

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gašper Kolenc

**Razvoj, izdaja in trženje iOS
aplikacije za Apple App Store**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Matija Marolt

Ljubljana 2013

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

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00078/2013

Datum: 05.04.2013

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

Kandidat: **GAŠPER KOLENC**

Naslov: **RAZVOJ, IZDAJA IN TRŽENJE IOS APLIKACIJE ZA APPLE APP STORE**

DEVELOPMENT, RELEASE AND MARKETING OF AN IOS APPLICATION IN APPLE APP STORE

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

V diplomski nalogi obravnavajte razvoj in izdajo iOS aplikacije za Apple App Store. Opišite posebnosti pri razvoju tovrstnih aplikacij, vključno s prilagajanjem napravam, zaslonom, lokalizaciji in integraciji v socialna omrežja. Prav tako na primeru razdelajte primerne pristope k izdaji in poslovnemu modelu trženja, oglaševanja in komunikacije z uporabniki.

Mentor:


doc. dr. Matija Marolt



Dekan:


prof. dr. Nikolaj Zimic

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Gašper Kolenc, z vpisno številko **63100220**, sem avtor diplomskega dela z naslovom:

Razvoj, izdaja in trženje iOS aplikacije za Apple App Store

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne 22. julija 2013

Podpis avtorja:

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Uporabljene tehnologije in oprema	3
2.1	iOS	3
2.2	Xcode	7
2.3	Objective-C	8
3	Razvoj aplikacije	9
3.1	Nail Designs	9
3.2	Prilagajanje večjemu zaslonu iPada	9
3.3	Internacionalizacija in lokalizacija	12
3.4	Integracija socialnih omrežij	14
3.5	Izdelava slik manikur	15
3.6	Prehod na brezplačni model s plačljivo dodatno vsebino	19
3.7	Raba analitičnih orodij in orodij za poročanje napak	21
4	Izdaja in trženje aplikacije	23
4.1	Testiranje	23
4.2	Pregled aplikacije s strani Appla	24
4.3	Izdelava spletne strani	26
4.4	Integracija socialnih omrežij	27

KAZALO

4.5	Začasno zmanjšanje cene aplikacije	28
4.6	Pridobivanje ocen zadovoljnih uporabnikov	28
4.7	Oglaševanje preko ostalih aplikacij	29
5	Sklepne ugotovitve	31
	Literatura	31

Povzetek

Razvijanje za iOS mobilni operacijski sistem se v zadnjih letih izredno širi. AppStore namreč nudi zelo perspektiven trg za prodajo aplikacij tako velikih kot tudi majhnih razvijalcev.

To diplomsko delo obravnava razvoj in izdajo iOS aplikacije Nail Designs za AppStore. Ob razvijanju aplikacije sem se soočil s pogostimi težavami, na katere lahko naletimo pri razvijanju za iOS. Med drugim kako se prilagaja aplikacija večjemu zaslonu iPada, lokalizacija aplikacije, integriranje socialnih omrežij, ipd. Vse to in še več je zajeto znotraj tega dela.

Programiranje aplikacije pa je le ena stran razvijanja za iOS. V ozadju stojijo še poslovni model trženja, oglaševanje, komunikacija z uporabniki, pridobivanje dobrih ocen in še kopica preostalih stvari. V tem diplomskem delu tako ne obravnavam le razvoja aplikacije za iOS, temveč tudi opisujem stvari, ki se dogajajo v zaledju.

Ključne besede: iOS, razvoj, izdaja, aplikacija.

Abstract

iOS development is getting more and more popular over the last few years. A huge factor in this growing industry is AppStore's indifference between big and small developers.

This thesis deals with development and release of an iOS application named Nail Designs. I faced many issues one can face while developing this application. From adapting application's view sizes to accommodate iPad's larger screen size, to localization and integration of social networks. All this and more is covered within this thesis.

But developing an application is just one side of the coin that is iOS development. There are also business models behind the app, marketing, user communication and so forth. So in this thesis, I don't only cover the development cycle of Nail Designs, but also dig into the little things that run in the background.

Keywords: iOS, development, release, application.

Poglavje 1

Uvod

Trgovina za iOS aplikacije App Store je bila predstavljena javnosti 10. julija 2008. Timothy Cook, trenutni izvršni direktor Appl, je na konferenci WWDC 2013 naznanil, da je bilo izvedenih že 35 milijard prenosov aplikacij iz več kot 575 milijonov Apple ID uporabniških računov. Za samo predstavo o hitri rasti App Stora: Apple, Inc je ravijalcem do junija 2013 izplačal 10 milijard dolarjev, od tega 5 milijard le v zadnjem letu.[1]

Datum	Število aplikacij	Število prenosov
11.7.2008	500	0
22.10.2008	7.500	200.000.000
4.11.2009	100.000	2.000.000.000+
20.10.2010	300.000+	7.000.000.000
4.10.2011	500.000+	18.000.000.000+
11.6.2012	650.000+	30.000.000.000+
10.6.2013	900.000+	50.000.000.000+

Tabela 1.1: Rast App Stora skozi leta.

Prav s pomočjo App Stora in temu pripadajočih naprav - iPhone, iPod in iPad se je Apple v zadnjih letih povzpел na trenutno 15. mesto na Forbes lestvici največjih podjetij na svetu in je na prvem mestu v svoji kategoriji

računalniške strojne opreme, hkrati pa zaseda 2. mesto med najbolj donosnimi podjetji in 1. mesto pri tržni vrednosti. Njihov največji rival Microsoft mu sledi šele na 41. mestu največjih podjetij.[2]

Vsa ta števila so v zadnjih letih povzročila kar nekakšno norijo med razvijalci, vsak namreč hoče svoj kos pogače. Tako se je začelo povpraševanje po razvijalcih za iOS večati in s tem tudi njihovo število. V tem besedilu bom podrobno predstavil proces razvijanja, trženja in izdaje svoje lastne iOS aplikacije Nail Designs.

Poglavje 2

Uporabljene tehnologije in oprema

2.1 iOS

iOS je mobilni operacijski sistem, ki ga uporabljajo izključno le Apple naprave. Predstavljen je bil leta 2007, izključno le za mobilne telefone iPhone, nakar se je nabor podprtih naprav z leti povečal. Med podprte naprave sedaj spadajo mobilni telefoni iPhoni, mp3 glasbeni predvajalniki iPodi, tablični računalniki iPadi in digitalni sprejemniki Apple TV.

