000
	N	200	200
Čas nalaganja: DOMContentLoaded	Pearson Correlation	,837**	1
	Sig. (2-tailed)	,000	
	N	200	200

Iz Tabela 5.22 in Tabela 5.23 razberemo, da obstaja korelacija v velikosti 0,813, kar pomeni, da signifikanca $p < .001$ obstaja za vseh 200 opazovanih primerov v analizi. Tudi tabela 5.18 poroča o korelaciji, saj je korelacija .837 in s tem potrjuje signifikanco $p < .001$.

Pearson korelacijska formula (5.2) [44] se glasi:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{var(x)} \cdot \sqrt{var(y)}} \quad (5.2)$$

kjer velja:

$cov(x,y)$ – verjetnost, da se dve naključni spremenljivki skupaj spreminjata,

$var(x)$ – primerek variacije spremenljivke x ,

$var(y)$ – primerek variacije spremenljivke y .

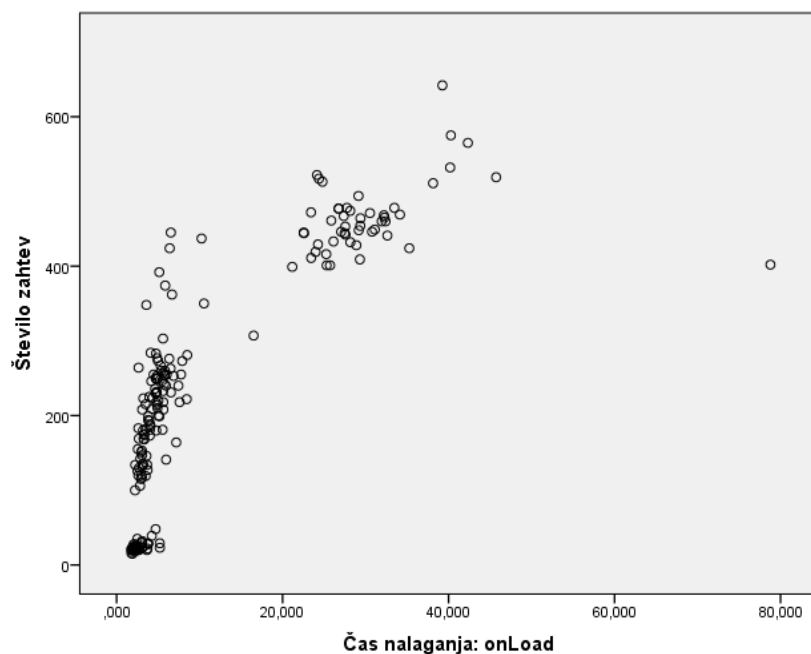
SPSS samostojno izračuna omenjene vrednosti, kjer smo s tabelo določili moč korelacije [44]:

- $.1 < |r| < .3$ ponazarja majhno/šibko korelacijo
- $.3 < |r| < .5$ ponazarja srednje veliko korelacijo
- $.5 < |r|$ ponazarja veliko/močno korelacijo

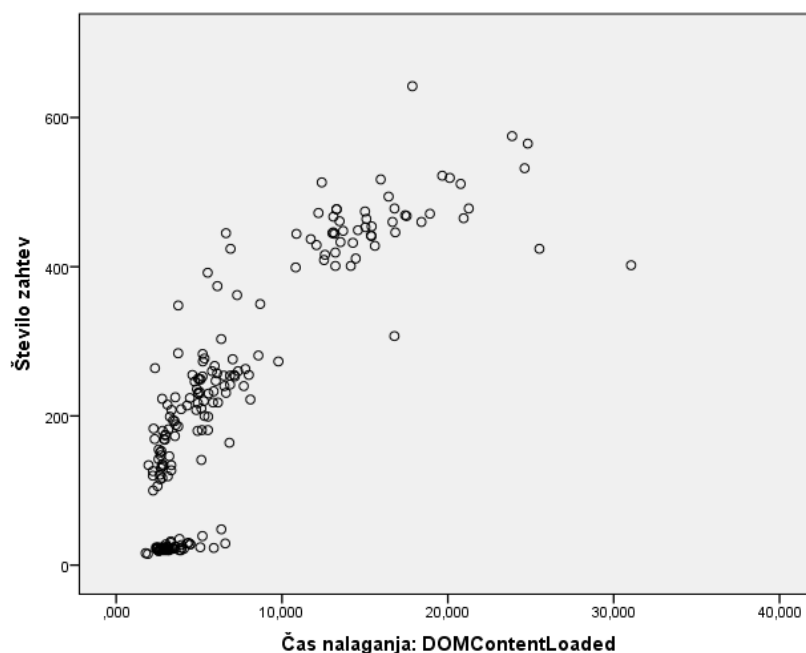
Tako lahko na podlagi teh rezultatov potrdimo:

- Število zahtev in čas nalaganja (onLoad in DOMContentLoaded) imata statistično signifikanten odnos ($p < .001$).
- Smer odnosa je pozitivna (spremenljivki sta v pozitivni korelaciji, kar nakazuje Pearson korelacijski rezultat .813 in .837). To pomeni, da vrednost spremenljivk narašča v korelaciji – večje, kot je število zahtev, daljši je čas nalaganja.
- Moč korelacije je močna.

Slika 5.10 in Slika 5.11 nakazujeta linearno korelacijo med spremenljivkami, kar je predpogoj za izvajanje Pearson korelacijskega testa. Čeprav prihaja do podatkov, ki ne sovpadajo z imaginarno linearno črto (ang. outliers), je le-teh malo in jih lahko v naši analizi zanemarimo.



Slika 5.10: Število zahtev v korelaciji z onLoad časom nalaganja



Slika 5.11: Število zahtev v korelaciji z DOMContentLoaded časom nalaganja

5.3.8 Analiza zmogljivosti mobilnih aplikacij temelječih na spletnih tehnologijah

Zmogljivost je definirana kot sposobnost obdelave določene količine podatkov v določenem času. To neposredno vpliva tudi na uporabniško izkušnjo in zadovoljstvo. Glavni del hitrosti

spletne aplikacije sestavljajo tri osnovne karakteristike [32]: število zahtev, velikost strani in čas nalaganja strani. Na osnovi virov [10][33][85][54] smo definirali 5-stopenjsko lestvico za ocenjevanje karakteristik, in sicer od nezadostno do odlično. Na podlagi HTTPArchive [33] trendov, povzetih dne 1. 4. 2016, smo pridobili povprečne vrednosti karakteristik, ki so služile kot orientacijska sredinska vrednost »dobro«.

Lestvica za obdelavo število zahtev (n = število zahtev, povprečna vrednost = 68 [32]):

- Nezadostno je $n > 200$ zahtev.
- Zadostno je $100 < n < 200$ zahtev.
- Dobro je $60 < n < 10$.
- Prav dobro je $45 < n < 6$.
- Odlično je $n < 45$.

Podatki za čas nalaganja so osnovani na KISSMetrics [85] analizi časa nalaganja in kdaj zazna uporabnik čas nalaganja kot zamudnega in opusti dejavnost. Na podlagi teh podatkov v kombinaciji z HTTPArchive [33] trendov smo postavili lestvico za obdelavo časa nalaganja (m = onLoad čas nalaganja, povprečna vrednost = 6,7 sekund [32]):

- Nezadostno je $m > 10$ sekund.
- Zadostno je $7 < m < 10$ sekund.
- Dobro $3 < m < 7$ sekund.
- Prav dobro $1 < m < 3$ sekund.
- Odlično $0,1 < m < 1$ sekunda.

Karkoli se obdeluje v času od 0,1 do 1 sekunde se kategorizira kot instantna obdelava podatkov oziroma brezhibni tok misli [54]. Maksimalna vrednost 10 sekund je dokazano mejnik, kjer uporabnik nima več potrpežljivosti ter opusti dejavnost oziroma aplikacijo. Karkoli se izvaja v času 1–10 sekund, uporabnik sicer zazna kot zamudno, vendar še vedno prevladuje občuteknadzora nad situacijo, zato smo čas nalaganja pod 10 sekund označili kot zadosten [54]. Tretja lestvica predstavlja velikost strani, saj so spletne mobilne aplikacije optimizirane spletne strani za mobilnike in torej sestavljene iz posameznih spletnih strani. Dokazano je, da velikost spletnih strani narašča iz leta v leto [10]. Samo v preteklem letu 2015 je velikost v povprečju narasla za 16 % [10]. Na podlagi tega je predlagana lestvica osnovana za testirane mobilne aplikacije v 2016. Lestvica za velikost strani (s = velikost strani v kB, povprečna vrednost = 1299 kB [33]):

- Nezadostno je $s > 5000$ kB.

- Zadostno je $2200 \text{ kB} < s < 5000 \text{ kB}$.
- Dobro je $1200 \text{ kB} < s < 2200 \text{ kB}$.
- Prav dobro je $500 \text{ kB} < s < 1200 \text{ kB}$.
- Odlično je $s < 500 \text{ kB}$.

V SPSS smo opravili frekvenčno analizo, na podlagi katere smo opredelili oceno za vse tri karakteristike: število zahtev, čas nalaganja strani in velikost strani. Na podlagi rezultatov frekvenčne analize lahko skupaj v kombinaciji z oceno uporabniške izkušnje (*ux* – ang. user experience) in zadovoljstva (*us* – ang. user satisfaction) postavimo skupno oceno zmogljivosti mobilne aplikacije, temelječe na spletnih tehnologijah. Skupna ocena zmogljivosti (5.3) se glasi:

$$\sum \text{zmogljivost} = \frac{n+m+s+ex+us}{5} \quad (5.3)$$

kjer velja:

n – število zahtev,

m – čas nalaganja strani,

s – velikost strani,

ux – uporabniška izkušnja,

us – uporabniško zadovoljstvo.

Ob tem upoštevamo postavljene lestvice, kjer se znakovne vrednosti pretvorijo v številčne po naslednjem pravilu:

- nezadostno = 1, zadostno = 2, dobro = 3, prav dobro = 4, odlično = 5.

Tudi za uporabniško izkušnjo in zadovoljstvo se uporabi enaka lestvice, le da se odstotki pretvorijo v celoštevilčne vrednosti po lestvici:

- 0–45 % = nezadostno = 1
- 46–59,9 % = zadostno = 2
- 60–74,9 % = dobro = 3
- 75–89,9 % = prav dobro = 4
- 90–100 % = odlično = 5

Rezultati analize zmogljivosti so predstavljeni v skupinah za vsako aplikacijo posebej (Facebook, Twitter, YouTube in GMail).

Tabela 5.24: Frekvenčna analiza Facebook aplikacije

N = 50		Frekvenca	Odstotek
Velikost strani (page size)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	0	0 %
	zadostno	0	0 %
	nezadostno	50	100 %
Čas nalaganja strani (page load)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	1	2% %
	zadostno	0	0 %
	nezadostno	49	98 %
Število zahtev (number of requests)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	0	0 %
	zadostno	0	0 %
	nezadostno	50	100 %

Če povzamemo rezultate opravljene frekvenčne analize (Tabela 5.24) in jih vstavimo v našo enačbo:

$$\sum \text{zmogljivost} = \frac{\text{nezadostno} + \text{nezadostno} + \text{nezadostno} + \text{nezadostno} + \text{nezadostno}}{5} = \frac{5}{5} = 1.$$

Iz tega je razvidno, da je skupna ocenjena zmogljivosti Facebook spletne aplikacije na podlagi kriterijev velikost strani, čas nalaganja strani, število zahtev, uporabniškega zadovoljstva in izkušenj nezadostna.

Tabela 5.25: Frekvenčna analiza GMail aplikacije

N = 50		Frekvenca	Odstotek
Velikost strani (page size)	odlično	0	0 %
	prav dobro	0	0 %

		Frekvenca	Odstotek
	dobro	47	94 %
	zadostno	3	6 %
	nezadostno	0	0 %
Čas nalaganja strani (page load)	odlično	0	0 %
	prav dobro	35	70 %
	dobro	15	30 %
	zadostno	0	0 %
	nezadostno	0	0 %
Število zahtev (number of requests)	odlično	49	98 %
	prav dobro	1	2 %
	dobro	0	0 %
	zadostno	0	0 %
	nezadostno	0	0 %

Tabela 5.25 poda rezultate glede ocen posameznih karakteristik zmogljivosti mobilne aplikacije GMail. Iz tega lahko ponovno izračunamo skupno zmogljivost:

$$\sum \text{zmogljivost} = \frac{\text{dobro} + \text{prav dobro} + \text{odlično} + \text{nezadostno} + \text{nezadostno}}{5} = \frac{14}{5} = 3.$$

Iz tega je razvidno, da je celostna zmogljivost spletne mobilne aplikacije GMail ocenjena z dobro na podlagi števila zahtev, časa nalaganja, velikosti strani, uporabniške izkušnje in zadovoljstva.

Tabela 5.26: Frekvenčna analiza Twitter aplikacije

N = 50		Frekvenca	Odstotek
Velikost strani (page size)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	0	0 %
	zadostno	4	8 %
	nezadostno	46	92 %

N = 50		Frekvenca	Odstotek
Čas nalaganja strani (page load)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	44	88 %
	zadostno	4	8 %
	nezadostno	2	4 %
Število zahtev (number of requests)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	0	0 %
	zadostno	3	6 %
	nezadostno	47	94 %

Tabela 5.26 nakazuje ocene posameznih karakteristik: velikost strani (nezadostno), čas nalaganja (dobro) in število zahtev (nezadostno). S pomočjo enačbe izračunamo zmogljivost:

$$\sum \text{zmogljivost} = \frac{\text{nezadostno} + \text{dobro} + \text{nezadostno} + \text{nezadostno} + \text{nezadostno}}{5} = \frac{7}{5} = 1,4 = 1.$$

Dokazali smo, da je zmogljivost v celoti nezadostna.

Tabela 5.27: Frekvenčna analiza YouTube aplikacije

N = 50		Frekvenca	Odstotek
Velikost strani (page size)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	2	4 %
	zadostno	43	86 %
	nezadostno	5	10 %
Čas nalaganja strani (page load)	odlično	0	0 %
	prav dobro	14	28 %
	dobro	32	64 %
	zadostno	3	6 %
	nezadostno	1	2 %

N = 50		Frekvenca	Odstotek
Število zahtev (number of requests)	odlično	0	0 %
	prav dobro	0	0 %
	dobro	1	2 %
	zadostno	38	76 %
	nezadostno	11	22 %

Tabela 5.27 nakazuje ocene posameznih karakteristik mobilne aplikacije za YouTube aplikacijo, in sicer se povzame večinska ocena. Iz tega določimo oceno velikosti strani (zadostno), časa nalaganja strani (dobro) in števila zahtev (zadostno). Skupno oceno zmogljivosti ponovno izračunamo na podlagi podane enačbe.

$$\sum \text{zmogljivost} = \frac{\text{zadostno} + \text{dobro} + \text{zadostno} + \text{zadostna} + \text{nezadostno}}{5} = \frac{10}{5} = 2$$

Tabela 5.28 prikazuje povzetek izračunov zmogljivosti na osnovi frekvenčne analize posameznih aplikacij.

Tabela 5.28: Povzetek ocen frekvenčne analize vseh aplikacij

Facebook	Twitter	GMail	YouTube
nezadostno	nezadostno	dobro	zadostno

5.3.9 Analiza podatkov za uporabniško zadovoljstvo in uporabniško izkušnjo

Poglavitni del analiziranja podatkov za določevanje uporabniškega zadovoljstva in izkušnje je tudi razumevanje afektivnega in kognitivnega zadovoljstva. Afektivno zadovoljstvo je zadovoljstvo, ki je povezano s čustvi posameznika, medtem ko kognitivno zadovoljstvo predstavlja razumevanje in razmišljanje posameznika. Posamezne kategorije izkušenj se delijo na poddomene [84]. Pridobljene podatke iz anket smo najprej razdelili v afektivno in kognitivno zadovoljstvo, nato pa na podlagi ostalih rezultatov sestavili končne ocene uporabniškega zadovoljstva in uporabniške izkušnje. Vsako vprašanje iz ankete so udeleženci ocenili s pomočjo 5-stopenjske lestvice (se popolnoma ne strinjam, se ne strinjam, sem neodločen/-a, se strinjam in se popolnoma strinjam), nanizane v Tabela 5.29, Tabela 5.30 in Tabela 5.31.

Tabela 5.29: Anketna vprašanja za določevanje afektivnega in kognitivnega zadovoljstva

Afektivno zadovoljstvo	Kognitivno zadovoljstvo
Odzivnost aplikacije mi je pomembna.	Aplikacija je odzivna.
Hitrost aplikacije mi je pomembna.	Aplikacija je hitra.
Veliko število funkcionalnosti v aplikaciji mi je pomembno.	Z aplikacijo sem zadovoljen/-a.
Enostavni uporabniški vmesnik mi je pomemben.	Aplikacija je uporabna.
Dosegljivost in stabilnost aplikacije sta mi pomembni.	Uporaba aplikacije je prijetna in intuitivna.

Tabela 5.30: Kriteriji za uporabniško zadovoljstvo

Kriteriji za uporabniško zadovoljstvo	Merila
Identificirani aplikacijski atributi	Vsak pozitivno identificiran atribut je +1, vsak negativen je -1. Maksimalno je 3 točke (100 %).
Vsesplošna stopnja zadovoljstva	Upoštevanje odgovorov »sem delno zadovoljen« in »sem popolnoma zadovoljen«.
NPS točkovanje	1–7 so nepromotorji, 7–8 pasivni udeleženci in 9–10 aktivni promotorji.
Afektivna in kognitivna stopnja zadovoljstva	Seštevek povprečja afektivnega in kognitivnega zadovoljstva.
Stopnja zadrževanja uporabnikov	Upoštevamo odgovore na vprašanje »Koliko časa že uporabljate testirano aplikacijo« in na podlagi »prvič« odgovora določimo stopnjo zadrževanja uporabnikov.

Net Promoter Score Definition oziroma NPS je merilo, s pomočjo katerega lahko določimo zadovoljstvo uporabnika na podlagi verjetnosti promoviranja aplikacije oziroma izdelka [67]. NPS ocena se izračuna na podlagi enačbe (5.4) [67]:

$$NPS = promoters (\%) - detractors (\%), \quad (5.4)$$

kjer velja:

promoters – uporabniki, ki promovirajo aplikacijo naprej,

detractors – pasivni uporabniki, ki ne promovirajo aplikacije.

Tabela 5.31: Kriteriji za uporabniško izkušnjo

Kriteriji za uporabniško izkušnjo	Merila
Procent uporabnikov, ki so zadovoljni s hitrostjo aplikacije.	Upoštevajo se odgovori »se strinjam« in »se popolnoma strinjam« na vprašanje »Aplikacija je hitra«.
Procent uporabnikov, ki so potrdili, da funkcionalnosti zadovoljijo njihove potrebe.	Upoštevajo se odgovori »se strinjam« in »se popolnoma strinjam« na vprašanje »Z aplikacijo sem zadovoljen/-a«.
Procent uporabnikov, ki so potrdili da je enostavno navigirati skozi aplikacijo.	Upoštevajo se odgovori »se strinjam« in »se popolnoma strinjam« na vprašanje »Z navigiranjem po aplikaciji nisem imel/-a problemov«.
Procent uporabnikov, ki so potrdili, da jim je "Iskanje" povrnilo relevantne zadetke.	Upoštevajo se odgovori »se strinjam« in »se popolnoma strinjam« na vprašanje »Preko iskanja sem našel/-a iskane podatke«.
Procent uporabnikov, ki so potrdili, da je enostavno locirati iskano informacijo v aplikaciji.	Upoštevajo se odgovori »se strinjam« in »se popolnoma strinjam« na vprašanje »Enostavno je locirati iskane podatke na vmesniku«.

Kriteriji za uporabniško izkušnjo	Merila
Procent uporabnikov, ki so ocenili aplikacijo kot vizualno prijetno.	Upoštevajo se odgovori »se strinjam« in »se popolnoma strinjam« na vprašanje »Ocenite aplikacijo na podlagi kriterijev – vizualna izdelava«.
Procent uporabnikov, ki so ocenili vsesplošno uporabniško izkušnjo kot pozitivno.	Upošteva se odgovor »popolnoma sem zadovoljen/-a« pri vsesplošni stopnji zadovoljstva.

Nato smo v SPSS izračunali vrednosti posameznih odgovorov za pridobitev uporabniške izkušnje tako za domorodno rešitev kot tudi za spletno mobilno aplikacijo za vse štiri aplikacije (Facebook, Twitter, YouTube in GMail). Uporabniška izkušnja kot tudi uporabniško zadovoljstvo se na podlagi odstotkov nato oceni s 5-stopenjsko lestvico, in sicer:

- 0–45 % je nezadostno.
- 46–59,9 % je zadostno.
- 60–74,9 % je dobro.
- 75–89,9 % je prav dobro.
- 90–100 % je odlično.

Uporabniška izkušnja se nato izračuna po enačbi (5.5):

$$\sum \text{uporabniška izkušnja} = \frac{\text{hitrost} + \text{funkcionalnost} + \text{navigiranje} + \text{iskanje} + \text{lociranje} + \text{vizualnost} + \text{splošna izkušnja}}{7} \quad (5.5)$$

kjer je:

hitrost – procent uporabnikov, ki so zadovoljni s hitrostjo aplikacije,

funkcionalnost – procent uporabnikov, kjer funkcionalnosti zadovoljijo njihove potrebe,

navigiranje – procent uporabnikov, ki so potrdili enostavno navigiranje,

iskanje – procent uporabnikov, kjer jim je iskanje povrnilo relevantne zadetke,

lociranje – procent uporabnikov, ki so potrdili enostavno lociranje iskanje informacij,

vizualnost – procent uporabnikov, ki so ocenili aplikacijo kot vizualno prijetno,

splošna izkušnja – procent uporabnikov, ki so ocenili vsesplošno uporabniško izkušnjo kot pozitivno.

Tabela 5.32: Ocena uporabniške izkušnje Twitter aplikacije

Kriteriji za uporabniško izkušnjo	Domorodna	Spletna
Hitrost	100 %	72 %
Funkcionalnost	96 %	48 %
Navigiranje	82 %	54 %
Iskanje	72 %	56 %
Lociranje	90 %	60 %
Vizualnost	76 %	14 %
Splošna izkušnja	44 %	8 %
Ocenjena uporabniška izkušnja	78,6 %	44,6 %

Tabela 5.33: Ocena uporabniške izkušnje GMail aplikacije

Kriteriji za uporabniško izkušnjo	Domorodna	Spletna
Hitrost	100 %	40 %
Funkcionalnost	100 %	18 %
Navigiranje	100 %	56 %
Iskanje	94 %	48 %
Lociranje	100 %	48 %
Vizualnost	96 %	18 %
Splošna izkušnja	94 %	0 %
Ocenjena uporabniška izkušnja	98 %	33 %

Tabela 5.34: Ocena uporabniške izkušnje YouTube aplikacije

Kriteriji za uporabniško izkušnjo	Domorodna	Spletna
Hitrost	98 %	64 %
Funkcionalnost	100 %	52 %
Navigiranje	88 %	74 %
Iskanje	94 %	76 %
Lociranje	96 %	78 %
Vizualnost	88 %	32 %
Splošna izkušnja	78 %	10 %
Ocenjena uporabniška izkušnja	92 %	55,1 %

Tabela 5.35: Ocena uporabniške izkušnje Facebook aplikacije

Kriteriji za uporabniško izkušnjo	Domorodna	Spletna
Hitrost	98 %	38 %
Funkcionalnost	96 %	20 %
Navigiranje	100 %	60 %
Iskanje	100 %	72 %
Lociranje	94 %	24 %
Vizualnost	90 %	2 %
Splošna izkušnja	64 %	2 %
Ocenjena uporabniška izkušnja	92 %	31,1 %

Tabela 5.32, Tabela 5.33, Tabela 5.34 in Tabela 5.35 prikazujejo rezultate ocenjevanja uporabniške izkušnje, kjer smo za Twitter aplikacijo dobili 78,6 % (domorodna različica) in 44,6 % (spletna različica), za GMail aplikacijo 98 % (domorodna različica) in 33 % (spletna različica), za YouTube aplikacijo 92 % (domorodna različica) in 55,1 % (spletna različica) ter za Facebook aplikacijo 92 % (domorodna različica) in 31,1 % (spletna različica). S pomočjo lestvice, predstavljene v začetku poglavja analize, pretvorimo procentne izračune v nize. Tako so uporabniki ocenili uporabniško izkušnjo pri Twitter aplikaciji kot prav dobro (domorodna različica) in nezadostno (spletna različica), pri GMail aplikaciji kot odlično (domorodna različica) in nezadostno (spletna različica), pri YouTube aplikaciji kot odlično (domorodna različica) in zadostno (spletna različica) ter pri Facebook aplikaciji kot odlično (domorodna različica) in nezadostno (spletna različica). Uporabniško zadovoljstvo smo izračunali po enačbi (5.6):

$$\sum \text{uporabniško zadovoljstvo} = \frac{\text{atributi} + \text{zadovoljstvo} + \text{NPS} + \text{afektivno in kognitivno} + \text{zadrževanje}}{5}, \quad (5.6)$$

kjer so:

atributi – identificirani aplikacijski atributi,

zadovoljstvo – vsesplošna stopnja zadovoljstva,

NPS – verjetnost priporočitve aplikacije prijateljem/sodelavcem/znancem,

afektivno in kognitivno – afektivna in kognitivna stopnja zadovoljstva,

zadrževanje – stopnja zadrževanja uporabnikov.

Tabela 5.36: Ocena uporabniškega zadovoljstva pri Twitter aplikaciji

Kriteriji za uporabniško zadovoljstvo	Domorodna	Spletna
Atributi	100 %	66 %
Zadovoljstvo	96 %	50 %
NPS	72,5 %	-50 %
Afektivno in kognitivno	80 %	80 %
Zadrževanje	80,4 %	12 %
Ocenjeno uporabniško zadovoljstvo	87,8 %	41,6 %

Tabela 5.37: Ocena uporabniškega zadovoljstva pri Gmail aplikaciji

Kriteriji za uporabniško zadovoljstvo	Domorodna	Spletna
Atributi	100 %	33 %
Zadovoljstvo	94 %	8 %
NPS	100 %	0 %
Afektivno in kognitivno	90 %	80 %
Zadrževanje	100 %	0 %
Ocenjeno uporabniško zadovoljstvo	96,6 %	24,2 %

Tabela 5.38: Ocena uporabniškega zadovoljstva pri YouTube aplikaciji

Kriteriji za uporabniško zadovoljstvo	Domorodna	Spletna
Atributi	100 %	66 %
Zadovoljstvo	100 %	52 %
NPS	88 %	0 %
Afektivno in kognitivno	90 %	80 %
Zadrževanje	100 %	6 %
Ocenjeno uporabniško zadovoljstvo	95,6 %	40,8 %

Tabela 5.39: Ocena uporabniškega zadovoljstva pri Facebook aplikaciji

Kriteriji za uporabniško zadovoljstvo	Domorodna	Spletna
Atributi	100 %	33 %
Zadovoljstvo	96 %	8 %
NPS	78 %	0 %
Afektivno in kognitivno	80 %	80 %

Zadrževanje	98 %	4 %
Ocenjeno uporabniško zadovoljstvo	90,4 %	25 %

Tabela 5.36, Tabela 5.37, Tabela 5.38 in Tabela 5.39 prikazujejo rezultate ocen uporabniškega zadovoljstva za vse štiri mobilne aplikacije, in sicer Twitter, GMail, YouTube in Facebook. Na podlagi teh rezultatov smo ocenili uporabniško zadovoljstvo za Twitter aplikacijo 87,8 % (domorodna različica) in 41,6 % (spletna različica), GMail aplikacijo 96,6 % (domorodna različica) in 24,2 % (spletna različica), YouTube aplikacijo 95,6 % (domorodna različica) in 40,8 % (spletna različica) ter Facebook aplikacijo 90,4 % (domorodna različica) in 25 % (spletna različica). Pridobljene rezultate lahko pretvorimo s pomočjo lestvice, in sicer je uporabniško zadovoljstvo pri Twitter aplikaciji prav dobro (domorodna različica) in nezadostno (spletna različica), pri GMail aplikaciji odlično (domorodna različica) in nezadostno (spletna različica), pri YouTube aplikaciji odlično (domorodna različica) in nezadostno (spletna različica) ter pri Facebook aplikaciji odlično (domorodna različica) in nezadostno (spletna različica). Tabela 5.40 prikazuje skupek rezultatov za vse omenjene aplikacije.

Tabela 5.40: Rezultat uporabniške izkušnje in zadovoljstva za vse aplikacije.

Ime aplikacije	Uporabniška izkušnja		Uporabniško zadovoljstvo	
	<i>Domorodna</i>	<i>Spletna</i>	<i>Domorodna</i>	<i>Spletna</i>
Twitter	prav dobro	nezadostno	prav dobro	nezadostno
GMail	odlično	nezadostno	odlično	nezadostno
YouTube	odlično	zadostno	odlično	nezadostno
Facebook	odlično	nezadostno	odlično	nezadostno

5.4 Analiziranje raziskovalnih vprašanj

Na podlagi opravljene pregleda literature lahko odgovorimo na zastavljena raziskovalna vprašanja.

RV 1: Kako vpliva izbira razvojnega pristopa na zmogljivost končnega izdelka?

V sklopu magistrskega dela smo se osredotočili na dva razvojna pristopa in njuna končna izdelka – domorodne in spletne mobilne aplikacije. V poglavju 5.3.8 smo izračunali zmogljivost spletnih mobilnih aplikacij, kjer se je najboljše odrezala aplikacija GMail (ocena

dobro), na drugem mestu je bila aplikacija YouTube (ocena zadostno) in na zadnjem mestu aplikaciji Facebook in Twitter (ocena nezadostno). Ker smo bili omejeni glede izvajanja eksperimenta na domorodnih mobilnih aplikacijah, smo lahko zmogljivost končnega izdelka v celoti izmerili samo za spletne različice. Lahko pa predpostavimo izmerjene vrednosti uporabniškega zadovoljstva in izkušnje iz Tabela 5.40, kjer je razvidno, da so udeleženci eksperimenta dojemali domorodne različice kot boljše, hitrejše in predvsem zmogljivejše rešitve v primerjavi s spletnimi. Tako lahko odgovorimo na raziskovalno vprašanje, da v našem primeru izbira razvojnega pristopa kritično vpliva na zmogljivost končnega izdelka (spletne rešitve so se slabo odrezale tako v zmogljivosti kot tudi v uporabniški izkušnji in zadovoljstvu).

RV 2: Kaj omogoča mobilnim aplikacijam, razvitim s pomočjo spletnih tehnologij, konkurenčni obstoj vzporedno z domorodnimi različicami?

V poglavju 4.5 izpostavimo prednosti in slabosti mobilnih aplikacij, temelječih na spletnih tehnologijah. Konkurenčnost spletnih mobilnih aplikacij izhaja iz dejstva, da omogočajo medplatformsko podporo, so relativno enostavnega in hitrega razvoja, predstavljajo cenejšo vzdrževanje in razvoj, omogočajo enostaven in univerzalen dostop do aplikacije preko spletnega naslova, kar posledično pomeni, da se tudi izognejo validaciji verifikacije pred vstopom v aplikacijsko trgovino.

RV 3: Kakšni so faktorji zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah?

Na osnovi pregleda in analize literature smo prišli do nabora faktorjev zmogljivosti, ki smo jih tudi definirali v obliki teoretičnega modela analize (Slika 5.1). Identificirani faktorji so tip vsebine aplikacije, kompleksnost vsebine, stabilnost, hitrost in uporabniško zadovoljstvo ter uporabniška izkušnja. Pri tem smo pri dokazovanju zmogljivosti v poglavju 5.3.8 upoštevali le tiste faktorje, ki se uporabljajo pri splošno razširjenih testnih orodjih za preverjanje zmogljivosti spletnih aplikacij in strani (YSlow, GTMetrix, PageSpeed in DynaTrace) – število zahtev, čas nalaganja strani, velikost strani. Dodali smo še faktorje uporabniške izkušnje in zadovoljstva, saj je zmogljivost po definiciji skupek teh karakteristik.

RV 4: Ali izbira mobilne aplikacije, temelječe na spletnih tehnologijah, vpliva na uporabniško izkušnjo in zadovoljstvo?

V poglavjih 5.3.8 in 5.3.9 smo analizirali podatke na podlagi ankete, ki so jo udeleženci izpolnili po opravljenem eksperimentu. Na podlagi teh podatkov smo lahko s pomočjo afektivnih in kognitivnih vprašanj in njihovih odgovorov določili uporabniško izkušnjo in

zadovoljstvo tako pri spletnih kot tudi pri domorodnih mobilnih aplikacijah. Iz Tabela 5.40 je razvidno, da izbira mobilne aplikacije, temelječe na spletnih tehnologijah, negativno vpliva tako na uporabniško izkušnjo kot tudi na zadovoljstvo uporabnikov. Domorodne mobilne aplikacije so se v povprečju odrezale z oceno 5 (odlična izkušnja in zadovoljstvo), spletne aplikacije imajo povprečno oceno 1 (nezadostna izkušnja kot tudi zadovoljstvo).

RV 5: Ali izbira mobilne aplikacije, temelječe na spletnih tehnologijah, vpliva na dojeto zmogljivost, enostavnost uporabe, kakovost in odzivnost?

Na zmogljivost smo se osredotočili v poglavju 5.3.8, kjer smo dokazali, da se spletne mobilne aplikacije odrežejo zelo slabo glede zmogljivosti (GMail je dobil oceno dobro, YouTube aplikacije le zadostno in Facebook ter Twitter oceno nezadostno). Statistično gledano so se glede dojete enostavnosti uporabe, kakovosti in odzivnosti ponovno spletne mobilne aplikacije odrezale slabše kot njihove domorodne različice, kar je razvidno v poglavju 5.3.5. V povprečju so udeleženci eksperimenta ocenili domorodne različice z prav dobro in odlično, medtem ko so spletne različice enakih aplikacij prejele ocene zelo slabo do dobro.

5.5 Analiziranje hipotez

Postavljene hipoteze smo s pomočjo metode eksperimenta in ankete raziskali ter jih bodisi potrdili ali ovrgli.

H1: Mobilna aplikacija, razvita s pomočjo spletnih tehnologij, izkazuje enako enostavnost uporabe, kakovost, odzivnost in vizualno izdelavo kot njena domorodna različica.

S pomočjo izvedene ankete, ki je bila del našega eksperimenta, smo pridobili ter analizirali odgovore glede enostavnosti uporabe, kakovosti aplikacije, odzivnosti in vizualne izdelave domorodne in spletne različice. Pri tem smo ugotovili, da prihaja do večjih razlik vrednosti tako med posameznimi spremenljivkami (enostavnost uporabe, kakovost, odzivnost in vizualna izdelava) kot tudi med ocenami za domorodno in spletno aplikacijo. Statistični izračunani podprejo dejstvo, da so bile v vseh štirih primerih aplikacij (Facebook, GMail, Twitter in YouTube) domorodne različice ocenjene bolje kot njihove alternative, zgrajene s pomočjo spletnih tehnologij. Podrobnejše informacije o analizi podatkov so v poglavju 5.3.5. Po teh podatkih smo hipotezo H1 ovrgli.

H2: Mobilna aplikacija, razvita s pomočjo spletnih tehnologij, pozitivno vpliva na zadovoljstvo uporabnikov v primerjavi z domorodno različico.

Kot smo dokazali v poglavjih 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.6 in 5.3.9, so bile domorodne različice testiranih aplikacij (Facebook, GMail, Twitter in YouTube) ocenjene bolje kot njihove spletne različice, temelječe na spletnih tehnologijah. Pri tem smo ugotovili, da so uporabniki občutili tudi boljšo uporabniško izkušnjo v primeru domorodnih aplikacij ter nakazali večjo verjetnost priporočitve in ponovne uporabe kot v primeru mobilnih aplikacij, razvitih s pomočjo spletnih tehnologij. V nobenem primeru nam ni uspelo dokazati, da mobilne aplikacije razvite s pomočjo spletnih tehnologij pozitivno vplivajo na zadovoljstvo uporabnikov, saj povzročijo ravno obratni efekt – uporabniki so nezadovoljni in občutijo negativno uporabniško izkušnjo.

Po tej podatkih smo hipotezo H2 ovrgli.

H3: Mobilne aplikacije, temelječe na spletnih tehnologijah, z večjim številom zahtev v seji izkazujejo večji čas nalaganja, kar negativno vpliva na uporabniško izkušnjo.

Rezultati analize iz poglavja 5.3.7 dokazujejo, da je karakteristika zmogljivosti (hitrost) odvisna od števila zahtev, ki jih aplikacija, temelječa na spletnih tehnologijah, obdela v uporabnikovi seji. Korelacija med spremenljivkami število zahtev in čas nalaganja (onLoad z .813 in DOMContentLoaded z .837) potrjuje linearno močan odnos. V poglavjih 5.3.1 do 5.3.4 smo dokazali, da mobilne aplikacije, temelječe na spletnih tehnologijah, izkazujejo slabšo uporabniško izkušnjo v primerjavi z domorodno različico, kar je posledica slabo dojete hitrosti oziroma časa nalaganja.

Po teh podatkih smo hipotezo H3 potrdili.

H4: Obstaja signifikantna razlika v zmogljivosti med testiranimi aplikacijami, temelječimi na spletnih tehnologijah.

Na podlagi opravljenega eksperimenta in ankete smo dobili podatke o uporabniški izkušnji in zadovoljstvu ter podatke o sami hitrosti in kompleksnosti posameznih mobilnih aplikacij (Facebook, Twitter, YouTube, GMail). Postavili smo enačbo za izračun zmogljivosti v poglavju 5.3.8, na podlagi katere smo izračunali ocene posameznih karakteristik (število zahtev, velikost strani in čas nalaganja). Aplikaciji Facebook in Twitter sta se izkazali zelo slabo, skupna ocena aplikacij je nezadostno (1), medtem ko je skupna zmogljivost spletne mobilne aplikacije YouTube ocenjena z zadostno (2) in GMail aplikacije z dobro (3). Pri primerjavi ocen zmogljivosti posameznih aplikacij dokažemo, da obstaja signifikantna razlika v njihovi zmogljivosti. Signifikantna razlika prihaja predvsem v primerjavi GMail aplikacije z ostalimi.

Po teh podatkih smo hipotezo H4 potrdili.

6 SKLEP

Mobilne aplikacije, temelječe na spletnih tehnologijah, omogočajo dostop do večjega števila končnih uporabnikov. Izbor razvojnega pristopa vpliva na zmogljivost končnega izdelka, saj vsaka tehnologija in pristop prinašata prednosti in slabosti. Danes se posveča pozornost pristopu »mobile first«, kar pomeni, da se spletne strani avtomatično razvijajo najprej za mobilne naprave, šele nato za ostale naprave. Tovrstna sprememba v mentaliteti prinaša tudi spremembo načina razvoja in posledično izbire razvojnega pristopa. Konkurenčni obstoj spletnih mobilnih aplikacij omogočajo predvsem spletne tehnologije, kot so označevalni jezik HTML5 in oblikovalski jezik CSS3 ter skriptni jezik JavaScript, ki lahko strukturirajo, oblikujejo ter dopolnijo funkcionalnosti spletnih strani tako, da se obnašajo in som videti popolnoma enako kot domorodne mobilne aplikacije. Obstaja tudi ogromno orodij (jQuery Mobile, Codiqa, PhoneGap, Titanium, Apache Cordova), s pomočjo katerih lahko ovijemo našo spletno mobilno aplikacijo in s tem ustvarimo hibridno mobilno rešitev. V magistrskem delu smo ugotovili, da je dobra optimizacija spletnih rešitev izredno pomembna tudi v korelaciji z uporabniško izkušnjo in zadovoljstvom. Sodeč po podatkih mobilnih analiz [11], je jasno, da se 89 % uporabnikov zadržuje v domorodnih aplikacijah, medtem ko samo 11 % končnih uporabnikov redno uporablja mobilne spletne aplikacije. Enako dejstvo smo razbrali iz naše analize podatkov anket, kjer smo dokazali višjo stopnjo zadržanosti pri domorodnih aplikacijah v primerjavi s spletnimi (88 % udeležencev je prvič uporabljajo Twitter spletno aplikacijo, odstotek je bil celo višji pri Facebook spletni različici s 96 % in celo 100 % pri GMail spletni aplikaciji). Zato je pomembno, da so karakteristike zmogljivosti aplikacije, kot so hitrost, uporabniška izkušnja in zadovoljstvo, kompleksnost vsebine, stabilnost in odzivnost, optimizirane do točke, kjer uporabniki ne občutijo pomanjkljivosti v vizualni izdelavi, počasen čas nalaganja, neodzivnost, neefektivnost in nepraktičnost aplikacije ter prekompliciranost. Ravno to so bile nekatere od težav, s katerimi so se udeleženci eksperimenta soočili (aplikacije so ocenili kot nepraktične, slabe kakovosti, prekomplicirane, neefektivne in nezanesljive). V sklopu eksperimenta smo se osredotočili na družbene medije, in sicer zgolj zaradi njihove razširjenost v našem vsakdanu. Testirali smo aplikacije Facebook, YouTube, GMail in

Twitter tako v spletni kot v domorodni različici. Pri tem smo dokaj hitro spoznali, da pravih spletnih aplikacij danes ni oziroma naš problem je bil predvsem odkriti aplikacije, kjer imamo tako spletno kot tudi domorodno rešitev. Zaradi tega smo v sklopu spletnih mobilnih aplikacij v večini testirali mobilne spletne strani. Spletne rešitve se v nobeni kategoriji našega eksperimenta niso izkazale enake ali celo boljše zmogljivosti kot njihove domorodne različice, kar v celoti služi kot dokaz, da se je opustil razvoj spletnih aplikacij (govorimo v primeru naših testiranih aplikacij) zaradi večjega interesa in dojetanja v domorodnih rešitvah. Največji razlog za slabe ocene zmogljivosti spletnih mobilnih aplikacij so bile nezadostne uporabniške izkušnje in zadovoljstvo. Testirane aplikacije so bile neoptimizirane, prevelike in preobremenjujoče za mobilnike in mobilno povezavo. Čeprav večjih razlik v času izvedbe eksperimentov med različicama ni, so udeleženci večkrat komentirali počasno in nepopolno delovanja pri spletnih mobilnih aplikacijah. Posledica neoptimiziranosti ter neprijetne in nezadostne uporabniške izkušnje je, da večina uporabnikov ne bi priporočala spletnih mobilnih rešitev naprej oz. jih ne bi uporabljala kot alternativo domorodnim rešitvam. Tako so naši rezultati empirične analize ovrgli hipotezo, da so spletne mobilne rešitve enako zmogljive kot njihove domorodne rešitve na področjih dojete enostavnosti uporabe, kakovosti uporabniškega vmesnika, odzivnosti, uporabniške izkušnje in zadovoljstva. S tem smo potrdili, da večje število zahtev vpliva tako na čas nalaganja kot tudi na korelativno uporabniško zadovoljstvo ter da obstaja signifikantna razlika v zmogljivosti med testiranimi aplikacijami, temelječimi na spletnih tehnologijah.