Glavna razlika v primerjavi s konkurenčnima mobilnima operacijskima sistemoma Android in Windows Phone je v tem, da iOS ne podpira namestitve na strojno opremo, ki ni izdana s strani Applla.

iOS izhaja iz operacijskega sistema OS X, s katerim oba bazirata na Darwin osnovi. Darwin je odprtokoden računalniški operacijski sistem skladen s POSIX standardi, ki ga je izdal Apple leta 2000.

2.1.1 Pogoji za razvijanje

Za razvoj aplikacij se potrebuje Intel Mac osebni računalnik, ki poganja operacijski sistem Mac OS X Leopard ali novejši. Prav tako mora biti nameščen iPhone SDK (programska oprema za razvijanje, angl. *Software Development*

Kit), ki vključuje Xcode razvojno okolje, iPhone simulator in ostala orodja za razvijanje. Za prenos iPhone SDK mora biti uporabnik registriran kot Registered iPhone developer (t.j. registriran iPhone razvijalec). S tem pridobi dostop do iPhone razvijalnega centra¹, vključno z vso dokumentacijo.

S simulatorjem se lahko doseže veliko, vendar se za bolj realistično testiranje aplikacije namešča na resnično napravo - iPhone, iPod, iPad ali Apple TV. iPad poganja isti iOS kot iPhone in iPod in se zato uporabljajo enake tehnike razvijanja z minimalnimi razlikami. Za nameščanje aplikacij na naprave iOS in za distribucijo aplikacij je potrebno biti včlanjen v iPhone Developer Program (t.j. program razvijalcev iOS opreme), letna članarina znaša 99 dolarjev.[3, p. xvi]

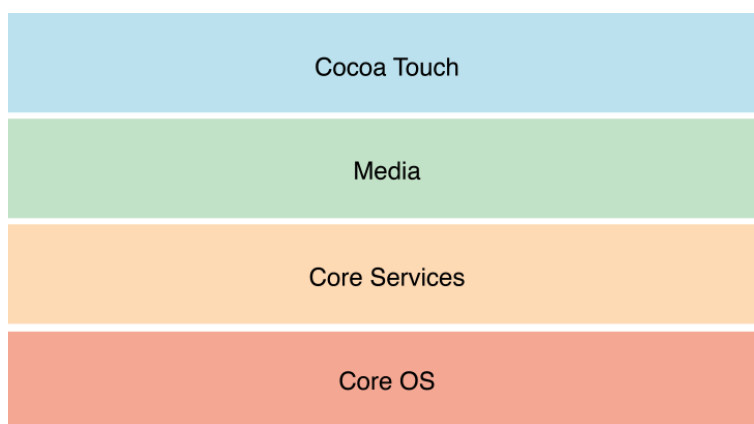
2.1.2 Arhitektura iOS

Računalniška arhitektura iOS je zgrajena iz štirih abstraktnih slojev, ki so prikazni na sliki 2.1. Na najvišjem sloju iOS služi kot vmesnik med strojno opremo in aplikacijo, ki je trenutno v uporabi. Aplikacije tako navadno komunicirajo s strojno opremo preko sistemskih vmesnikov in so s tem zaščitene pred spremembami v strojni opremi. Na najnižjih slojih sistema so temeljne storitve in tehnologije, na katerih slonijo vse aplikacije, višji sloji pa vsebujejo bolj prefinjene storitve in tehnologije. Ob pisanju kode Apple spodbuja razvijalce k rabi najvišjih nivojev arhitekture, kjer je to le mogoče.[4]

Core OS

Core OS sloj je najnižji sloj v arhitekturi iOS in vsebuje nizko-nivojske funkcije, na katerih je zgrajena večina preostalih tehnologij. Tehnologije vsebovane v tem sloju večinoma niso dostopane neposredno s strani razvijalca; izjema so izrecno ukvarjanje z varnostjo ali komunikacijo z zunanjo dodatno strojno opremo.

¹<http://developer.apple.com/>



Slika 2.1: Sloji arhitekture iOS.

Core Services

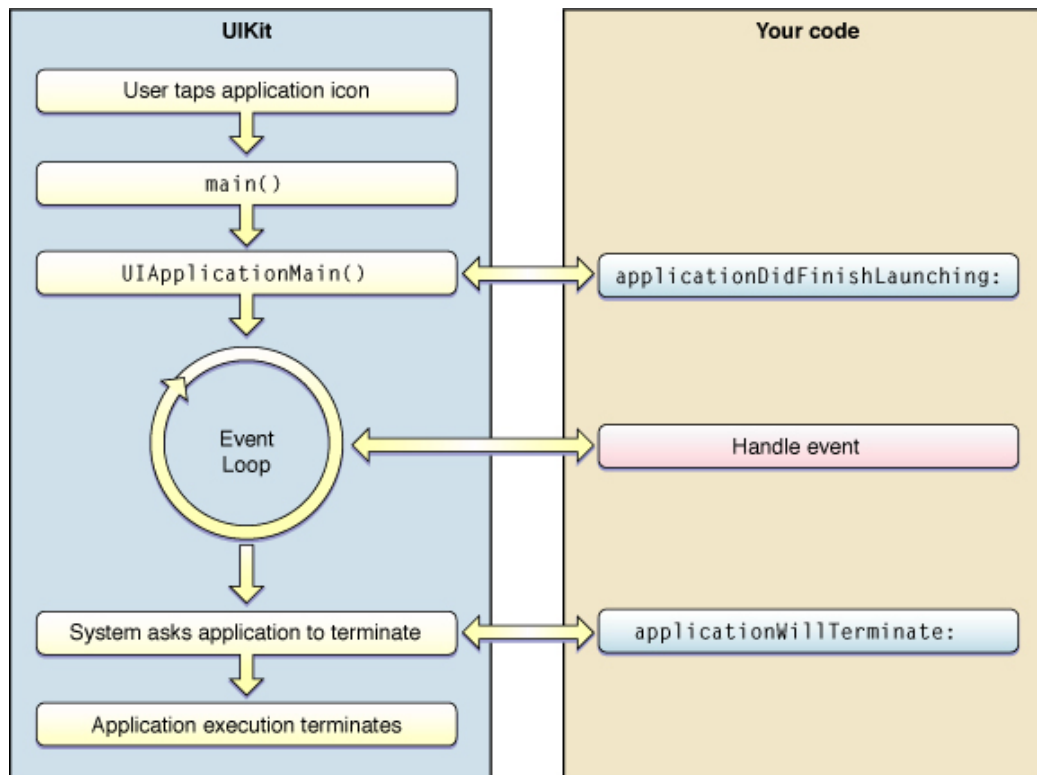
Core Services sloj vsebuje temeljne sistemske storitve, ki jih uporabljajo vse aplikacije. Vsebuje tudi že nekaj višjenivojskih funkcij, kot so iCloud (Appleova implementacija računalništva v oblaku), ARC (t.j. Automatic Reference Counting; funkcija prevajalnika, ki poenostavi proces upravljanja življenjskega cikla Objective-C objektov), In-App Purchases (kupovanje znotraj aplikacije), XML podpora, SQLite...