Raziskovalno delo bi koristilo predvsem razvijalcem, ki želijo spoznati prednosti in slabosti posameznih pristopov ter se seznaniti s področji, kjer spletne mobilne aplikacije ne morejo konkurirati svojim domorodnim rešitvam. Prav tako lahko delo koristi podjetjem bodisi malim bodisi velikim, ki želijo spoznati koristi spletnih mobilnih aplikacij v primerjavi z domorodnimi različicami. Rezultati eksperimenta prikazujejo vpliv karakteristik zmogljivosti na uporabniško izkušnjo in zadovoljstvo, ki posledično vplivata na stopnjo zadržanosti uporabnikov in priporočljivost drugim.

Raziskavo bi lahko nadaljevali ter jo razširili tako, da bi v eksperiment vpeljali še beleženje karakteristik domorodnih rešitev, kar v našem primeru ni bilo omogočeno zaradi finančnih omejitev pri eksperimentu. Tako bi lahko neposredno primerjali razvojna pristopa ter posledično tudi razširili teoretični model. V sam eksperiment bi lahko vpeljali še hibridne mobilne aplikacije, ki niso bile cilj magistrskega dela. Zraven družbenih aplikacij bi lahko vpeljali še druge (mogoče celo aplikacijske igre) ter razširili tudi vzorec eksperimenta. Kljub omejitvam pri eksperimentu in manjšemu vzorcu smo lahko potrdili ali ovrgli zastavljene hipoteze ter odgovorili na raziskovalna vprašanja.

7 VIRI

- [1] Android Developer. Debugging Web Apps. Android, 2015. Dostopno na: <http://developer.android.com/guide/webapps/debugging.html> [22.7.2015]
- [2] Android Developer. Web Apps. Android, 2015. Dostopno na: <http://developer.android.com/guide/webapps/index.html> [22.7.2015]
- [3] Android Developers. Developer Workflow. Android, 2015. Dostopno na: <https://developer.android.com/tools/workflow/index.html> [08.05.2015]
- [4] Android Developers. Material Design. Android, 2015. Dostopno na: <https://developer.android.com/design/material/index.html> [08.05.2015]
- [5] Android Developers. User Interface. Android, 2015. Dostopno na: <https://developer.android.com/guide/topics/ui/index.html> [08.05.2015]
- [6] Bell, K., Google Play now has more apps than Apple's App Store, report says. Mashable, 2015. Dostopno na: <http://mashable.com/2015/01/15/google-play-more-apps-than-ios/> [21.7.2015]
- [7] Boswell, W., Cross-Platform Development: What The Stats Say. Intel.com, 2013. Dostopno na: <https://software.intel.com/en-us/blogs/2013/03/07/cross-platform-development-what-the-stats-say> [08.05.2015]
- [8] Bouras, C., Papazois, A., Stasinou, N., A Framework for Cross-platform Mobile Web Applications Using HTML5. *Future Internet Of Things and Cloud (FiCloud)*, 2014 International Conference, (2014), str. 420 – 424.
- [9] Bristowe, J., What is a Hybrid Mobile App? Telerik Developer Network, 2015. Dostopno na: <http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/> [20.05.2015]
- [10] Buckler, C., Average Page Weight Increased Another 16% in 2015. Dostopno na: <http://www.sitepoint.com/average-page-weight-increased-another-16-2015/> [16.04.2016]
- [11] Chaffey, D., Mobile Marketing Statistics compilation. Smartinsights.com, 2016. Dostopno na: <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/> [17.04.2016]
- [12] Chrome. Chrome DevTools Overview. Dostopno na: <https://developer.chrome.com/devtools> [22.7.2015]
- [13] Chrome. Editing Styles and The Dom Introduction. Developer.chrome.com, 2015. Dostopno na: <https://developer.chrome.com/devtools/docs/dom-and-styles> [22.7.2015]

- [14] Chrome. Evaluating network performance. Developer.chrome.com, 2015. Dostopno na: <https://developer.chrome.com/devtools/docs/network> [22.7.2015]
- [15] Chrome. Managing application storage. Developer.chrome.com, 2015. Dostopno na: <https://developer.chrome.com/devtools/docs/resource-panel> [22.7.2015]
- [16] Chrome. Performance profiling with the Timeline. Developer.chrome.com, 2015. Dostopno na: <https://developer.chrome.com/devtools/docs/timeline> [22.7.2015]
- [17] Chrome. Profiles Panel. Developer.chrome.com, 2015. Dostopno na: <https://developer.chrome.com/devtools/docs/profiles> [22.7.2015]
- [18] Codiqa. Codiqa: Build JQuery Mobile apps, the easy way. Dostopno na: <https://codiqa.com/> [19.04.2016]
- [19] Cohen, A., Understanding the Pros and Cons of Using jQuery Mobile and HTML5. Ulancer.com, 2015. Dostopno na: <http://www.ulancer.com/2015/02/understanding-the-pros-and-cons-of-using-jquery-mobile-and-html5/> [08.05.2015]
- [20] Cohen, J., Statistical Power Analysis for the Behavioral Sciences. Routledge; 1988, 2.izdaja. Hillsdale, NJ: Lawrence Erlbaum Associates.
- [21] Colwyn, S., New consumer media consumption research. Dostopno na: <http://www.smartinsights.com/marketplace-analysis/customer-analysis/consumer-media-device-use/> [08.05.2015]
- [22] Constantinou, A., Developer Economics: State of the developer nation Q1 2015. VisionMobile.com, 2015. Dostopno na: <http://www.visionmobile.com/product/developer-economics-q1-2015-state-developer-nation/> [08.05.2015]
- [23] Dalmasso I., Datta Kanti S., Bonnet C., Nikaein N. Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools. *Wireless Communications and Mobile Computing Conference (IWCMC)*, 9th International, (2013), str. 323 – 328.
- [24] Dyke, A., Visual Glossary: HAR file. MaxCDN.com, 2015. Dostopno na: <https://www.maxcdn.com/one/visual-glossary/har-file/> [12.1.2016]
- [25] Edwards, J., Mobile Apps Are Killing The Free Web, Handing A Censored Duopoly to Google And Apple. Business Insider Tech, 2014. Dostopno na: <http://www.businessinsider.com/mobile-web-vs-app-usage-statistics-2014-4> [08.05.2015]
- [26] Field, A. (2009). Discovering statistics using SPSS, 3rd edition. London: Sage Publications

- [27] Garsiel, T., Irish, P., How Browsers Work: Behind the scenes of modern web browsers. Html5rocks.com. Dostopno na: <http://www.html5rocks.com/en/tutorials/internals/howbrowserswork/> [20.05.2015]
- [28] Gasston P., *The Book of CSS3: A Developer's Guide to the Future of Web Design*. San Francisco: No Starch Press, 2011.
- [29] GlobalWebIndex: Social Media Engagement Q1 2015. Dostopno na: <http://insight.globalwebindex.net/social-media-engagement-q1-2015> [12.1.2016]
- [30] Google, Google HAR Analyzer. Toolbox.googleapps.com, 2016. Dostopno na: https://toolbox.googleapps.com/apps/har_analyzer/ [12.1.2016]
- [31] Google. Material Design: Introduction. Google, 2015. Dostopno na: <http://www.google.com/design/spec/material-design/introduction.html#introduction-goals> [08.05.2015]
- [32] GTmetrix: Analyze your site's speed and make it faster. Dostopno na: <https://gtmetrix.com/> [16.04.2016]
- [33] HTTPArchive: Interesting stats, dne 1.4.2016. Dostopno na: <http://httparchive.org/interesting.php> [16.04.2016]
- [34] IBM. Native, web or hybrid mobile-app development. IBM.com, 2012. Dostopno na: <ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf> [08.05.2015]
- [35] iOS Developer Library. Start Developing iOS Apps Today. Apple, 2015. Dostopno na: https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/index.html#//apple_ref/doc/uid/TP40011343-CH2-SW1 [08.05.2015]
- [36] iOS Developer Library. Designing for iOS. Developer, 2015. Dostopno na: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html#//apple_ref/doc/uid/TP40006556-CH66-SW1 [08.05.2015]
- [37] Jacobs, M., Introduction to Universal Windows Platform (UWP) apps for designers. Microsoft, 2016. Dostopno na: <https://msdn.microsoft.com/library/windows/apps/dn958439.aspx> [18.04.2016]
- [38] Jacobs, M., Responsive design 101 for Universal Windows Platform (UWP) apps. Microsoft, 2016. Dostopno na: <https://msdn.microsoft.com/en-us/library/windows/apps/dn958435.aspx> [18.04.2016]
- [39] Janssen, C., Three-Tier Architecture. Techopedia.Techopedia.com, 2015. Dostopno na: <http://www.techopedia.com/definition/24649/three-tier-architecture> [21.7.2015]

- [40] Johansson D., Andersson K., Web-based Adaptive Application Mobility. *Cloud Networking (CLOUDNET)*, 1st International Conference, (2012), str. 87–94.
- [41] JQuery Mobile. A Touch-Optimized Web Framework. Dostopno na: <http://jquerymobile.com/about/> [08.05.2015]
- [42] JSripters. JavaScript: Advantages and Disadvantages. Dostopno na: <http://www.jsripters.com/javascript-advantages-and-disadvantages/> [13.05.2015]
- [43] Kendo UI. Kendo UI jQuery and HTML5 widgets. Dostopno na: <http://www.telerik.com/kendo-ui> [08.05.2015]
- [44] Kent State University: SPSS Tutorials, Pearson Correlation. Dostopno na: <http://libguides.library.kent.edu/SPSS/PearsonCorr> [14.04.2016]
- [45] Khalaf, S., Shopping, Productivity and Messaging Give Mobile Another Stunning Growth Year. Dostopno na: <http://flurrymobile.tumblr.com/post/115194992530/shopping-productivity-and-messaging-give-mobile#.VTfPDCGqpBf> [08.05.2015]
- [46] Kirubakaran, B., Karthikeyani, V., Mobile Application Testing – Challenges and Solution Approach through Automation. *Pattern Recognition, Informatics and Mobile Engineering (PRIME)*, 2013 International Conference , 2013, str. 79–84
- [47] Koetsier, J., HTML5 vs. native vs. hybrid mobile apps: 3,500 developers say all three, please. Venturebeat.com, 2013. Dostopno na: <http://venturebeat.com/2013/11/20/html5-vs-native-vs-hybrid-mobile-apps-3500-developers-say-all-three-please/> [08.05.2015]
- [48] Leeheer, N., How well does your browser support HTML5? Html5test.com, 2015. Dostopno na: <https://html5test.com/results/desktop.html> [13.05.2015]
- [49] Li, X., Bao, Z., Performance Characterization of Web Applications with HTML5 Enhancements. *Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on*, 2014, str. 252–258
- [50] Lionbridge. Mobile Web Apps vs. Mobile Native Apps: How to Make the Right Choice. Lionbridge.com, 2012. Dostopno na: http://www.lionbridge.com/files/2012/11/Lionbridge-WP_MobileApps2.pdf [08.02.2015]
- [51] Lund, A., Lund., M., Cronbach's Alpha using SPSS Statistics. Dostopno na: <https://statistics.laerd.com/spss-tutorials/cronbachs-alpha-using-spss-statistics.php> [20.04.2016]

- [52] Lund, A., Lund., M., Mann-Whitney U Test using SPSS Statistics. Dostopno na: <https://statistics.laerd.com/spss-tutorials/mann-whitney-u-test-using-spss-statistics.php> [6.4.2016]
- [53] Microsoft. User interface for Windows Phone 8. Microsoft, 2015. Dostopno na: [https://msdn.microsoft.com/en-us/library/windows/apps/ff967556\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/ff967556(v=vs.105).aspx) [08.05.2015]
- [54] Nielsen, J., Website Response Times. Nngroup.com, 2016. Dostopno na: <https://www.nngroup.com/articles/website-response-times/> [16.04.2016]
- [55] Pilgrim M., Dive Into HTML5. Mark Pilgrim, 2011. Dostopno na: <http://diveintohtml5.info/introduction.html> [13.05.2015]
- [56] ProgrammableWeb. The leading source of news & information about APIs, chronicling the evolution of the global API economy & providing the web's most relied-on API Directory. 2015. Dostopno na: <http://www.programmableweb.com/category> [08.05.2015]
- [57] Pronschinske, M., The State of Native vs. Web vs. Hybrid. DZone/Mobile Zone, 2014. Dostopno na: <https://dzone.com/articles/state-native-vs-web-vs-hybrid> [18.04.2016]
- [58] Quintagroup. Responsive web design – adapt, respond and overcome. Quintagroup.com. Dostopno na: <http://quintagroup.com/services/web-design/responsive-web-design> [16.04.2016]
- [59] Raj, J., The Top 7 Hybrid Mobile App Frameworks. Sitepoint, 2014. Dostopno na: <http://www.sitepoint.com/top-7-hybrid-mobile-app-frameworks/> [08.05.2015]
- [60] Rende, J., Native vs. HTML5 Mobile App Development: Which option is best? Appcelerator.com, 2012. Spletna stran ne obstaja več.
- [61] Riyanto, S., Segmentation and Classification for Display Devices Website of Heterogeneous. LIPI, 2015. Dostopno na: <http://www.pdii.lipi.go.id/read/2015/01/08/segmentasi-dan-klasifikasi-untuk-tampilan-website-dari-perangkat-yang-heterogen/> [19.04.2016]
- [62] Rouse, M., Definition of Performance. WhatIs.com, 2006. Dostopno na: <http://whatIs.techtarget.com/definition/performance> [18.04.2016]
- [63] Rudolphi, T., Using Sencha Touch as a Strategic Platform – An Evaluation. Zuehlke Blog, 2013. Dostopno na: <http://blog.zuehlke.com/en/using-sencha-touch-as-a-strategic-platform-an-evaluation/> [08.05.2015]
- [64] Salesforce. Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options. Dostopno na:

- https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options [20.05.2015]
- [65] Sencha Touch. Cross-platform Mobile Web App Development Framework for HTML5 and JS. Dostopno na: <http://www.sencha.com/products/touch/#overview> [08.05.2015]
- [66] Statista. Statistics and facts about Mobile App Usage. Statista.com, 2014. Dostopno na: <http://www.statista.com/topics/1002/mobile-app-usage/> [08.05.2015]
- [67] SurveyMonkey. Net Promoter® Score Definition and Formula. Dostopno na: <https://www.surveymonkey.com/mp/net-promoter-score-definition-formula/> [17.04.2016]
- [68] Svedsen, E., Mobile Paradigms Explained. Ais Blog, 2013. Dostopno na: <http://blog.appliedis.com/2013/01/20/mobile-paradigms-explained/> [08.05.2015]
- [69] Telerik Marketing. Everything You Wanted To Know About Native, Hybrid and Web Apps; but Were Afraid To Ask. Telerink.com, 2014. Dostopno na: <http://blogs.telerik.com/telerikmarketing/posts/14-05-21/everything-you-wanted-to-know-about-native-hybrid-and-web-apps-but-were-afraid-to-ask> [08.05.2015]
- [70] Telerik, Fiddler: The free web debugging proxy for any browser, system or platform. Dostopno na: <http://www.telerik.com/fiddler> [12.1.2016]
- [71] Trice, A., PhoneGap explained visually. PhoneGap.com, 2015. Dostopno na: <http://phonegap.com/2012/05/02/phonegap-explained-visually/> [25.7.2015]
- [72] Van der Meulen, R., Rivera, J., Gartner Identifies Top 10 Mobile Technologies and Capabilities for 2015 and 2016. Gartner, 2014. Dostopno na: <http://www.gartner.com/newsroom/id/2669915> [08.05.2015]
- [73] Van der Meulen, R., Rivera, J., Gartner Says by 2013, More Than 50 Percent of Mobile Apps Deployed Will Be Hybrid. Gartner, 2013. Dostopno na: <http://www.gartner.com/newsroom/id/2324917> [08.05.2015]
- [74] Van der Meulen, R., Rivera, J., Gartner Says by 2017, Mobile Users Will Provide Personalized Data Streams to More Than 100 Apps and Services Every Day. Gartner, 2014. Dostopno na: <http://www.gartner.com/newsroom/id/2654115> [08.05.2015]
- [75] Viswanathan, P., The Pros and Cons of Native Apps and Mobile Web Apps. About Tech, 2015. Dostopno na: <http://mobiledevices.about.com/od/additionalresources/qt/The-Pros-And-Cons-Of-Native-Apps-And-Mobile-Web-Apps.htm> [08.05.2015]

- [76] Webkit. Webkit: Open Source Web Browsing Engine. Dostopno na: <https://webkit.org/> [18.04.2016]
- [77] Wikipedia, Web performance. Dostopno na: https://en.wikipedia.org/wiki/Web_performance [18.04.2016]
- [78] Wikipedia. Appcelerator Titanium: Architecture. Dostopno na: http://en.wikipedia.org/wiki/Appcelerator_Titanium [20.05.2015]
- [79] Wikipedia. Comparison of web browser engines. Dostopno na: https://en.wikipedia.org/wiki/Comparison_of_web_browser_engines [22.7.2015]
- [80] Wikipedia. Cronbach's alpha. Dostopno na: https://en.wikipedia.org/wiki/Cronbach%27s_alpha [20.04.2016]
- [81] Wikipedia. HTML5. Dostopno na: <http://en.wikipedia.org/wiki/HTML5> [13.05.2015]
- [82] Wikipedia. JavaScript. Dostopno na: <http://en.wikipedia.org/wiki/JavaScript> [13.05.2015]
- [83] Wikipedia. Web Worker. Dostopno na: http://en.wikipedia.org/wiki/Web_worker [13.05.2015]
- [84] Wilson O. L., Three domains of learning – What are the differences between the cognitive, affective, and psychomotor taxonomies? Dostopno na: <http://thesecondprinciple.com/instructional-design/threedomainsoflearning/> [17.04.2016]
- [85] Work, S., How loading time affects your bottom line. Kissmetrics.com, 2016. Dostopno na: <https://blog.kissmetrics.com/loading-time/> [16.04.2016]

8 PRILOGE

Magistrskem delu so priložene naslednje priloge:

- navodila za izvajanje eksperimenta,
- proces izvedbe eksperimenta,
- grafi in tabele statističnih analiz mobilnih aplikacij.

Priloga A – Navodila za izvajanje eksperimenta

Testiranje mobilnih aplikacij

Pozdravljeni!

Pred vami so navodila za izvedbo določenih aktivnosti na različnih aplikacijah, kot so Twitter, Facebook, YouTube in GMail (Google Mail). Vaše predhodne izkušnje s temi aplikacijami ne igrajo ključne vloge.

Prosim, da sledite navodilu po predvidevanih korakih ter da po zadnjem koraku obvestite opazovalca, da ste končali s testiranjem.

Kakršna koli vprašanja med izvajanjem testiranja niso dovoljena, razen če imate občutek, da nimate dovolj znanja o uporabnosti aplikacije, da bi te testne aktivnosti izvedli, kot je predvidevano.