Media

Ta sloj vsebuje tehnologije, ki skrbijo za grafiko, zvok in video. Nudi funkcije kot so risanje z 2D vektorji, animiranje pogledov, predvajanje zvoka različnih formatov, upravljanje s sliko iz vgrajene kamere...

Cocoa Touch

Cocoa Touch sloj je najvišji sloj arhitekture iOS in vsebuje ključna ogrodja (angl. *frameworks*) za gradnjo iOS aplikacij. Definira osnovno infrastrukturo aplikacije in ključne tehnologije kot so večopravilnost, sprejemanje dotikov, razporejanje pogledov...



Slika 2.2: Nenehno aktivna zanka dogodkov.

2.1.3 Dogodkovno vodeno programiranje

Pri programiranju za iOS se uporablja Dogodkovno vodeno programiranje. Ob zagonu aplikacije ta vstopi v zanko, kot je prikazano na sliki 2.2.

Objekt `UIApplication` nenehno čaka na nek dogodek. Ko se uporabnik dotakne zaslona, je dogodek dotika posredovan s strani `UIApplication` do pogleda, kjer je bil dogodek sprožen. To je navadno nek kontrolni element, kot na primer `UIButton` gumb, ki nato posreduje svoje sporočilo o zahtevani akciji svojemu krmilniku. S tem se sproži razvijalčeva koda.

Pri programiranju za iOS je zanka dogodkov nenehno aktivna in je odgovorna za pravilno delovanje aplikacije.[3, p. 16]

2.2 Xcode

Za razvijanje iOS aplikacij mora imeti razvijalec na voljo določena orodja. Vsa potrebna orodja so dosegljiva s strani Apple Developer programa in so brezplačna za uporabo.

Xcode je integrirano razvojno okolje (angl. *IDE - Integrated Development Environment*), ki se uporablja za izdelavo aplikacij za iOS in OS X. Poleg tega, da je urejevalnik kode, vsebuje tudi podporo za samodokončevanje ukazov, analizo statične kode pred prevajanjem in iskanje puščanja spomina (angl. *memory leaks*). Ima tudi vgrajen razhroščevalnik.

Interface Builder

Za lažjo vizualno razvijanje pogledov v aplikaciji se uporablja Interface Builder, ki je v zadnjih iteracijah Xcodea kar vgrajen. Vizualni gradniki, kot so gumbi, oznake, tekstovna polja ipd., se lahko enostavno povlečejo na uporabniški vmesnik. Vse gradnike lahko tudi prilagodimo svojim potrebam z namenskimi drsniki in vnosnimi polji za razne vrednosti (višina in širina gradnikov, pozicija, ipd.). Na gradnike se lahko vežejo tudi akcije (kaj naj se zgodi ob kliku na gumb) in tarče (komu naj se pošljejo podatki o interakciji).

Instruments

Je prav tako kot Interface Builder vgrajen v Xcode in je orodje za analizo in testiranje učinkovitosti aplikacije v dinamičnem času. Omogoča sledenje enemu ali več procesov hkrati in pregledovanje zbranih podatkov. Uporablja se pretežno za odkrivanje puščanja spomina.

Cocoa Frameworks

Apple nudi veliko okvirjev (angl. *frameworks*), s katerimi si razvijalci lahko izredno olajšajo delo in so prav tako vgrajeni v Xcode. Okvirji nudijo dostop do izrisovanja grafik, predvajanja zvokov in pa vse do prikazovanja spletnih

strani. Veliko kompleksnih operacij se tako lahko izvede le s kakšno vrstico ali dvema kode.

2.3 Objective-C

Programski jezik, ki se uporablja za razvijanje iOS aplikacij, se imenuje Objective-C, ki je enostavna razširitev programskega jezika C. Je refleksivno objektno usmerjen programski jezik in je glavni programski jezik, ki ga uporablja Apple za operacijske sisteme OS X in iOS ter za vmesnik Cocoa za programiranje aplikacij (angl. *API - application programming interface*).

Razvit je bil leta 1980 za potrebe NeXTSTEP operacijskega sistema, iz katerega izhajata OS X, kot tudi iOS.[5, p. 29]

Za Objective-C so značilni oglati oklepaji, ki se uporabljajo za komunikacijo z objekti. V primeru 2.1 se tako izvaja klic `drain` nad objektom `pool`. Primer programa, ki v konzolo izpiše "Hello, World!":

Listing 2.1: "Hello World!"

```
1 #import <Foundation/Foundation.h>
2
3 int main (int argc, const char * argv[])
4 {
5     NSAutoreleasePool *pool =
6     [[NSAutoreleasePool alloc] init];
7     NSLog(@"Hello, World!");
8     [pool drain];
9     return 0;
10 }
```