Testiranje aktivnosti Twitter

1. Vpišite se v družbeno omrežje Twitter.

Uporabnik obiše Twitter preko Google brskalnika (mobile.twitter.com) ter vnese svoje vpisne podatke (uporabniško ime in geslo).

Uporabniško ime: TestniDroid

Geslo: testni086maribor

2. Objavite svoj prvi »tweet«.

Uporabnik navigira po aplikaciji in zraven besedila »*To je samo test* 😊« objavi še sliko, kai jo zajame s pomočjo mobilne kamere.

3. Preglejte »trending topics« in »retweetajte« objavo iz te kategorije.

Uporabnik pregleda »trending topics« (teme, ki se trenutno objavljajo na Twitter družbenem omrežju) in »retweeta« poljubno. »Retweet« pomeni, da objavimo izbrano temo iz tujega profila na svojem.

4. Odjavite se iz aplikacije.

Testiranje aktivnosti Facebook

1. Vpišite se v družbeno omrežje Facebook.

Uporabnik obiše Facebook preko Google brskalnika (m.facebook.com) ter vnese svoje vpisne podatke (e-mail in geslo).

E-mail: testni.research@GMail.com

Geslo: facebook086maribor

2. Posodobite svoj Facebook status.

Uporabnik navigira po aplikaciji in zraven besedila »*Testiram Facebook aplikacijo*« še objavi sliko, ki jo zajame s pomočjo mobilne kamere, izbere trenutno lokacijo ter navede, kako se trenutno počuti.

3. Poiščite poljubno podjetje in ga všečkajte.

Uporabnik poišče preko Facebook iskalnika poljubno znano osebo ali podjetje, prebere nekaj objav in všečka poljubno objavo.

4. Izbranemu prijatelju napišite ena-na-ena sporočilo.

Uporabnik navigira v aplikaciji do sporočil ter napiše novo sporočilo »*To je samo test. Izvajam eksperiment 😊*« izbranemu prijatelju.

5. Odjavite se iz aplikacije.

Testiranje aktivnosti YouTube

1. Vpišite se v družbeno omrežje YouTube.

Uporabnik obiše YouTube preko Google brskalnika (m.youtube.com) ter izbere v naprej pripravljen račun za testiranje aplikacije.

2. Poiščite poljubni videoposnetek

Uporabnik preko YouTube iskalnika poišče poljubni videoposnetek, ga izbere ter ga odpre v celozaslonskem načinu. Uporabnik naj si ogleda video posnetek za nekaj sekund (lahko tudi preskoči do sredine in do konca) ter izbere izhod iz celozaslonskega načina.

3. »Thumbs up« ali »thumbs down« video posnetek in ga shranite med favorite.

Uporabnik všečka izbrani video posnetek ter ga doda med svoje favorite.

4. Odjavite se iz aplikacije

Testiranje aktivnosti GMail

1. Vpišite se v družbeno omrežje GMail.

Uporabnik obiše GMail preko Google brskalnika (mail.google.com) ter izbere v naprej pripravljen račun za testiranje aplikacije.

2. Ustvarite novo elektronsko sporočilo

Uporabnik navigira po aplikaciji in izbere pošiljanje novega elektronskega sporočila.

- **Naslovnik:** fajfar.andreja@GMail.com
- **Naslov:** Testiranje GMail
- **Priponka:** Dodamo sliko zajeto s pomočjo mobilne kamere (*attach a file*).
- **Besedilo:** Testiram GMail aplikacijo. Preverjamo, kako se aplikacijo razlikuje v različnih oblikah. ☺

3. Odjavite se iz aplikacije

Priloga B – Proces izvedbe eksperimenta

Naslov teme	Analiza zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah	
Avtor	Andreja Fajfar	
Ozadje problema	<p>Izpostavlja se problematika zmogljivosti mobilnih aplikacij, sestavljenih s pomočjo spletnih tehnologij, zaradi tega se v fokus vzame faktor zmogljivosti ter analizira, ali je zmogljivost zares toliko slabša v spletnih mobilnih aplikacijah kot v njenih sorodnih domorodnih rešitvah. Pri raziskovanju se bomo osredotočili na zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah. Zmogljivost smo definirali kot sposobnost obdelave podatkov v določenem času oziroma kot popolno učinkovitost sistema, kar vključuje njegovo sposobnost obdelave podatkov v določenem času, odzivnost, stabilnost in izkoriščenost virov. Osredotočili se bomo na to, kako tip vsebine in njuna kompleksnost vplivata na uporabniško izkušnjo kot tudi uporabniško zadovoljstvo ter kako uporabljavnost virov, hitrost in stabilnost vplivajo na vsesplošno zmogljivost mobilne aplikacije.</p>	
Raziskovalna vprašanja	<p>RV 1: Kako vpliva izbira razvojnega pristopa na zmogljivost končnega izdelka?</p> <p>RV 2: Kaj omogoča mobilnim aplikacijam, razvitim s pomočjo spletnih tehnologij, konkurenčni obstoj vzporedno z domorodnimi različicami?</p> <p>RV 3: Kakšni so faktorji zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah?</p> <p>RV 4: Ali izbira mobilne aplikacije, temelječe na spletnih tehnologijah, vpliva na uporabniško izkušnjo in zadovoljstvo?</p> <p>RV 5: Ali izbira mobilne aplikacije, temelječe na spletnih tehnologijah, vpliva na dojeto zmogljivost, enostavnost uporabe, kakovost in odzivnost?</p>	
Ideja eksperimenta	<p>Ideja eksperimenta je analizirati zmogljivost mobilnih aplikacij temelječih na spletnih tehnologijah s pomočjo testiranja njihove stabilnosti, hitrosti in izkoriščenost virov. S pomočjo opazovanja in ankete pa pridemo mnenja glede uporabniške izkušnje in zadovoljstva ter določimo kako posamezni indikatorji vplivajo na vsesplošno zmogljivost in dojemanje zmogljivosti aplikacije.</p>	
Definicija (The Goal-Question-Metric model by Basili &	Motivacija	Razumevanje zmogljivosti mobilnih aplikacij, temelječih na spletnih tehnologijah
	Objekt opazovanja	Izdelki (mobilna aplikacija Facebook, Twitter, GMail in YouTube)

Weiss, 1984 and Basili & Rombach, 1988)	Namen raziskave	Napovedovanje konkurenčnosti spletnih mobilnih aplikacij v primerjavi z drugimi rešitvami.
	Vidiki kakovosti	Zmogljivost mobilne aplikacije
	Vidik opazovalca, pogled	Razvijalec, zunanja oseba, uporabnik.
	V kontekstu	Potrebna je internetna povezava, dostopnost aplikacije preko strežnika kot tudi strojna oprema (brskalnik) za testiranje zmogljivosti.

Načrt eksperimen- ta	Natančna določitev konteksta	<ul style="list-style-type: none"> On-line
	Formuliranje hipotez	<ul style="list-style-type: none"> H1: Mobilna aplikacija, razvita s pomočjo spletnih tehnologij, izkazuje enako enostavnost uporabe, kakovost, odzivnost in vizualno izdelavo kot njena domorodna različica. H2: Mobilna aplikacija, razvita s pomočjo spletnih tehnologij, pozitivno vpliva na zadovoljstvo uporabnikov v primerjavi z domorodno različico. H3: Mobilne aplikacije, temelječe na spletnih tehnologijah, z večjim številom zahtev v seji izkazujejo večji čas nalaganja, kar negativno vpliva na uporabniško izkušnjo. H4: Obstaja signifikantna razlika v zmogljivosti med testiranimi aplikacijami, temelječimi na spletnih tehnologijah.
	Teoretični model	<p>Slika modela</p> <p>The diagram illustrates a theoretical model with the following components and relationships:</p> <ul style="list-style-type: none"> Tip vsebine aplikacije (Application Content Type) is influenced by: Št. ostalih datotek, Št. slik, Št. JS datotek, Št. HTML datotek, Št. CSS datotek, Št. gostiteljev, Št. zahtev. Kompleksnost vsebine (Content Complexity) is influenced by: Velikost zahtev, Velikost CSS kode, Velikost HTML kode, Velikost JS kode. Uporabniška izkušnja (User Experience) is influenced by: Tip vsebine aplikacije, Kompleksnost vsebine, and a large set of user-centric metrics: % uporabnikov, ki so zadovoljni z hitrostjo aplikacije; % uporabnikov, ki so opustili aplikacijo; % uporabnikov, ki so potrdili, da funkcije zadovoljujejo njihove potrebe; % uporabnikov, ki so potrdili, da je navigirati skozi aplikacijo enostavno; % uporabnikov, ki so potrdili, da jih povrnili relevantne zadetke; % uporabnikov, ki so potrdili, da je iskano informacijo v aplikaciji; % uporabnikov, ki so ocenili aplikacijo kot vizualno prijetno; % uporabnikov, ki so ocenili vsebovane uporabniško izkušnjo kot pozitivno. Uporabniško zadovoljstvo (User Satisfaction) is influenced by: Uporabniška izkušnja, and: Identificirani atributi aplikacije, Vseplodna stopnja zadovoljstva, NTS (povprečne ocene), Afektivna in kognitivna stopnja zadovoljstva, Stopnja zadrževanja uporabnikov. Hitrost (Speed) is influenced by: Uporabniško zadovoljstvo, and: Čas izvedbe eksperimenta, Čas nalaganja (onload in DOM/contentloaded), Čas prvotne inicializacije, Tip internetne povezave, Čas odzivnosti oz. latentnost. Stabilnost (Stability) is influenced by: Hitrost, and: Konsistentnost odzivnosti, Dostopnost.

			Ime	Merska lestvica (nominalna, ordinalna, intervalna, razmernostna)	Zaloga vrednosti
		Neodvisne	Tip vsebine aplikacije	Nominalna	Št. datotek (ostalo, Slike, Javascript, HTML, CSS)
		Odvisne	Kompleksnost vsebine	Ordinalna	<ul style="list-style-type: none"> • št. gostiteljev in št. zahtev • Velikost zahteve • Velikost CSS kode • Velikost HTML kode • Velikost JS kode

			Uporabniška izkušnja	<p>Razmernost na (korak opustitve), Intervalna (Likertova lestvica: 5 – Se popolnoma strinjam, 4 – Se strinjam, ... 1- Se popolnoma ne strinjam)</p>	<ul style="list-style-type: none"> • Korak opustitve aplikacije oziroma scenarija • % uporabnikov, ki so zadovoljni s hitrostjo aplikacije • % uporabnikov, ki so potrdili, da funkcionalnost i zadovoljijo njihove potrebe • % uporabnikov, ki so potrditi, da je enostavno navigirati skozi aplikacijo • % uporabnikov, ki so potrdili, da jim je iskanje povrnilo relevantne zadetke • % uporabnikov, ki so potrdili, da je enostavno locirati iskano informacijo v aplikaciji • % uporabnikov, ki so ocenili aplikacijo kot vizualno prijetno • % uporabnikov, ki so ocenili vesplošno uporabniško izkušnjo kot pozitivno
--	--	--	----------------------	--	---

			Uporabniško zadovoljstvo	Ordinalna, intervalna (Likertova lestvica: 5 – Se popolnoma strinjam, 4 – Se strinjam, ... 1- Se popolnoma ne strinjam)	<ul style="list-style-type: none"> • Identificirani aplikacijski atributi • Vsesplošna stopnja zadovoljstva • NTS točkovanje • Afektivna in kognitivna stopnja zadovoljstva • Stopnja zadrževanja uporabnikov
			Hitrost	Nominalna	<ul style="list-style-type: none"> • Čas odzivnosti / latentnost • Čas transakcije / nalaganja • Čas prvotne inicializacije • Čas izvedbe eksperimenta • Tip internetne povezave
			Stabilnost	Nominalna	<ul style="list-style-type: none"> • Št. vrnjenih napak s strani strežnika (http napake) • Čas konsistentnosti odziva
Izbira sodelujočih	<ul style="list-style-type: none"> • Sodelujoči bodo izbrane osebe iz starostne skupine 18–55+. • Sodelovalo bo 50 oseb, ki jih bomo individualno nadzorovali v sklopu eksperimenta in opazovali. • Sodelujoči se morajo spoznati na tehnologijo in princip mobilnih aplikacij, saj je potreben zalogaj izkušenj, na podlagi katerih lahko sodelujoči ocenijo percepcijo zmogljivosti izbrane mobilne aplikacije, razvite s pomočjo spletnih tehnologij. • Predhodno znanje o mobilni aplikaciji ni potrebno. 				
Načrt eksperimenta			<i>En faktor z dvema obravnavama</i>	<ul style="list-style-type: none"> • Completely randomized • Paired comparison design 	
			<i>En faktor z več kot dvema obravnavama</i>	<ul style="list-style-type: none"> • Completely randomized • Randomized complete block design (for some blocks of subjects we consider all treatments) 	

		<i>Dva faktorja</i>	<ul style="list-style-type: none"> • 2x2 factorial design (two factors, each with two treatments) • Two-stage nested design (second factor alters for different treatments of the first)
		<i>Več kot 2 faktorja</i>	<ul style="list-style-type: none"> • 2^k factorial design • 2^k fractional factorial design
	Inštrument eksperimenta	Pripomočki in ostali predmeti	<p>Orodja, ki jih potrebujemo za izvedbo eksperimenta:</p> <ul style="list-style-type: none"> • Orodja za komuniciranje so trivialna (e-mail) in se uporabljajo samo med opazovalci • Priprava DevTools in dokumentov za zajemanje posameznih podatkov testiranja sodelujočih. <p>Pripomočki, ki jih potrebujemo za izvedbo eksperimenta:</p> <ul style="list-style-type: none"> • Osebe, ki bodo nadzorovale izvajanje eksperimenta, ter sistem za zapisovanje kakršnih koli opomb. • Navodila za izvajanje aktivnosti v fazi testiranja. Navodila so sestavljena iz enostavno izvedljivih aktivnosti, ki jih posameznik izvede znotraj izbrane mobilne aplikacije. • Števec za merjenja časa posamezne seje testiranja mobilne aplikacije. • Od sodelujočih se pričakuje, da imajo obstoječi račun na izbrani mobilni aplikacij. V primeru da ga nimajo, se izvede aktivnost za registracijo uporabnika. <p>Nepredvideni dogodki:</p> <ul style="list-style-type: none"> • Testiranje mobilne aplikacije ni možno izvesti zaradi nepredvidenih strojnih ali programskih komplikacij. • Testni vzorec sodelujočih je manjši kot pričakovani. • Lahko pride do nedelovanja testnih orodjih, zaradi česar lahko pride do izgube pridobljenih podatkov. • Nedelovanje omrežja (internet).

		<p><i>Navodila</i></p>	<p>Navodila za sodelujoče:</p> <ul style="list-style-type: none"> • V naslednjih 10 –20 minutah boste testirali mobilno aplikacijo na mobilnem aparatu, kjer boste izvedli serijo aktivnosti na podlagi katerih bomo pridobili sistematične podatke zmogljivosti in odzivnosti aplikacije. • Skrbno sledite aktivnost. V primeru ,da pride do nepričakovanega obnašanja mobilne aplikacije ali eksperimentnega okolja, prosim o tem obvestite opazovalca. • V nobenem primeru ne zapirajte mobilne aplikacije ali izvajajte več aktivnosti, kot je v vašem testnem primeru predvideno, saj lahko to močno vpliva na zajete sistematične podatke. • Po končani testni seji prosim počakajte na nadaljnja navodila opazovalca. Ta vam bo posredoval povezavo do spletne ankete. • Prosim odgovorite na spletno anketo resnično in v celoti. <p>Navodila za opazovalce:</p> <ul style="list-style-type: none"> • Opazovalec je odgovoren za aktivno sodelovanje pri opazovanju izvajanje testnih scenarijev in za pridobivanje kakršnih koli povratnih informacij, ki lahko pomagajo pri podajanju zaključne ocene zmogljivosti mobilne aplikacije. • Opazovalec lahko poseže v testiranje samo ob nedelovanju strojne ali programske opreme. V nobenem primeru opazovalec ne sme vplivati na subjektivnost anket ali objektivnost sistematičnih podatkov. • Vsakodnevno zapisovanje splošnega opazovanja je obvezno. Opazovanja se nato shranijo v skupni dokument in obravnavajo ob koncu eksperimenta.
--	--	------------------------	--

		<p><i>Merilni inštrumenti</i></p>	<p>Anketa – vprašalnik: Sodelujočim, ki so sodelovali v eksperimentu, se na koncu opravljenih aktivnosti postavi anketa za določevanje uporabniške izkušnje in zadovoljstva.</p> <p>1. Kako zadovoljni ste z uporabo aplikacije? (1 – Popolnoma nisem zadovoljen z uporabo aplikacije. 5 – Popolnoma sem zadovoljen z uporabo aplikacije.)</p> <p>2. S katerimi besedami bi opisali preizkušeno aplikacijo? (Zanesljiva, visoke kakovosti, uporabna, unikatna, nepraktična, neefektivna, slabe kakovosti, nezanesljiva)</p> <p>3. Aplikacija je popolnoma izpolnila vaša pričakovanja (1 – Se popolnoma ne strinjam. 2 – Se ne strinjam. 3 – Sem neodločen. 4 – Se strinjam. 5 – Se popolnoma strinjam.)</p> <p>4. Kako verjetno je, da bi aplikacijo priporočali prijateljem/znancem/sodelavcem? (Lestvica 1–10) – lestvica se potem enako ovrednoti kot Likertova lestvica.</p> <p>5. Aplikacija se lahko primerja z vašo idealno predstavitevjo delujoče aplikacije. (1 – Se popolnoma ne strinjam. 2 – Se ne strinjam. 3 – Sem neodločen. 4 – Se strinjam. 5 – Se popolnoma strinjam.)</p> <p>6. Z aplikacijo sem zadovoljen. (1 – Se popolnoma ne strinjam. 2 – Se ne strinjam. 3 – Sem neodločen. 4 – Se strinjam. 5 – Se popolnoma strinjam.)</p>
--	--	-----------------------------------	--

			<p>7. Aplikacija je uporabna. Enostavno je bilo uporabljati aplikacijo. Uporaba aplikacije je prijetna. (1 – Se popolnoma ne strinjam. 2 – Se ne strinjam. 3 – Sem neodločen. 4 – Se strinjam. 5 – Se popolnoma strinjam.)</p> <p>8. Odzivnost aplikacije mi je zelo pomembna. Hitrost aplikacije mi je zelo pomembna. Enostavni vmesnik mi je zelo pomemben. Veliko funkcionalnosti v aplikaciji so mi zelo pomembne. Dosegljivost in stabilno delovanje aplikacije mi je zelo pomembno. (vsa vprašanja so po enakem principu, zato so skupaj nanizana) (1 – Se popolnoma ne strinjam, 2 – Se ne strinjam, 3 – Sem neodločen, 4 – Se strinjam, 5 – Se popolnoma strinjam)</p> <p>8. Kako dolgo že uporabljate testirano aplikacijo? (Prvič, manj kot 6 mesecev, 6 – 12 mesecev, 1 – 3 let, več kot 3 leta)</p> <p>9. Težave sem imel/a pri izvajanju testnih korakov zaradi aplikacije same. (1 – Se popolnoma ne strinjam, 2 – Se ne strinjam, 3 – Sem neodločen, 4 – Se strinjam, 5 – Se popolnoma strinjam)</p> <p>10. Ocenite kvaliteto aplikacije njeno odzivnost enostavnost za uporabo glede na vašo izkušnjo (1 – zelo slabo, 2 – slabo, 3 – dobro, 4 – prav dobro, 5 – odlično)</p>
--	--	--	---

Načrt eksperimenta (nadaljevanje)	Ocenitev veljavnosti	<p>Theory</p>	
		<p><i>Slika:</i> Trochim, W. (1999). <i>The Research Methods Knowledge Base, 2nd ed.</i> Ithaca, NY: Cornell Custom Publ., Cornell University.</p>	
		<p>Veljavnosti (glej sliko zgoraj)</p>	<p>Nevarnosti veljavnosti (razložene v Wohlin, p. 67–73)</p>
	<p><i>Veljavnost sklepanja</i></p>	<p>Statistična značilnost (signifikantnost) povezave med obravnavo in rezultatom</p>	<ul style="list-style-type: none"> • Low statistical power • Violated assumption of statistical tests • Fishing and the error rate • Reliability of measures • Reliability of treatment implementation • Random irrelevancies in experimental setting • Random heterogeneity of subjects

		<i>Notranja veljavnost</i>	Povezava je rezultat faktorja	<ul style="list-style-type: none"> • History • Maturation • Testing • Instrumentation • Statistical regression • Selection • Mortality • Ambiguity about direction of causal influence • Interactions with selection • Diffusion of imitation of treatments • Compensatory equalization of treatments • Compensatory rivalry • Resentful demoralization
		<i>Veljavnost konstruktov</i>	<p>Če je povezava med vzrokom in učinkom kavzalna, potem:</p> <ul style="list-style-type: none"> • obravnava dobro odraža vpliv, • rezultat obravnave je dober pokazatelj učinka. 	<ul style="list-style-type: none"> • Inadequate preoperational explication of constructs • Mono-operation bias • Mono-method bias • Confounding constructs and levels of constructs • Interaction of different treatments • Interaction of testing and treatment • Restricted generalizability across constructs • Hypothesis guessing • Evaluation apprehension • Experiment expectancies
		<i>Zunanja veljavnost</i>	Ali lahko v primeru obstoja kavzalne povezave vplivom in učinkom, povezavo posplošimo?	<ul style="list-style-type: none"> • Interaction of selection and treatment • Interaction of setting and treatment • Interaction of history and treatment • Interaction of different treatments

<p>Izvedba eksperimenta</p>	<p>Priprava eksperimenta</p>	<ul style="list-style-type: none"> • Ustrezne posameznike bomo poiskali na podlagi prostovoljnega sodelovanja iz fakultete ali iz kroga sorodnikov in prijateljev. Starostna omejitev ne obstaja. • Velikost vzorca je 50 posameznikov, njihova skupna značilnost so izkušnje z uporabo mobilnih aplikacij. Pomembno je, da se spoznajo na uporabo mobilnih aplikacij ter kot redni uporabniki imajo zahteve ter pričakovanja na izbrano problematiko. Med izbranimi posamezniki bo naključno številko moških in žensk, saj spol posameznika ne vpliva na razumevanje in dojetje zmogljivosti mobilnih aplikacij. Starost bo od 18 do 55+. Kot že predhodno rečeno, je poznavanje mobilnih aplikacij s strani uporabnika zaželeno. • Aktivnosti, ki so potrebne za pripravo materialov, vključujejo: predhodna priprava prenosnika za testiranje in zajemanje podatkov ob uporabi mobilnih aplikacij (Google Chrome DevTools), priprava naprave za zajemanje odzivov posameznikov ob uporabi mobilnih aplikacij ter priprava papirja/naprave za zapisovanje opomb kot opazovalec. Zraven programske opreme je potrebno zagotoviti tudi delovanje omrežja kot tudi zagotoviti, da imajo uporabniki že registrirani račun na mobilnih aplikacij pripravljene za testiranje. Prehodno se tudi pripravi mobilni aparat, preko katerega se bodo testirale aplikacije in zajemale informacije. Povezava med mobilnim aparatom in prenosnikom bo potekala preko USB-povezave in tako imenovanega »remote debugging«. • Kot dodatna aktivnost se pojavi še priprava prostora za izvajanje eksperimenta (sejna soba) kot tudi priprava ankete, ki jo bomo ustvarili s pomočjo Google Forms.
------------------------------------	-------------------------------------	--

	<p>Izvedba</p>	<p>Koraki izvedbe eksperimenta:</p> <ul style="list-style-type: none"> • Pripravi se orodje za snemanje indikatorjev zmogljivosti ob uporabi mobilne aplikacije (Google DevTools). • Mobilni aparat se preko »remote debugging« priključi na prenosnik in pripravi za testiranje. • Izvajanje eksperimenta bo potekalo tako, da bo sodelujoči izvedel v naprej določene korake aktivnosti. Ob tem se bodo sistematični podatki beležili. • Opazovalec je na voljo sodelujočemu v primeru nedelovanja programske ali strojne opreme ter za odpravljanje drugih problemov, vendar ne sme subjektivno vplivati na uporabnikovo mnenje ali izvedbo. • Opazovalec tudi poskrbi, da se zabeležijo vsi podatki o testiranju in da uporabnik izvede vse predvidene aktivnosti v predvidenem zaporedju. • Opazovalec si sproti zapisuje kakršnekoli opazke glede načina izvajanja eksperimenta ali problemov, na katere je sodelujoči naletel zaradi nedelovanja mobilne aplikacije ali podobnih problemov. • Testiranje se bo izvajalo približno mesec dni od ponedeljka do petka v jutranjih in dopoldanskih urah glede na dosegljivost posameznikov. • Zapiski opazovalcev se ob koncu vsakega dneva izvajanja eksperimenta povzamejo in shranijo v dokument. • Na koncu testiranja se ti sistematični podatki povzamejo in shranijo v dokument. • Opazovalci sproti preverjajo napredke posameznikov ter poskrbijo, da se podatki pravilno zavajajo ter da eksperiment ostane pod nadzorom. • Ob končanem testiranju mobilne aplikacije se uporabniku posreduje 10–20 minut dolga anketa. • Opazovalci morajo poskrbeti, da vsak posameznik izpolni anketo pravilno in poda resnične podatke (eliminiranje posameznika ob primeru, da je ta vzel anketo neresno ter podal neresnične, nerealne podatke in informacije). • Rezultati ankete se avtomatično shranijo na strežnik in kasneje uporabijo za pridobivanje relevantnih podatkov. <p>Postopek zbiranja podatkov:</p> <ul style="list-style-type: none"> • Podatki se bodo zbirali redno, vsak aktiven dan izvajanja eksperimenta v času izvajanja. Po vsaki končani seji testiranja se zajemajo sistematični podatki posameznika in njegove seje ter se ažurirajo in shranijo v dokument. • Ob koncu eksperimenta se bodo pregledale oddane ankete in filtrirale glede na statistično uporabnost. Odstranile se bodo tiste, ki se bodo izkazale za lažne ali neveljavne.
	<p>Preverjanje podatkov</p>	<ul style="list-style-type: none"> • Preverite, ali so podatki smiselni. • Preverite ali so bili podatki pravilno pridobljeni. • Preverite, ali so bile posamezne obravnave izpeljane v pravilnem vrstnem redu in skladno z načrtom.

Priloga C – Grafi in tabele statističnih analiz mobilnih aplikacij

Tabela 8.1: Notranje korelacije med elementi kognitivnega zadovoljstva

	Odzivnost aplikacije	Hitrost aplikacije	Zadovoljstvo ob uporabi aplikacije	Uporabna aplikacija	Uporaba je intuitivna in prijetna
Odzivnost aplikacije	1,000	,810	,684	,651	,600
Hitrost aplikacije	,810	1,000	,680	,649	,671
Zadovoljstvo ob uporabi aplikacije	,684	,680	1,000	,715	,723
Uporabna aplikacije	,651	,649	,715	1,000	,643
Uporaba je intuitivna in prijetna	,600	,671	,723	,643	1,000

Tabela 8.2: Celotna statistična postavka elementov kognitivnega zadovoljstva

	Povprečje, če izbrišemo element	Stopnja odstopanja, če izbrišemo element	Popravljen korelacija	Kvadratna korelacija	α , če izbrišemo element
Odzivnost aplikacije	15,04	14,266	,784	,697	,887
Hitrost aplikacije	15,19	13,791	,804	,715	,882
Zadovoljstvo ob uporabi aplikacije	15,34	11,524	,807	,667	,889
Uporabna aplikacija	15,10	14,720	,760	,584	,893
Uporaba je intuitivna in prijetna	15,44	13,716	,756	,596	,891

Tabela 8.3: Celotna statistična postavka elementov uporabniške izkušnje

	Povprečje, če izbrišemo element	Stopnja odstopanj, če izbrišemo element	Popravljen korelacija	Kvadratna korelacija	α , če izbrišemo element
Hitrost aplikacije	22,38	29,288	,721	,553	,884
Zadovoljstvo ob uporabi aplikacije	22,58	26,044	,840	,744	,868
Z navigiranjem nisem imel/a težav	22,26	31,100	,624	,494	,894
Preko iskanja sem našel/a iskane rezultate	22,26	32,455	,523	,413	,903
Enostavno locirati iskane elemente	22,38	30,721	,641	,444	,893
Vizualna izdelava aplikacije	22,89	24,987	,783	,695	,878
Zadovoljstvo ob uporabi aplikacije	22,52	24,731	,858	,798	,866

Tabela 8.4: Notranje korelacije med elementi afektivnega zadovoljstva

	Pomembnost odzivnosti aplikacije	Pom. hitrosti aplikacije	Število funk.	Pom. enost. vm.	Pom. stab. in odz.
Pomembnost odzivnosti aplikacije	1,000	,498	-,013	,311	,137
Pomembnost hitrosti aplikacije	,498	1,000	,038	,529	,167
Število funkcionalnosti	-,013	,038	1,000	,049	-,070
Pomembnost enostavnega vmesnika	,311	,529	,049	1,000	,266
Pomembnost stabilnosti in odzivnosti	,137	,167	-,070	,266	1,000

Tabela 8.5: Celotna statistična postavka elementov afektivnega zadovoljstva

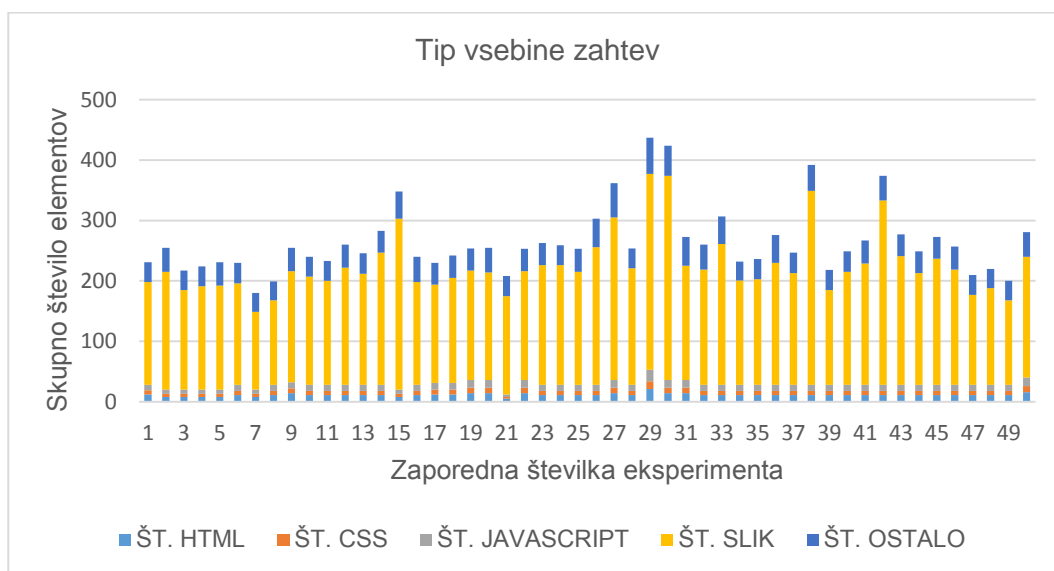
	Povprečje, če izbrišemo element	Stopnja odstopanja, če izbrišemo element	Popravljen korelacija	Kvadratna korelacija	α , če izbrišemo element
Pomembnost odzivnosti aplikacije	15,48	2,456	,275	,254	,237
Pomembnost hitrosti aplikacije	15,47	2,365	,408	,403	,167
Število funkcionalnosti	16,75	1,881	-,004	,011	,635
Pomembnost enostavnega vmesnika	15,49	2,436	,391	,316	,188
Pomembnost stabilnosti in odzivnosti	15,59	2,659	,115	,081	,339

Tabela 8.6: Novi izračun notranje korelacije med elementi afektivnega zadovoljstva

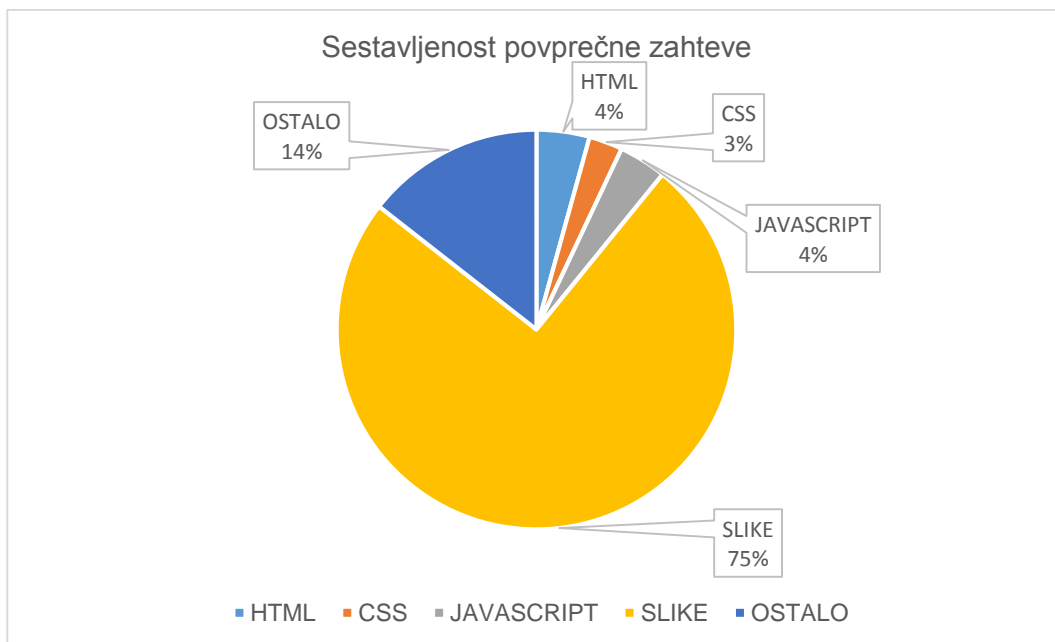
	Pomembnost odzivnosti aplikacije	Pomembnost hitrosti aplikacije	Pomembnost enostavnega vmesnika
Pomembnost odzivnosti aplikacije	1,000	,498	,311
Pomembnost hitrosti aplikacije	,498	1,000	,529
Pomembnost enostavnega vmesnika	,311	,529	1,000

Tabela 8.7: Novi izračun statistične postavke elementov afektivnega zadovoljstva

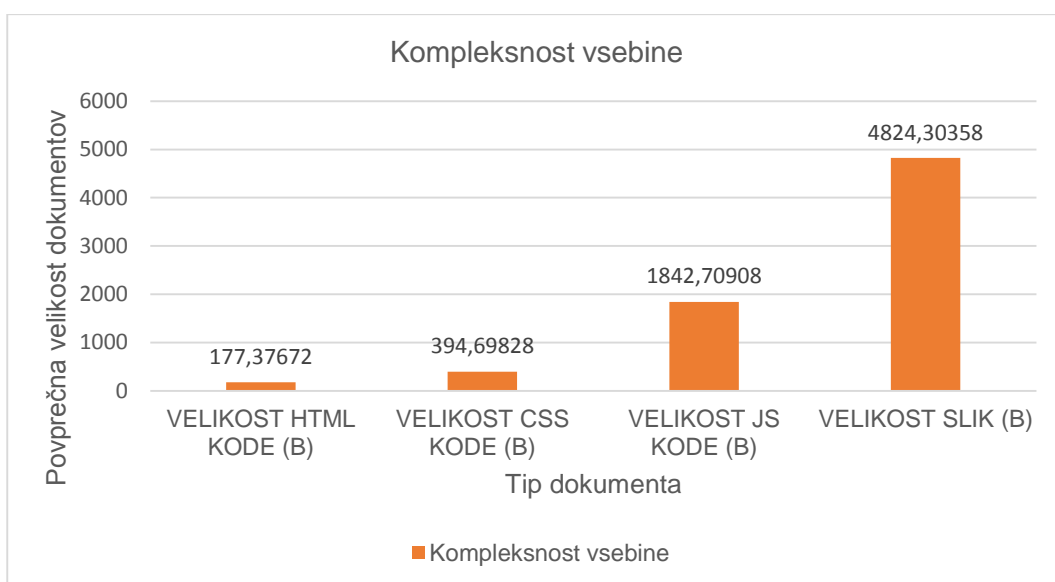
	Povprečje, če izbrišemo element	Stopnja odstopanja, če izbrišemo element	Popravljen korelacija	Kvadratna korelacija	α , če izbrišemo element
Pomembnost odzivnosti aplikacije	8,43	,632	,466	,251	,691
Pomembnost hitrosti aplikacije	8,42	,605	,631	,403	,470
Pomembnost enostavnega vmesnika	8,44	,728	,478	,282	,663



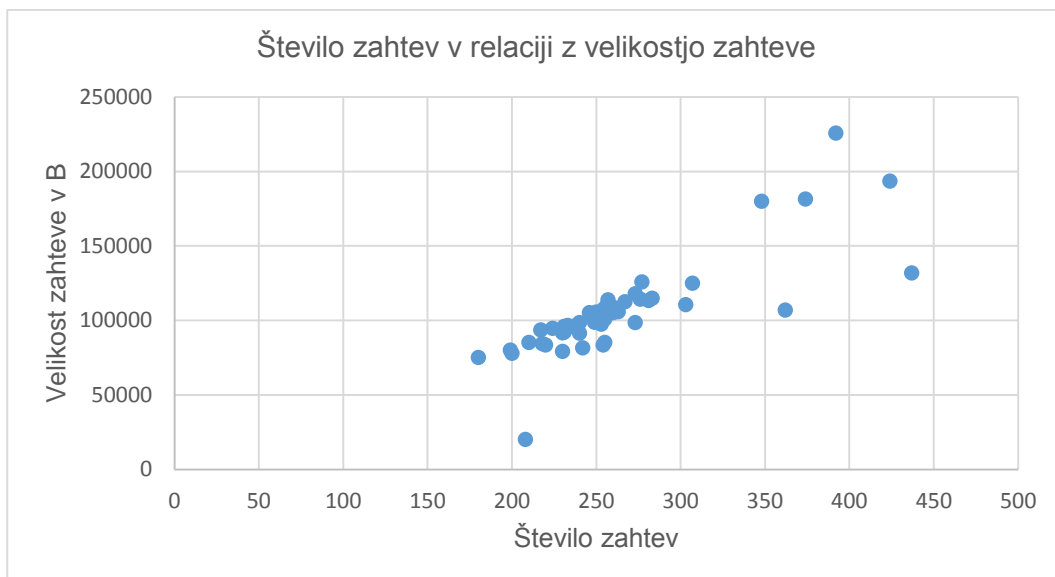
Slika 8.1: Prikaz sestavljenosti posameznih zahtev vsakega udeleženca – Twitter



Slika 8.2: Prikaz sestavnih delov po velikosti posamezne zahteve – Twitter



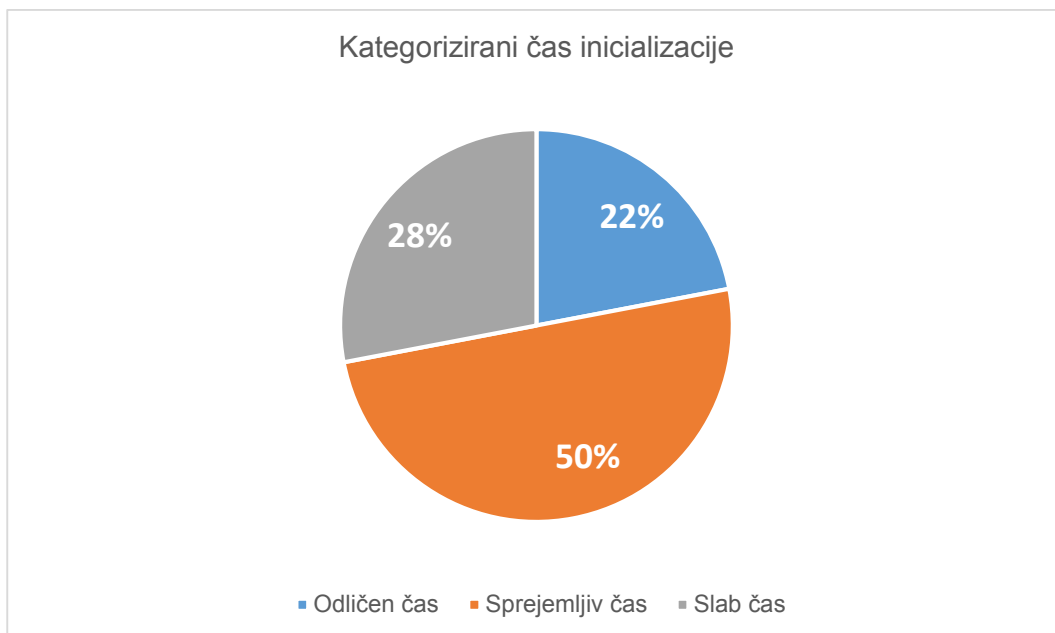
Slika 8.3: Kompleksnost vsebine pri spletni različici Twitter aplikacije