Poglavje 3

Razvoj aplikacije

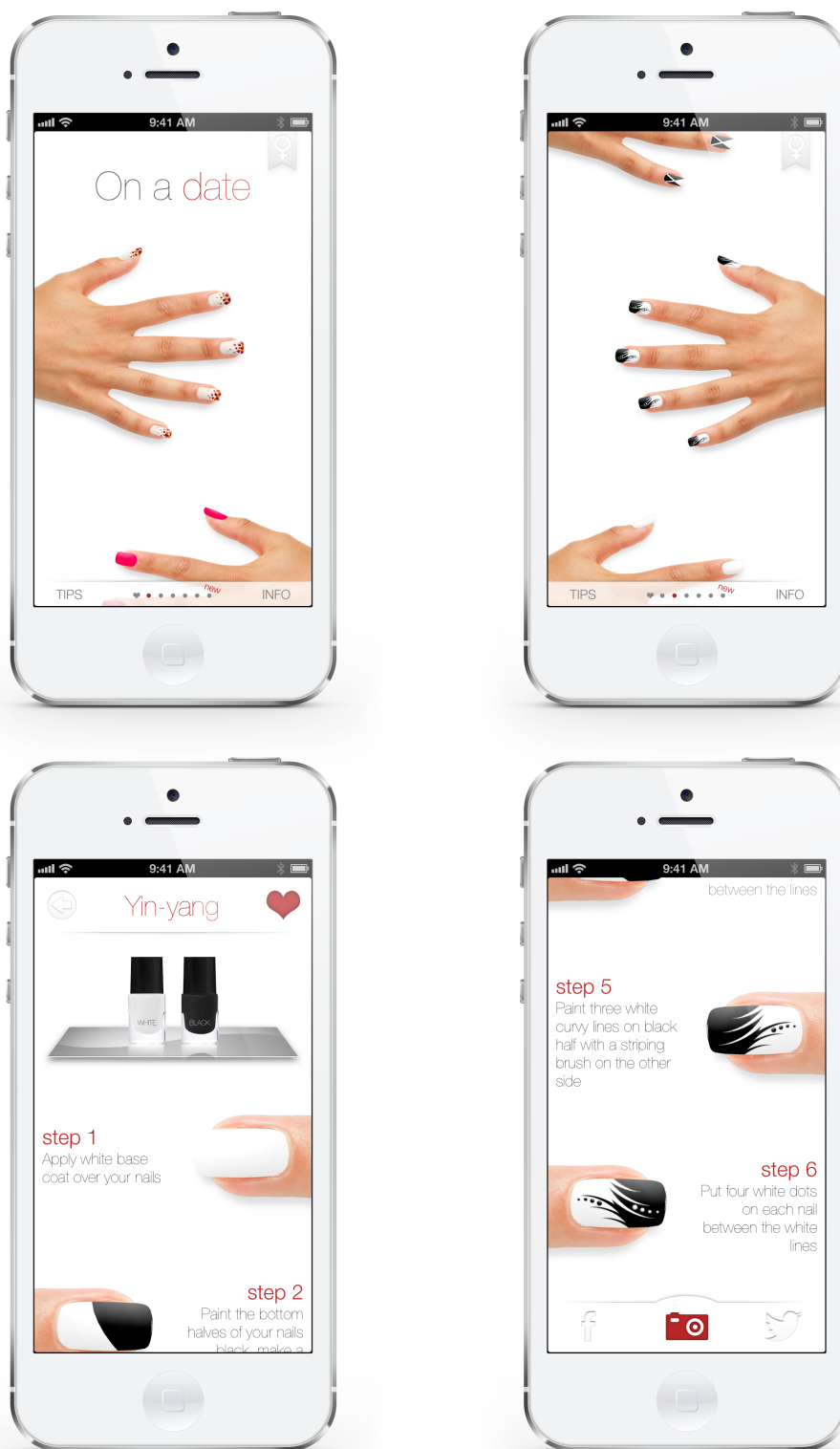
3.1 Nail Designs

Nail Designs je iOS aplikacija, ki uporabnika nauči narediti raznovrstne oblike manikur preko navodil po korakih z vmesnimi slikami in besedilom. Razdeljena je na več tematsko obarvanih paketov manikur, kjer vsak paket vsebuje po 20 različnih oblik manikur. Izgled aplikacije je prikazan na sliki 3.1.

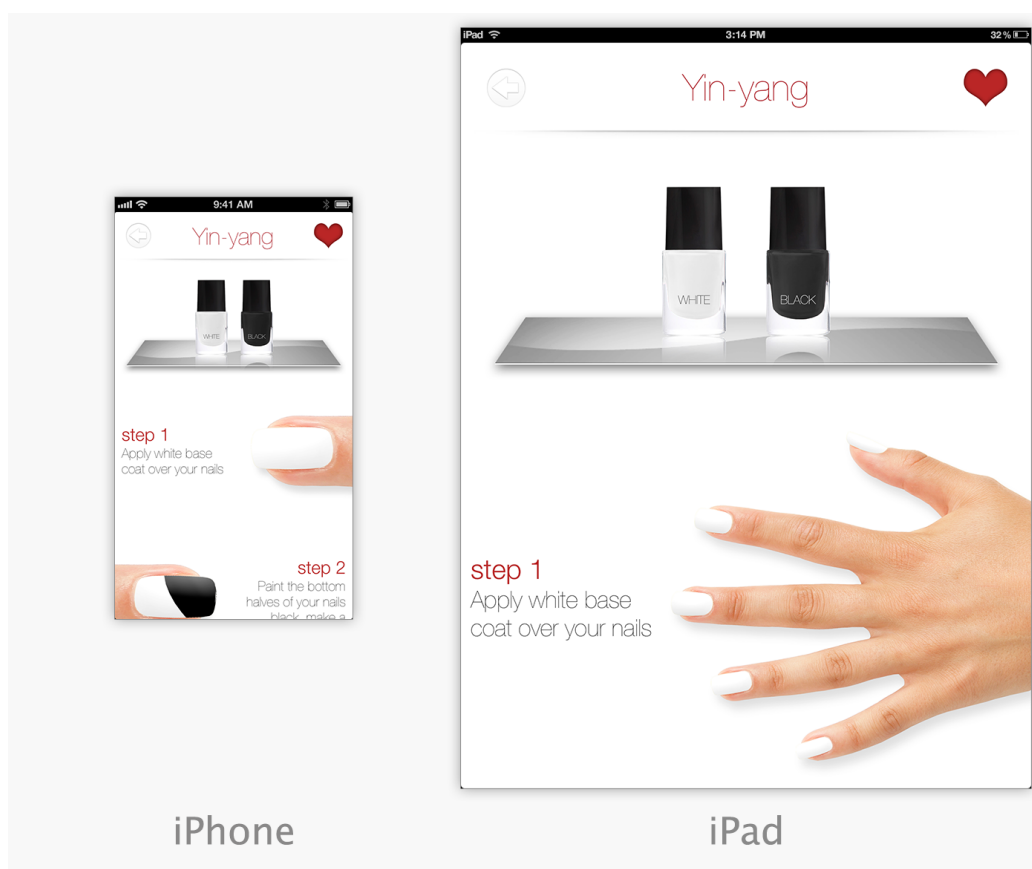
Aplikacija je na voljo za brezplačen prenos na App Storu, ponuja pa dodatno plačljivo vsebino v obliki zaklenjenih paketov. Uporabnik ima tako ob prenosu aplikacije na voljo prvi paket brezplačno, če pa mu je aplikacija všeč, pa lahko odkleni tudi preostale pakete. Možno je odkleniti posamezen paket za \$0,99 ali pa vse pakete hkrati za \$1,99.