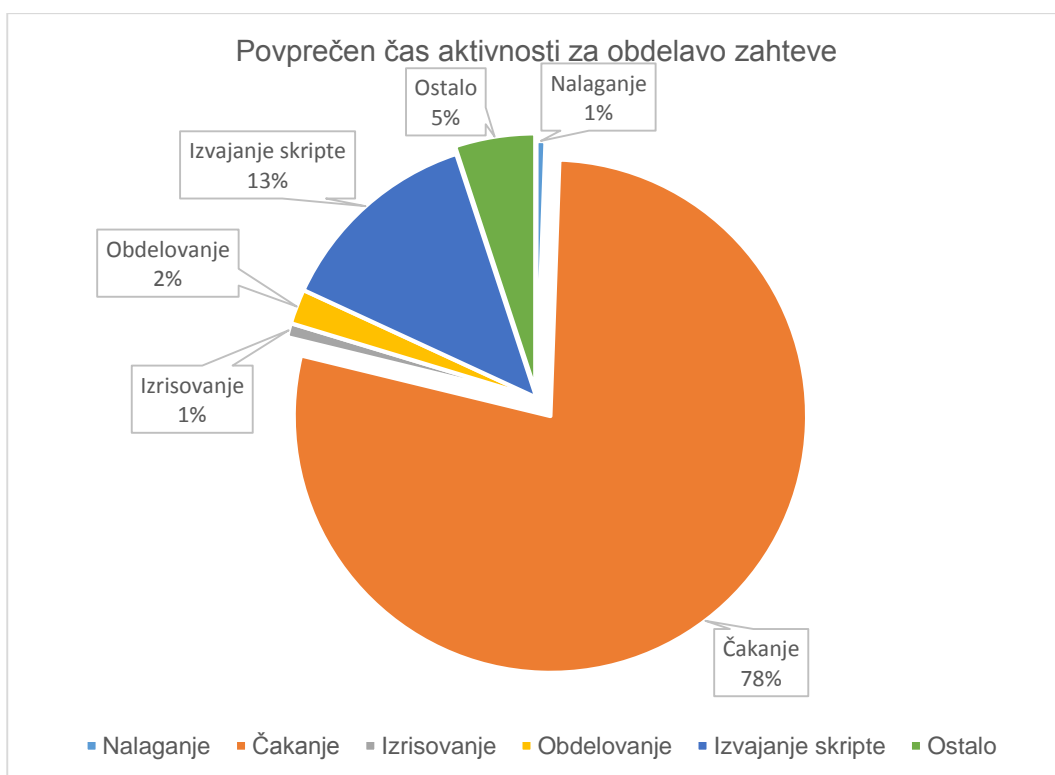
Slika 8.4: Povprečne vrednosti števila zahtev v korelaciji z njihovo velikostjo – Twitter



Slika 8.5: Prikazovanje inicializacijskega časa Twitter aplikacije



Slika 8.6: Kategorizirani čas inicializacije Twitter aplikacije



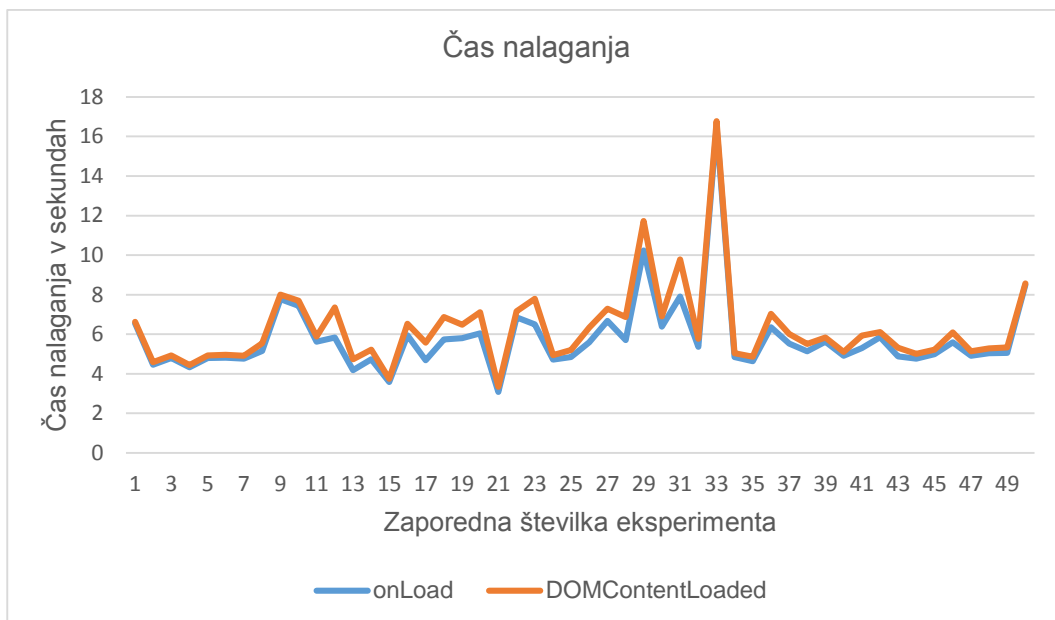
Slika 8.7: Povprečen čas porabljen v brskalniku – Twitter aplikacija



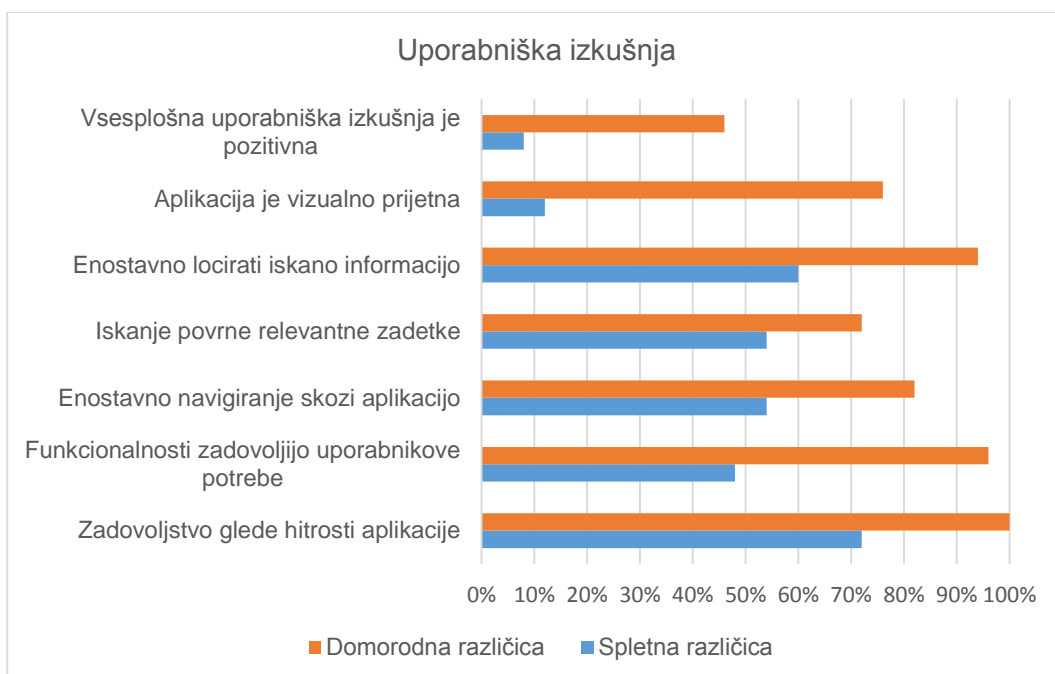
Slika 8.8: Dostopnost Twitter aplikacije pri posamezni seji udeleženca



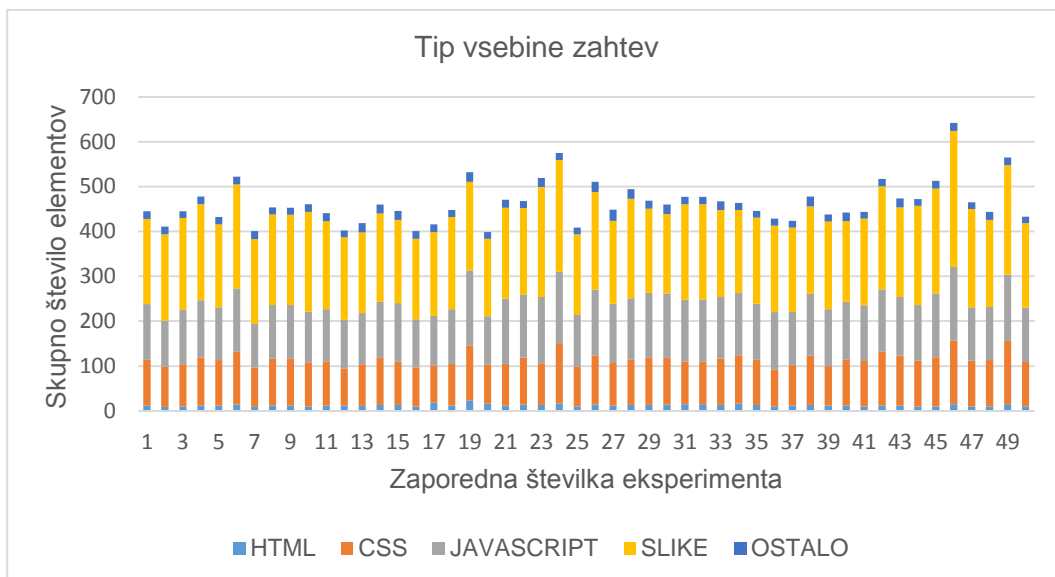
Slika 8.9: Konsistentnost odziva Twitter aplikacije v posamezni seji udeleženca



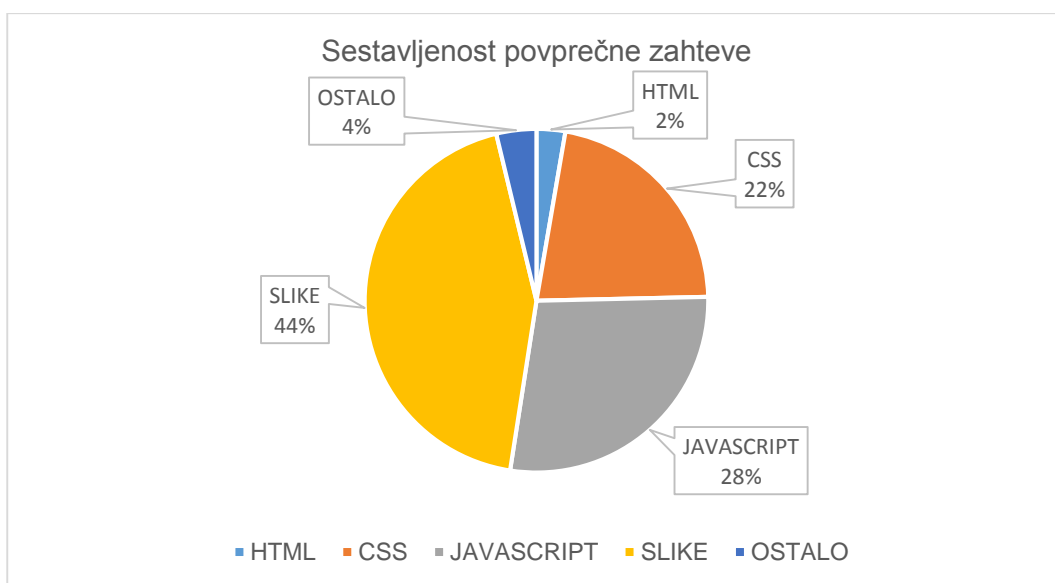
Slika 8.10: Čas nalaganja Twitter



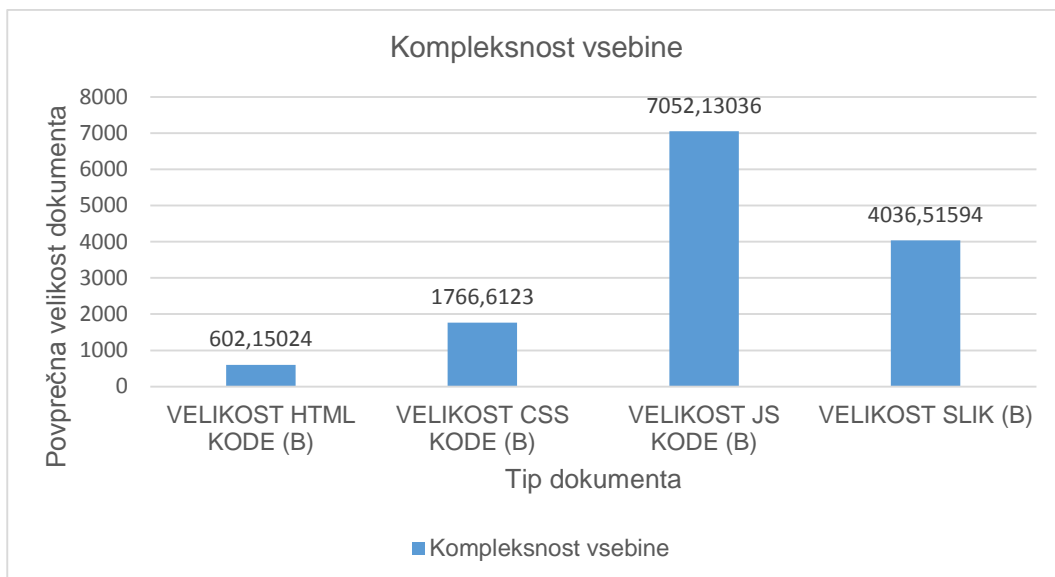
Slika 8.11: Uporabniška izkušnja pri Twitter aplikaciji



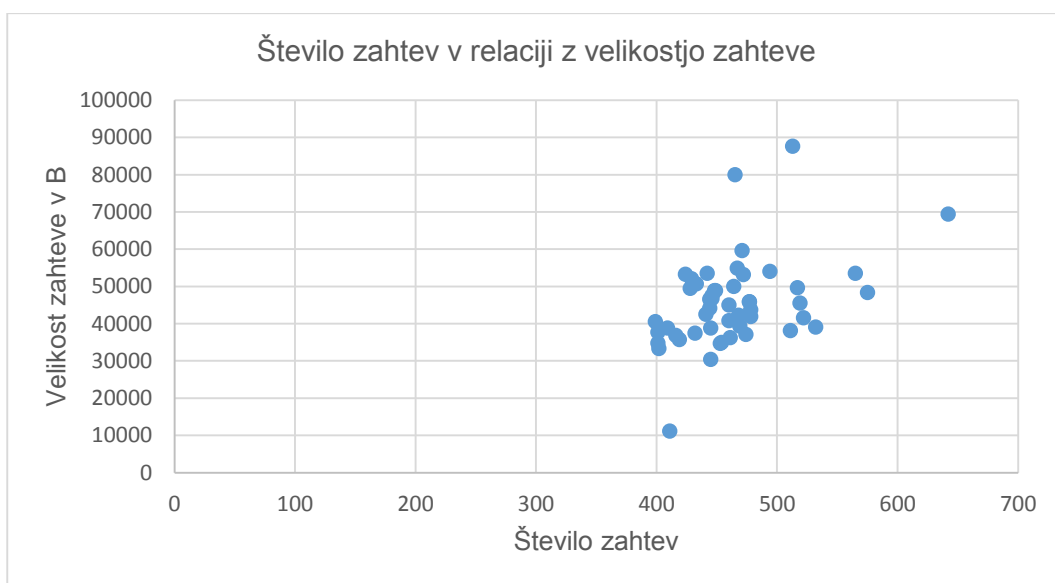
Slika 8.12: Prikaz sestavljenosti posameznih zahtev vsakega udeleženca – Facebook



Slika 8.13: Prikaz sestavnih delov po velikosti posamezne zahteve – Facebook



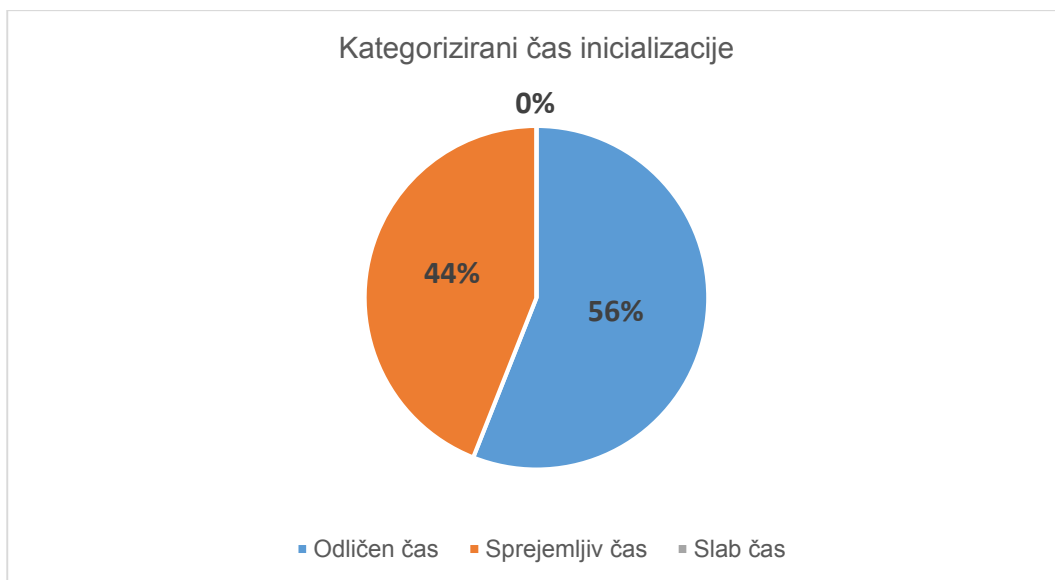
Slika 8.14: Kompleksnost vsebine pri spletni različici Facebook aplikacije



Slika 8.15: Povprečne vrednosti števila zahtev v korelaciji z njihovo velikostjo – Facebook



Slika 8.16: Prikazovanje inicializacijskega časa Facebook aplikacije



Slika 8.17: Kategorizirani čas inicializacije Facebook aplikacije



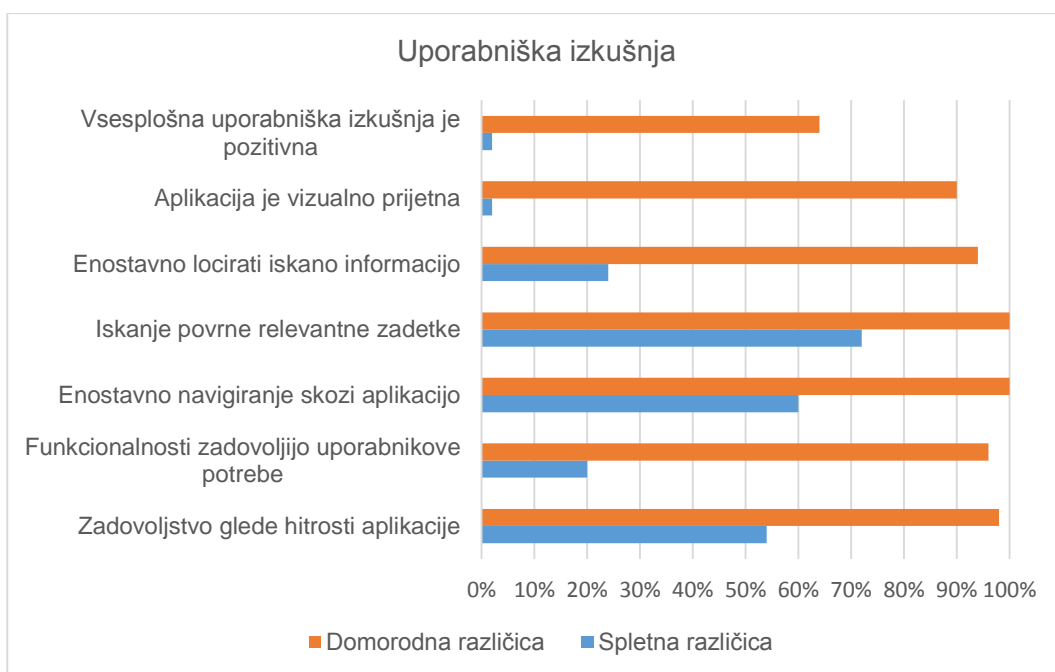
Slika 8.18: Povprečen čas porabljen v brskalniku - Facebook aplikacija