3.2 Prilagajanje večjemu zaslonu iPada

Aplikacija je univerzalna in je na voljo tako za iPhone in iPod (za lažje razumevanje v nadaljevanju pisano le iPhone), kot tudi za iPad. S tem sem dosegel, da uporabniki, ki si lastijo tako iPhone in tudi iPad, ob prenosu aplikacije na eno napravo aplikacijo pravzaprav dobijo na obeh. Posledično, če znotraj aplikacije kupijo kakšen paket preko In-App Purchasov, se le-ta odklene tudi na drugi napravi, torej uporabniku ni potrebno odšteti denarja



Slika 3.1: Prikaz posnetkov aplikacije na belem iPhonu 5.



Slika 3.2: Primerjava slik korakov med iPhonom in iPadom.

večkrat za enako vsebino na več napravah.

Za prikazovanje na večjem zaslonu iPada je uporabniški vmesnik nekoliko dodelan, vendar ohranja enak izgled in s tem prepoznavnost aplikacije. Glavna razlika med prikazovanjem na različnih napravah so slike posameznih korakov manikur, kar je razvidno iz slike 3.2. Koraki so namreč na iPhone verziji aplikacije prikazani kot posamezen prst, na katerem je prikazan korak manikure, na iPad verziji¹ pa je za posamezen prikaz koraka uporabljena kar cela roka, ker je na voljo več prostora za prikaz slike.

Ker se potreba po razlikovanju kode za iPhone oziroma iPad pojavlja

¹Pravzaprav ni "iPhone verzije" in "iPad verzije", temveč je le ena aplikacija. S tem je mišljen prikaz aplikacije na iPhonu oziroma iPadu.

velikokrat, sem ustvaril C macro za hitro in enostavno deljenje kode na dva dela. Prav tako se v zadnjem letu zaradi izida naprave iPhone 5 z nekoliko večjim zaslonom pojavlja potreba tudi po razlikovanju med posameznimi napravami znotraj iPhone družine. Tudi to težavo sem odpravil z uvedbo macro izraza. Prikaz definicije macro ukazov in primer uporabe je prikazan v primeru 3.1.

Listing 3.1: Prikaz definicije in rabe C macro ukazov

```
1 #define isIdiomPhone()      ([[UIDevice currentDevice]
    userInterfaceIdiom] == UIUserInterfaceIdiomPhone)
2 #define isIphone5()        ([[UIScreen mainScreen].bounds.size.
    height == 568)
3
4 // Skupna programska koda
5 if (isIdiomPhone()) {
6     // Koda specifična iPhonom
7 } else {
8     // Koda specifična iPadom
9 }
```

3.3 Internacionalizacija in lokalizacija

Nail Designs je internacionalizirana aplikacija, kar pomeni, da je strukturirana tako, da je pripravljena na dinamično dodajanje, brisanje in spreminjanje različnih lokalizacij. Apple trenutno nudi možnosti lokaliziranja v več kot 40 jezikov. Nail Designs je trenutno lokalizirana v angleški, nemški in francoski jezik. Lokalizacija je zelo močno orodje za prodor na specifične trge različnih držav - ključno vlogo pri tem igrajo ključne besede, ki so besede oziroma besedne fraze, po katerih uporabniki iščejo aplikacije na App Storu.[6]

Za internacionaliziranje aplikacije je potrebno prirediti nib datoteke², da lahko sprejmejo prevedena besedila, statična besedila in grafike aplikacije. Te morajo biti prevedene, posebno pozornost pa je treba posvetiti prilagojenim velikostim gumbov, ker se lahko dolžine prevodov izjemno razlikujejo.

Za lokaliziranje aplikacije je potrebno v projektu ustvariti datoteko `Localizable.strings`, ki vsebuje vsa besedila, ki jih aplikacija potrebuje. Ta datoteka se nato lokalizira v željene jezike, posamezne lokalizacije pa se avtomatično premaknejo v datotečno strukturo prilagojeno lokalizaciji, kjer ima vsak željen jezik lokalizacije svojo mapo. Kratek izsek primerjave iste vrstice iz različno lokaliziranih `Localizable.strings` je prikazan v listini 3.2. Primer rabe te lokalizirane besede pa je prikazan v primeru programske kode 3.3. Naprava ob zagonu aplikacije sama določi jezik aplikacije glede na izbran jezik v nastavitvah naprave.

Listing 3.2: Primerjava iste vrstice iz različno lokaliziranih `Localizable.strings`

```
1 // English localization (en.lproj)
2 "FAVORITES" = "My_favorites";
3
4 // German localization (de.lproj)
5 "FAVORITES" = "Meine_Favoriten";
6
7 // French localization (fr.lproj)
8 "FAVORITES" = "Mes_favoris";
```

Listing 3.3: Raba lokalizirane besede za ime gumba

```
1 [favoritesButton setTitle:NSLocalizedString(@"FAVORITES", nil)
   forState:UIControlStateNormal];
```

²Za lažje razumevanje bralca, nib datoteke so datoteke z končnico `.xib`, ki so pravzaprav posebej strukturane xml datoteke, ki vsebujejo informacije o aplikacijskih oknih, pogledih in postavitvijo vseh podpogledov. Uporabljajo se za oblikovanje in strukturiranje uporabniških vmesnikov.

3.4 Integracija socialnih omrežij

Nail Designs omogoča objavlanje objav in slik na socialni omrežji Facebook in Twitter. iOS SDK (t.j. Software development kit) od verzije iOS 5 dalje vsebuje programsko ogrodje `Twitter.framework`, ki se uporablja za komunikacijo s Twitterjevimi strežniki. Od verzije iOS 6 dalje pa je prej omenjeno ogrodje označeno kot opuščeno (angl. *deprecated*), zamenjalo pa ga je novo programsko ogrodje `Social.framework`, ki omogoča komunikacijo s Twitterjem, Facebookom in kitajskim socialnim omrežjem Sina Weibo.

Ker Nail Designs podpira naprave z nameščenim operacijskim sistemom iOS 5 in novejši, je za starejše naprave za objavlanje na omrežje Facebook uporabljen Facebook SDK za iOS, ki je na voljo za prenos na spletni strani za Facebook razvijalce³. Pri napravah z novejšim operacijskim sistemom pa se uporablja novo ogrodje `Social`, ki vsebuje vse potrebne metode. Primer objave na omrežje Facebook preko ogrodja `Social` je prikazan v izseku programske kode 3.4. Programska koda je skoraj enaka za objavo na vseh ostalih socialnih omrežjih, ki so podprta s tem ogrodjem.

Listing 3.4: Objava slike s kratkim besedilom na omrežje Facebook

```
1 if ([SLComposeViewController class]) { //Social.framework?
2     SLComposeViewController *controller = [SLComposeViewController
        composeViewControllerForServiceType:SLServiceTypeFacebook];
        //Ustvari krmilnik
3
4     NSString *besedilo = ... //Pridobi besedilo
5     [controller setInitialText:besedilo]; //Nastavi besedilo
6
7     UIImage *slika = ... //Pridobi sliko
8     [controller addImage:slika]; //Nastavi sliko
9
10    [self presentViewController:controller animated:YES completion:
        nil]; //Prikazi krmilnik
11 }
```

³<https://developers.facebook.com/ios/>

3.5 Izdelava slik manikur

3.5.1 Lakiranje in slikanje

Prvotna različica aplikacije je vsebovala resnične fotografije manikur. Proces je bil zelo zamuden in utrujajoč; nohte se je nalakiralo za prvi korak, roko se je fotografiralo, nato pa se je ta proces ponavljal, dokler manikura ni bila končana. Vse narejene fotografije se je potem naknadno obdelalo v Adobe Photoshopu (v nadaljevanju le Photoshop) in obrezalo.

Slike vsebovane v aplikaciji so bile tako nekonsistentno obarvane, kakovost slik ni bila dovolj visoka, prav tako pa tudi kakovost samih manikur ni bila na profesionalnem nivoju. Zaradi slednjih razlogov sem se ob izdaji nove verzije aplikacije, verzije 2.0, odločil za menjavo celotnega procesa izdelave slik manikur.

3.5.2 Uporaba Adobe Photoshopa

Prvi korak je bilo profesionalno slikanje nenalakirane roke ženskega modela. Dobljena slika se je nato natančno obrezala okoli roke zaradi potrebe transparentnega ozadja v aplikaciji. Obrezana slika roke se je nato svetlobno in barvno uredila.

Temu je sledila izdelava različnih manikur s pomočjo Photoshopa. Na sliki 3.3 se lepo vidi razlika v kakovosti fotografirane manikure in manikure narejene s pomočjo Photoshopa. Po končani izdelavi vseh manikur se je naknadno izrezal še prst, ker se pri iPhonu uporablja le ta za prikaz posameznih korakov, kot je razloženo v poglavju 3.2 na strani 9. Primerjava slike istega koraka med iPadom in iPhonom je prikazana na sliki 3.4.

Po končanem delu v Photoshopu so bile slike izvožene v 24 bitni PNG format. Za vsak korak posamezne manikure so bile izvožene štiri slike:

- 1000 x 1000 px, celotna roka, iPad retina,
- 500 x 500 px, celotna roka, iPad ne-retina,



Fotografirano



Photoshop

Slika 3.3: Razlika v kakovosti fotografirane manikure in manikure narejene s pomočjo Photoshopa.



iPhone



iPad

Slika 3.4: Primerjava slike istega koraka med iPadom in iPhoneom.

- 500 x 500 px, posamezni prst, iPhone retina,
- 250 x 250 px, posamezni prst, iPhone ne-retina

Poleg že omenjenih slik posameznih korakov pa so bile za iPhone potrebne tudi slike končanih manikur na celotni roki za prikaz v izbirnih menijih (vidno na zgornjih dveh posnetkih aplikacije na sliki 3.1 na strani 10). Za vsako manikuro je bilo torej potrebno izvoziti še dodaten par slik, ena slika velikosti 500 x 500 px za iPhone z retina zaslonom in polovično velikost za iPhone brez retina zaslona⁴.

3.5.3 Manjšanje velikosti aplikacije

Aplikacija je imela tako v novi različici visoko kvalitetne slike, ki so bile med seboj konsistentne. Težava pa se je pojavila, ker je bilo v novi različici aplikacije na voljo petkrat več manikur kot v prejšnji različici. Ob upoštevanju, da je aplikacija univerzalna za vse naprave in vsebuje štiri različice slik za vsak posamezni korak manikure, je razumljivo, da je sama velikost aplikacije izredno narasla. To se je izkazalo kot velik problem, ker uporabniki niso imeli možnosti aplikacije namestiti preko mobilnih omrežij.

Težavo sem razrešil s sestavljanjem slojev večih prosojnih slik skupaj, kot je prikazano na sliki 3.5. Ker potrebujem roko oziroma prst na vsaki prikazani sliki manikure, je povsem nepotrebno imeti te podatke zapisane v vsaki sliki. Tako sem najprej izvozil sliko roke in prsta v potrebnih velikostih brez nalakiranih nohtov in s tem pridobil spodnjo plast prikazanih slik. Čeznjo nato prikažem skoraj povsem prosojno sliko, ki vsebuje le nalakirane nohte. Kot primer razlike v zasedanju spomina bom v primerjavo vzel iPad retina sliki ključnega koraka pred in po optimizaciji. Do sedaj je slika enega koraka zasedala 932 KB spomina, po opravljeni optimizaciji pa slika zaseda le 65 KB. To je kar 93 % znižanje velikosti slike.

⁴Retina zaslon je blagovna znamka, ki jo uporablja Apple za označevanje visoko ločljivih zaslonov vgrajenih v njihove naprave zadnjih generacij.