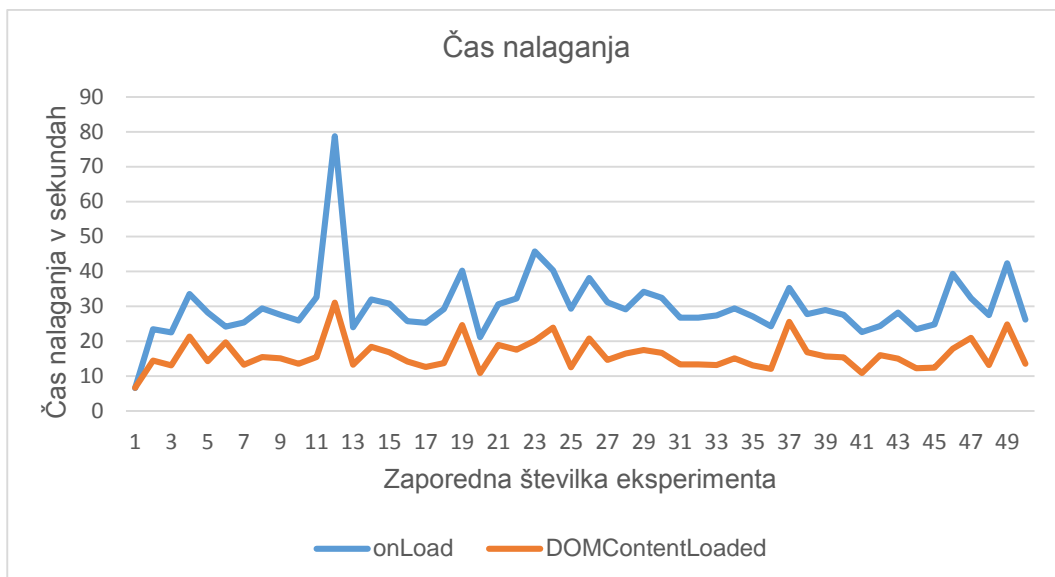
Slika 8.19: Dostopnost Facebook aplikacije pri posamezni seji udeleženca



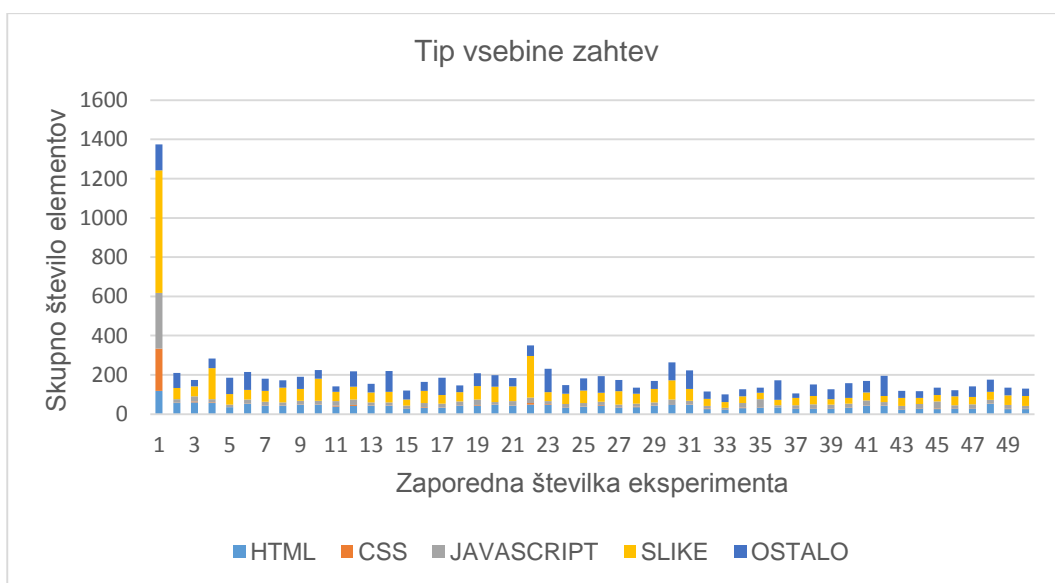
Slika 8.20: Konsistentnost odziva Facebook aplikacije v posamezni seji udeleženca



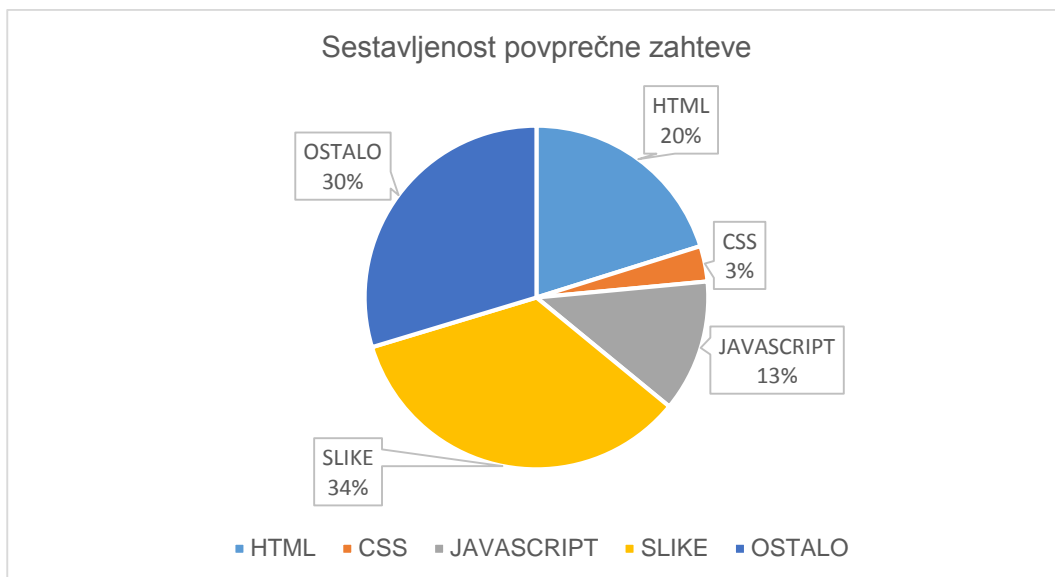
Slika 8.21: Uporabniška izkušnja pri Facebook aplikaciji



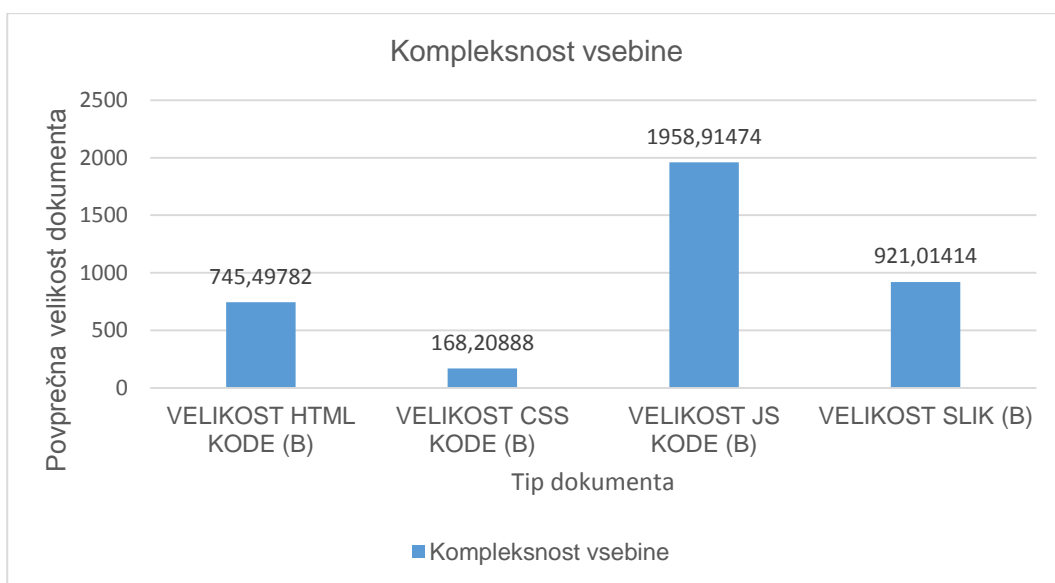
Slika 8.22: Čas nalaganja Facebook



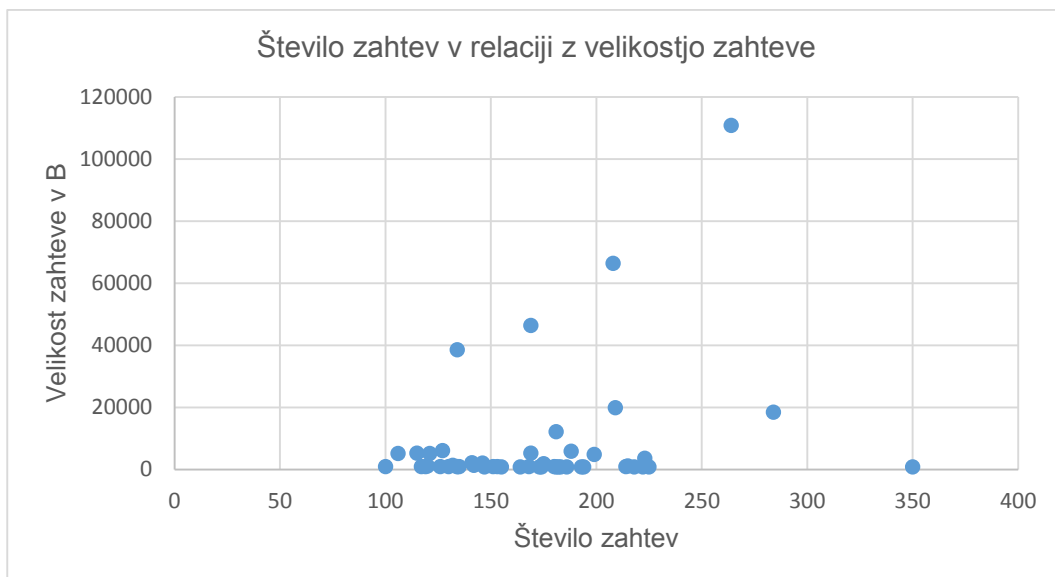
Slika 8.23: Prikaz sestavljenosti posameznih zahtev vsakega udeleženca – YouTube



Slika 8.24: Prikaz sestavnih delov po velikosti posamezne zahteve – YouTube



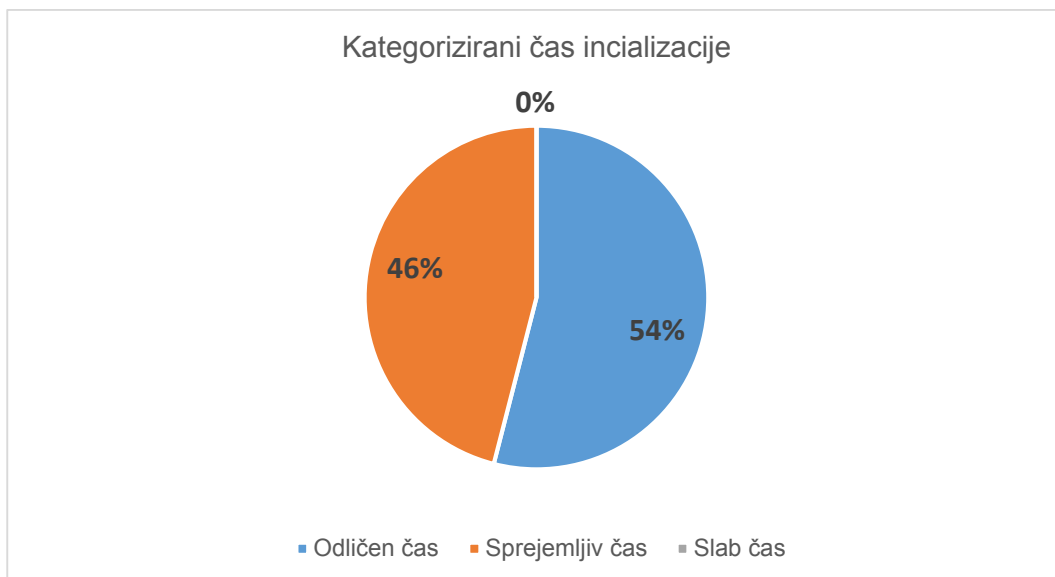
Slika 8.25: Kompleksnost vsebine pri spletni različici YouTube aplikacije



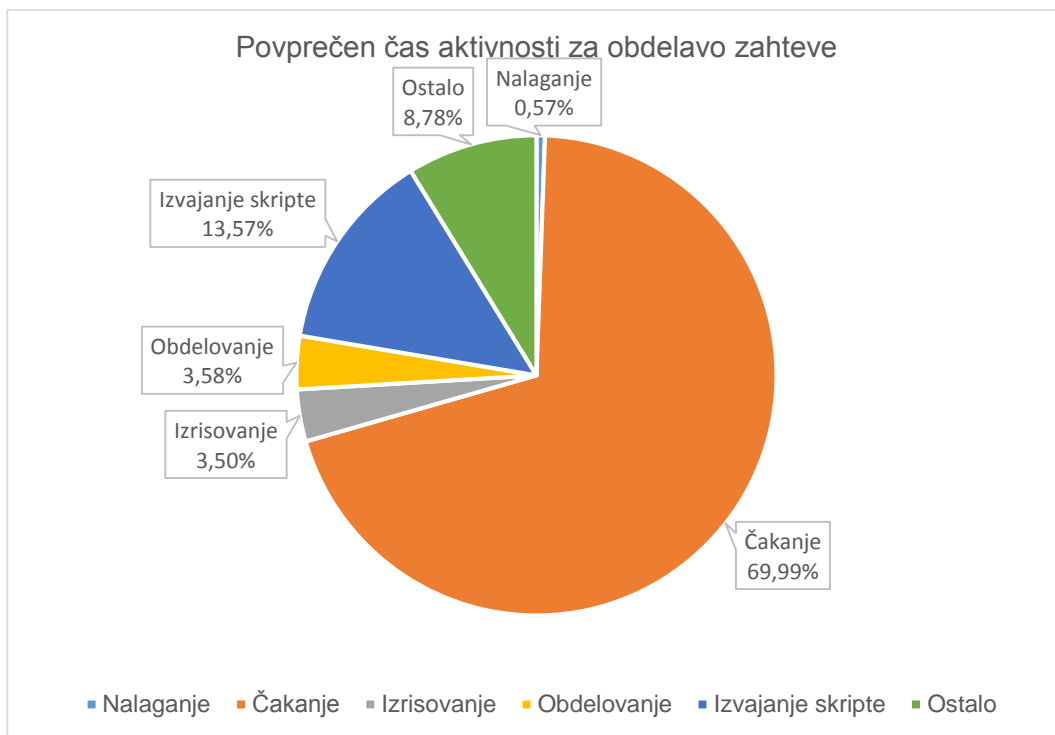
Slika 8.26: Povprečne vrednosti števila zahtev v korelaciji z njihovo velikostjo – YouTube



Slika 8.27: Prikazovanje inicializacijskega časa YouTube aplikacije



Slika 8.28: Kategorizirani čas inicializacije YouTube aplikacije



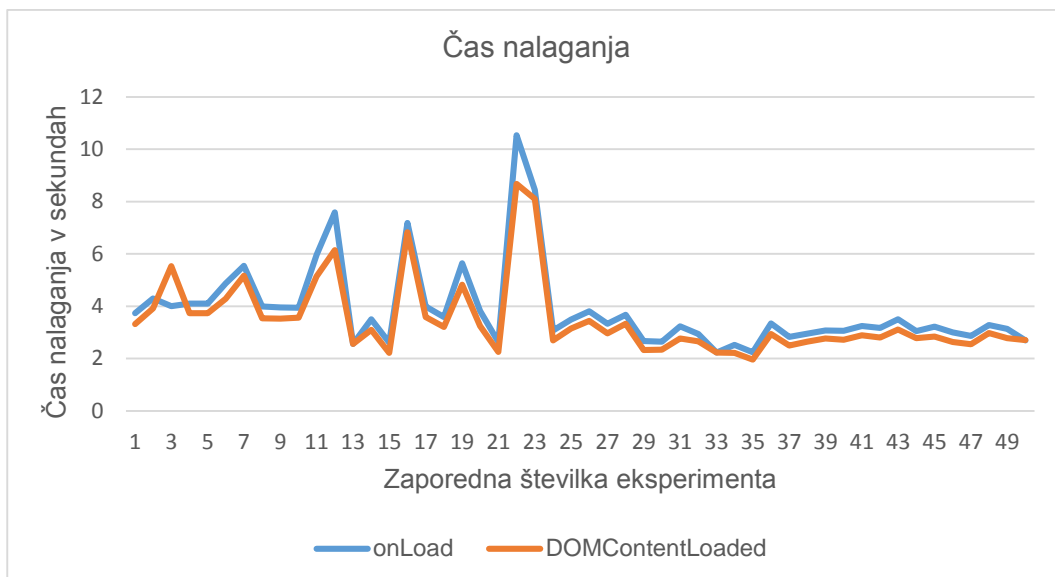
Slika 8.29: Povprečen čas porabljen v brskalniku - YouTube aplikacija



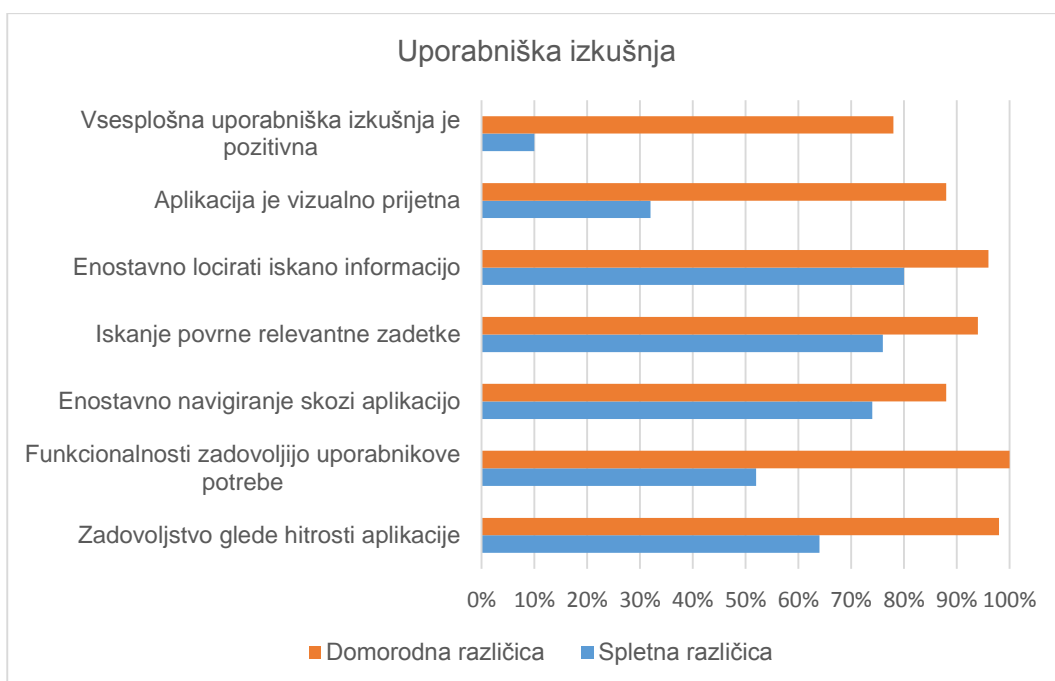
Slika 8.30: Dostopnost YouTube aplikacije pri posamezni seji udeleženca



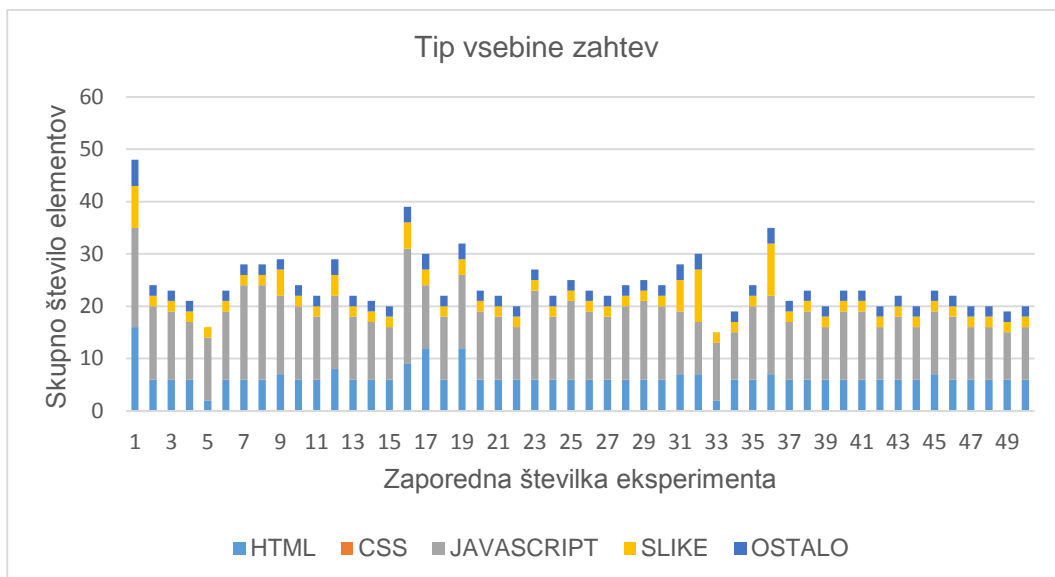
Slika 8.31: Konsistentnost odziva YouTube aplikacije v posamezni seji udeleženca