Slika 3.5: Sestavljanje večih slojev slik v skupno sliko.

Opazil sem tudi podvajanje določenih slik. Končne slike celotnih rok, ki jih prikazujejo iPhone retina naprave so namreč enake tistim, ki jih prikazujejo iPadi brez retina zaslona za zadnji korak manikur, torej velikosti 500 x 500 px. To podvajanje je nepotrebno, ker lahko s preverjanjem modela naprave med tekom aplikacije poiščemo ustrezno sliko za prikaz na zaslonu. Programska koda, ki skrbi za pravilno izvajanje te rešitve, je predstavljena v listini 3.5.

Listing 3.5: Prikaz iPad slike končnega koraka manikure na retina iPhonu.

```

1 NSString *imeSlike = ... //Ime slike
2 NSString *koncnica; //Koncnica
3 if (isIdiomPhone() && isRetina()) {
4     koncnica = [NSString stringWithFormat:@"%d~ipad", [[tutorial
5         objectForKey:@"Tutorial"] count]]; //Koncnica za iPad sliko
6         zadnjega koraka
7 } else {
8     koncnica = @"_Finish"; //Koncnica za ne retina sliko
9 }
10 UIImage *slika = [UIImage imageNamed:[imeSlike
11     stringByAppendingString:koncnica]]; //Pridobitev slike

```

S prej omenjenim optimizacijami sem dokončno zmanjšal velikost aplikacije za približno 80 %.

3.6 Prehod na brezplačni model s plačljivo dodatno vsebino

Aplikacija je bila sprva postavljena na trg s ceno \$0,99 in brez dodatnih paketov na voljo za nakup. Na dan je dosegala v povprečju med 20 in 40 nakupov. Z namenom nekaj dnevne izpostavljenosti sem aplikaciji za nek konec tedna znižal ceno na brezplačno, nakar se je v roku parih dni prenesla preko 50.000 krat.

Ob koncu promocije sem se s tem številom v mislih odločil spremeniti poslovni model aplikacije. Aplikacija naj bo brezplačna za namestitev in začetno uporabo z dodatno doplačljivo vsebino. V tujini se tak način poslovnega modela največkrat poimenuje *freemium* model.

3.6.1 Implementacija plačljive vsebine znotraj aplikacije

Za implementacijo plačljive vsebine znotraj aplikacije (angl. *In-App Purchases*) se uporablja ogrodje `StoreKit.framework`. Apple podpira štiri kategorije objektov, ki so lahko na voljo za prodajo znotraj aplikacije; to so vsebina, dodatna funkcionalnost, storitev in naročnine. V mojem primeru nudim uporabnikom dodatno vsebino v obliki odklenljivih paketov navodil za manikure. Izgled pogleda za odklenitev dodatne vsebine je prikazan na sliki 3.6.[7]

Vsebina, ki jo uporabnik odklene, ostane odklenjena in je ne more kupiti ponovno. Glede na to lastnost se opredeli kot nepotrošna plačljiva vsebina znotraj aplikacije (angl. *Non-Consumable In-App Purchase*). Vsebina, ki jo uporabnik v Nail Designs odklene, je pravzaprav že zapakirana znotraj paketa aplikacije (angl. *Bundle*), tako da ni potrebnega dodatnega prenosa podatkov.



Slika 3.6: Pogled za odklenitev dodatne plačljive vsebine znotraj aplikacije.

MKStoreKit

V Nail Designs je dodatna plačljiva vsebina implementirana s pomočjo zunanega ogrodja MKStoreKit⁵. To ogrodje izredno poenostavi implementacijo te vsebine ter usklajevanje kupljene vsebine preko več različnih naprav. Primer nakupa dodatne plačljive vsebine z MKStoreKitom je prikazan v izseku programske kode 3.6. Cel nakup se pravzaprav opravi le s klicem metode `buyFeature:onComplete:onCancelled:`, ki ji za argumente podamo identifikacijo objekta, za katerega naj se nakup opravi, ter dva bloka kode za povratni klic (angl. *Callback methods*): enega za primer uspešnega nakupa in drugega za primer, če se kupec za nakup ne odloči.

⁵<https://github.com/MugunthKumar/MKStoreKit/>

Listing 3.6: Primer nakupa dodatne plačljive vsebine z MKStoreKitom.

```
1 if ([[SKPaymentQueue canMakePayments]]) { // Preveri, ce ima
    naprava omogocene nakupe
2     [[MKStoreManager sharedManager] buyFeature:kFeature onComplete
       :^(NSString* purchasedFeature, NSData* data)
3     {
4         // Blok kode za povratni klic (angl. Callback) - nakup je
            tukaj ze opravljen
5         [[NSNotificationCenter defaultCenter] postNotificationName:@"
            Purchased" object:self]; // Obvesti kontroler o
            opravljenem nakupu, da prikaze odklenjeno vsebino
6     }
7     onCancelled:^
8     {
9         // Nakup je bil prekinjen
10    };
11 }
```

3.7 Raba analitičnih orodij in orodij za poročanje napak

Nail Designs uporablja tako analitična orodja, kot tudi orodja za poročanje napak.

Flurry Analytics

Flurry Analytics je brezplačno analitično orodje, ki služi za prejemanje informacij o uporabnikovi izkušnji z aplikacijo. Nudi zbiranje podatkov o novih uporabnikih, obstoju uporabnikov, toku rabe aplikacije ipd. Sledi tudi proženju dogodkov znotraj aplikacije, katerim želim slediti, tako da lahko sledim trendom novih in starih funkcionalnosti v aplikaciji ter se na podlagi le-teh odločam o nadaljnjem razvoju.[8]

Raba ogrodja Flurry je enostavna. Ob zagonu aplikacije se mora pokli-

cati metoda `startSession:`, za sledenje določenemu dogodku pa ga lahko registriramo s klicem metode `logEvent:`.

Zbrani podatki so nam na voljo na Flurry spletni strani za razvijalce, kjer se osvežijo v povprečju dvakrat na dan⁶.

Crashlytics

Crashlytics je brezplačno orodje, ki skrbi za poročanje napak. Razvijalcu omogoča vpogled v vse napake in hrošče, do katerih je prišlo v resničnem času in tako omogoča hiter odzivni čas in popravila. Implementacija je enostavna - po vezavi ogrodja Crashlytics z aplikacijo se ob zagonu aplikacije pokliče metoda `startWithAPIKey:`. Crashlytics nato samodejno prične z obveščanjem o morebitnih napakah, do katerih pride med življenjskim ciklom aplikacije.