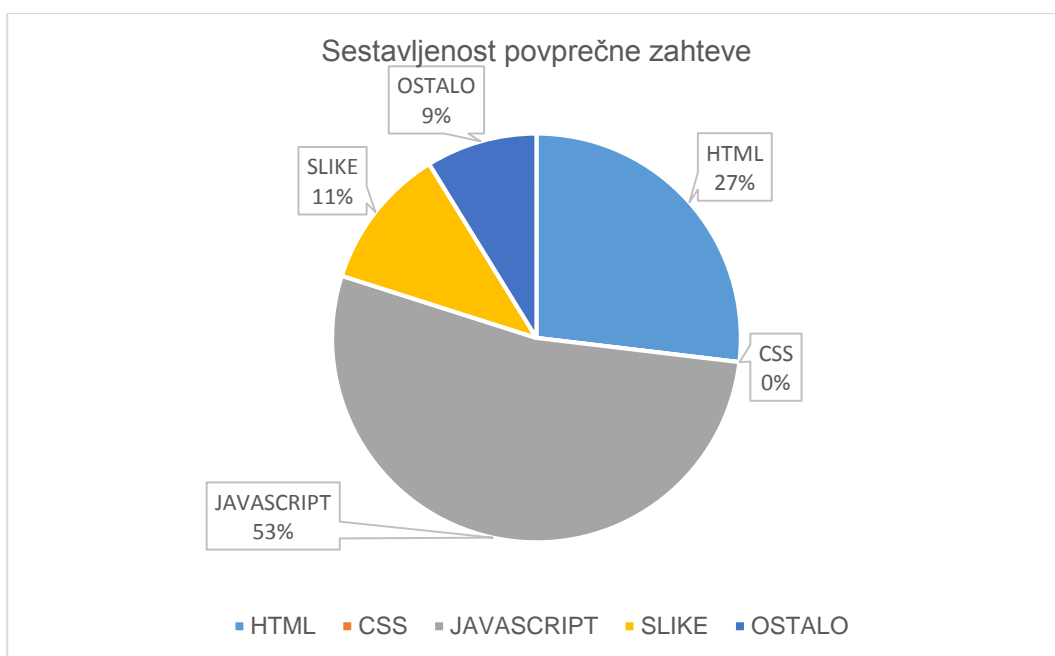
Slika 8.32: Čas nalaganja YouTube



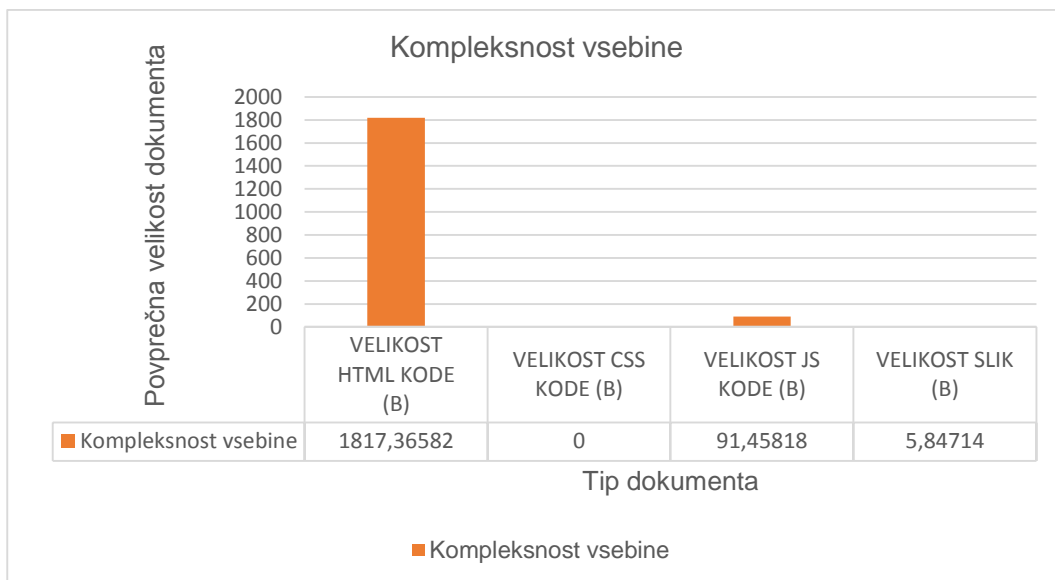
Slika 8.33: Uporabniška izkušnja YouTube aplikacije



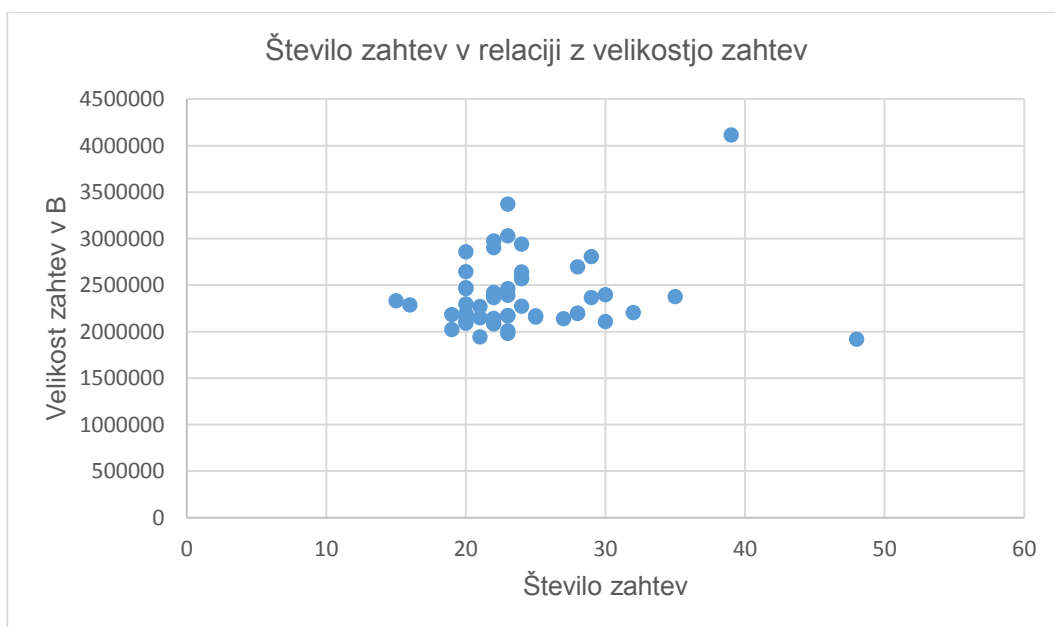
Slika 8.34: Prikaz sestavljenosti posameznih zahtev vsakega udeleženca – GMail



Slika 8.35: Prikaz sestavnih delov po velikosti posamezne zahteve – GMail



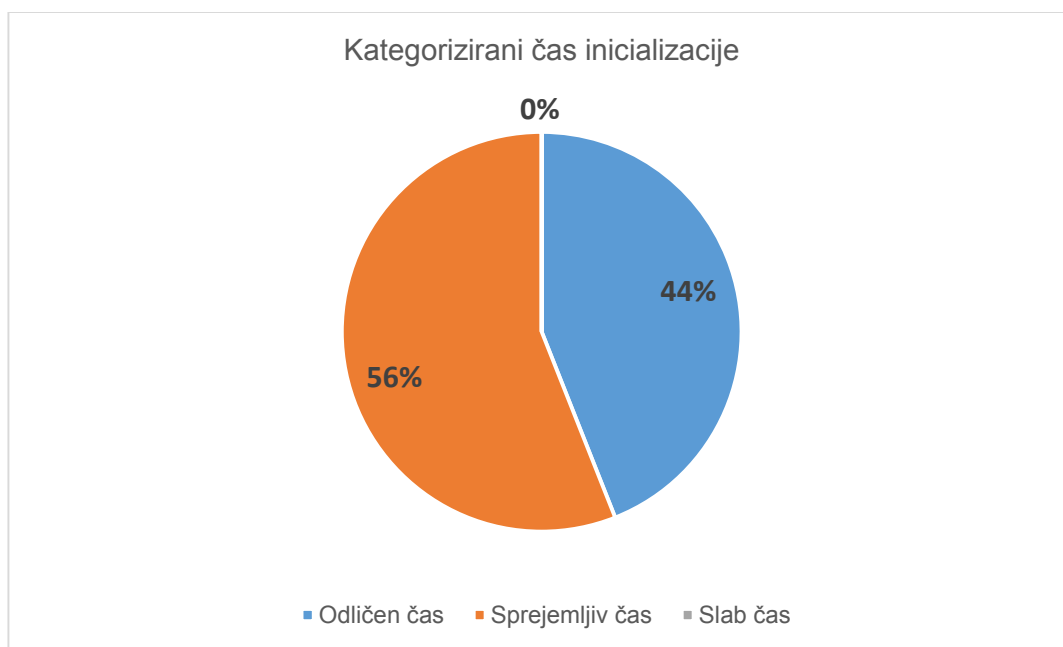
Slika 8.36: Kompleksnost vsebine pri spletni različici GMail aplikacije



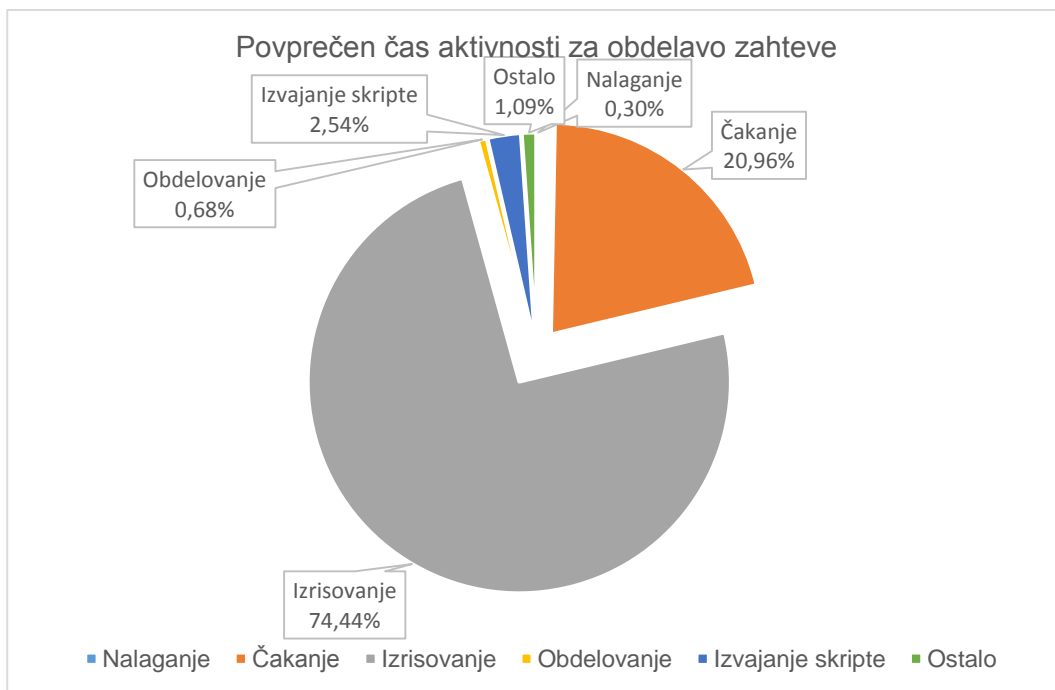
Slika 8.37: Povprečne vrednosti števila zahtev v korelaciji z njihovo velikostjo – GMail



Slika 8.38: Prikazovanje inicializacijskega časa GMail aplikacije



Slika 8.39: Kategorizirani čas inicializacije GMail aplikacije



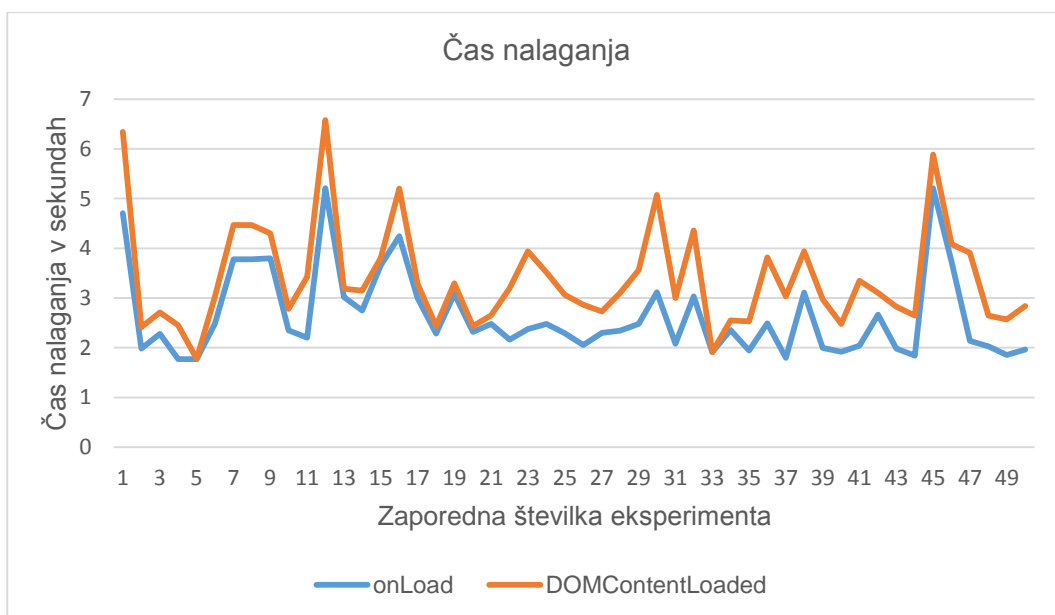
Slika 8.40: Povprečen čas porabljen v brskalniku - GMail aplikacija



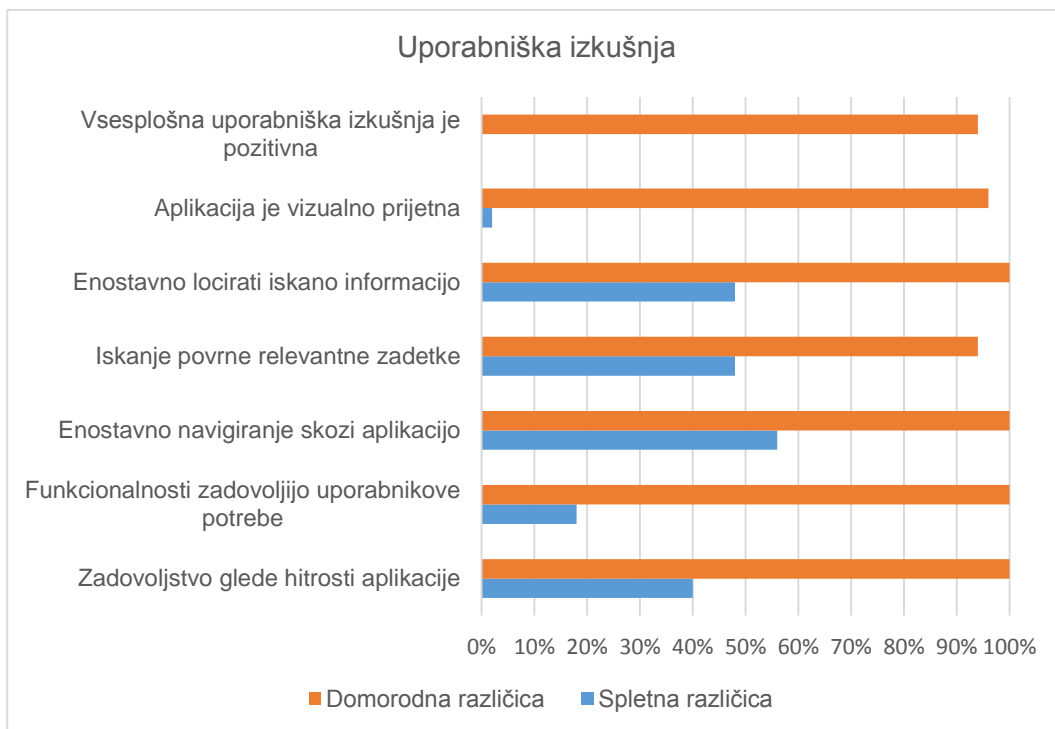
Slika 8.41: Dostopnost GMail aplikacije pri posamezni seji udeleženca



Slika 8.42: Konsistentnost odziva Gmail aplikacije v posamezni seji udeleženca



Slika 8.43: Čas nalaganja Gmail



Slika 8.44: Uporabniška izkušnja GMail aplikacije



Fakulteta za elektrotehniko,
računalništvo in informatiko



IZJAVA O AVTORSTVU

Spodaj podpisani/-a ANDREJA FAJFAR
z vpisno številko E5012589
sem avtor/-ica magistrskega dela z naslovom: _____

ANALIZA ZMOGLJIVOSTI MOBILNIH APLIKACIJ, TEMELJEČIH NA SPLETNIH
TEHNOLOGIJAH
(naslov magistrskega dela)

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

doc. dr. Boštjan Šumak, univ. dipl. inž. rač. in inf.

in somentorstvom (naziv, ime in priimek)

/

- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela.
- soglašam z javno objavo elektronske oblike magistrskega dela v DKUM.

V Mariboru, dne 21.04.2016

Podpis avtorja/-ice:

Andreja Fajfar



Fakulteta za elektrotehniko,
računalništvo in informatiko



IZJAVA O USTREZNOSTI ZAKLJUČNEGA DELA

Podpisani mentor :

BOŠTJAN ŠIMAK

(ime in priimek mentorja)

in somentor (eden ali več, če obstajata):

(ime in priimek somentorja)

Izjavljam (-va), da je študent

Ime in priimek: ANDREJA FAJFAR

Vpisna številka: E5012589

Na programu: INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA

izdelal zaključno delo z naslovom:

ANALIZA ZMOGLJIVOSTI MOBILNIH APLIKACIJ, TEMELJEČIH NA SPLETNIH
(naslov zaključnega dela v slovenskem in angleškem jeziku) TEHNOLOGIJAH

PERFORMANCE ANALYSIS OF MOBILE APPLICATIONS BASED ON WEB
TECHNOLOGIES

v skladu z odobreno temo zaključnega dela, Navodilih o pripravi zaključnih del in mojimi (najinimi oziroma našimi) navodili.

Preveril (-a, -i) in pregledal (-a, -i) sem (sva, smo) poročilo o plagiatorstvu.

Datum in kraj: 25.4.2016, MARIBOR

Podpis mentorja: _____

Datum in kraj:

Podpis somentorja (če obstaja):

Priloga:

- Poročilo o preverjanju podobnosti z drugimi deli.



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko



**IZJAVA O ISTOVETNOSTI TISKANE IN ELEKTRONSKE VERZIJE ZAKLJUČNEGA
DELA IN OBJAVI OSEBNIH PODATKOV DIPLOMANTOV**

Ime in priimek avtorja-ice: ANDREJA FAJFAR

Vpisna številka: E5012589

Študijski program: INFORMATIKA IN TEHNOLOGIJE KOMUNICIRANJA

Naslov zaključnega dela: ANALIZA ZMOGLJIVOSTI MOBILNIH APLIKACIJ,
TEMELJEČIH NA SPLETNIH TEHNOLOGIJAH

Mentor: BOŠTJAN ŠUMAK

Somentor: _____

Podpisani-a ANDREJA FAJFAR izjavljam, da sem za potrebe arhiviranja oddal elektronsko verzijo zaključnega dela v Digitalno knjižnico Univerze v Mariboru. Zaključno delo sem izdelal-a sam-a ob pomoči mentorja. V skladu s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovoljujem, da se zgoraj navedeno zaključno delo objavi na portalu Digitalne knjižnice Univerze v Mariboru.

Tiskana verzija zaključnega dela je istovetna z elektronsko verzijo elektronski verziji, ki sem jo oddal za objavo v Digitalno knjižnico Univerze v Mariboru.

Zaključno delo zaradi zagotavljanja konkurenčne prednosti, varstva industrijske lastnine ali tajnosti podatkov naročnika: _____ ne sme biti javno dostopno do _____ (datum odloga javne objave ne sme biti daljši kot 3 leta od zagovora dela).

Podpisani izjavljam, da dovoljujem objavo osebnih podatkov, vezanih na zaključek študija (ime, priimek, leto in kraj rojstva, datum zaključka študija, naslov zaključnega dela), na spletnih straneh in v publikacijah UM.

Datum in kraj: 25.04.2016, MARIBOR Podpis avtorja-ice: Andreja Fajfar

Podpis mentorja: _____
(samo v primeru, če delo ne sme biti javno dostopno)

Podpis odgovorne osebe naročnika in žig: _____
(samo v primeru, če delo ne sme biti javno dostopno)