⁶<https://dev.flurry.com/>

Poglavje 4

Izdaja in trženje aplikacije

4.1 Testiranje

Pred izdajo končne verzije aplikacije je potrebno aplikacijo testirati, da vidimo, kako se obnese v resničnem okolju.

Prvo stopnjo testiranja izvede razvijalec sam s pomočjo programske opreme (poglavje 2.2 na strani 7). Na voljo ima razhroščevalnik z grafičnim uporabniškim vmesnikom in program Instruments, s katerim lahko prečesava podatke v resničnem času. V tej stopnji testiranja se odkrivajo večinoma hrošči povezani z nepravilnim dodeljevanjem spomina znotraj aplikacije.

Ko aplikacija preстане interno testiranje, se navadno izvede tako imenovano Ad Hoc testiranje. Tekom Ad Hoc testiranja se aplikacijo namesti določenemu številu uporabnikov, ki so izven razvijalne skupine. Ti nato poročajo o morebitnih napakah, hroščih ali možnih izboljšavah.

4.1.1 Avtomatično testiranje

Uporablja se lahko tudi avtomatično testiranje, pri katerem se ustvari zbirke poskusov. Te so pravzaprav zbirke testnih primerkov aplikacije (angl. *unit tests*), ki so osredotočene na majhne skupke programske kode, navadno na določeno metodo ali razred.

Testni primerki dovoljujejo razvijanje aplikacije z manj hrošči - za vsak

najden hrošč se ustvari nov testni primerek in s tem si zagotovimo, da se nam hrošč v prihodnje ne pojavi več. Cilj je torej izvesti testni primerek vsakokrat, ko spremenimo kodo. Za praktičen primer avtomatizacije tega postopka lahko uporabimo tri storitve[9]:

- GitHub: Brezplačni javni Git repozitoriji
- Jenkins: Strežnik, ki spremlja spremembe na Git repozitoriju in avtomatično testira nove verzije aplikacije
- TestFlight: Storitev za samodejno pošiljanje iOS aplikacij preko brezžičnih omrežij do testnih oseb

4.2 Pregled aplikacije s strani Appla

Po končanem razvojnem ciklu in testiranju aplikacije se le-to stisne v zip datoteko in naloži na Applove strežnike. Preden pa aplikacija postane na voljo za prenos, mora prestati testiranje s strani Appla. Čakalna doba, da aplikacija preide v stanje pregleda, znaša v povprečju 7 dni, sam pregled pa je odvisen od kompleksnosti aplikacije. Aplikacija Nail Designs je bila v pregledu en dan, nakar je bila odobrena za prodajo. Z aplikacijo se upravlja preko spletne strani iTunes Connect¹, kjer ji lahko med drugim določamo prodajno ceno in datum izdaje. Na sliki 4.1 je prikazan izsek spletne strani za upravljanje z aplikacijo Nail Designs.

Po odobritvi se lahko aplikacija prične prodajati takoj, ali pa se izdaja zadrži za poljubno število dni. Navadno se razvijalci poslužujemo druge možnosti zaradi usklajenosti z zunanjim trženjem in lovljenjem najbolj dobičkonosnih dni, ki aplikacijo v prvih dneh postavijo na visoko mesto na lestvici.

Predvsem med izdajami iger je možno zaslediti trend izdajanja ob četrtek. Apple namreč vsak četrtek osveži svoj seznam izpostavljenih aplikacij (angl. *Featured Apps*) - seznam, na katerega se želi umestiti vsak razvijalec.

¹<https://itunesconnect.apple.com/>

The screenshot displays the iTunes Connect interface for the 'Nail Designs' app. The page is titled 'Nail Designs' and features several sections:

- App Information** (with an 'Edit' button):
 - Identifiers**: SKU (com.1234567890.NailDesigns), Bundle ID (com.1234567890.NailDesigns), Apple ID (492114299), Type (iOS App), and Default Language (English).
 - Links**: View in App Store, Customer Reviews.
 - Management Buttons**: Rights and Pricing, Transfer App, Manage In-App Purchases, Manage Game Center, and Manage IAd Network.
- Versions**:
 - Current Version**:
 - Version: 2.2
 - Status: ● Ready for Sale
 - Date Created: Apr 8, 2013
 - Date Released: Jun 22, 2013
 - Buttons: View Details, Add Version.

At the bottom of the interface, there is a 'Done' button and a footer with links for Home, FAQ, Contact Us, Sign Out, Terms of Service, and Privacy Policy, along with the copyright notice: Copyright © 2013 Apple Inc. All rights reserved.

Slika 4.1: Izsek spletne strani iTunes Connect za upravljanje z aplikacijo Nail Designs.

Posebno pozornost je potrebno nameniti načrtovanju roka izdaje v primeru, da aplikacija ne prestane pregleda. V primeru zavrnitve aplikacije se lahko sicer na zavrnitev pritožimo, nakar nastopi ponovna čakalna doba v povprečju 7 dni, nato pa se aplikacija znova pregleda. V primeru ponovne zavrnitve in ponovne pritožbe ne pride do ponovnega pregleda, temveč do telefonskega pogovora z zaposlenim iz Appa, ki je zadolžen za testiranje aplikacij. Tako se lahko sama izdaja aplikacije zamakne za en mesec, v katerem ne moremo spreminjati programske kode.

4.3 Izdelava spletne strani

Apple nudi dodajanje dveh povezav do zunanjih spletnih strani za vsako aplikacijo. Te povezave so predstavljene uporabnikom znotraj App Stora. Prva povezava je namenjena za podporo aplikaciji (angl. *Support URL*), druga pa za trženje aplikacije (angl. *Marketing URL*).

Za podporo se navadno uporabniku nudi spletna forma, kjer lahko uporabnik pride v stik z razvijalcem. Za sam uspeh aplikacije pa je nekoliko bolj zanimiva druga opcija, kjer lahko uporabnika preusmerimo na našo lastno spletno stran.

Spletno stran lahko uporabimo za bolj podrobno predstavitev naše aplikacije in tako obiskovalca prepričamo v nakup. Navadno so na spletnih straneh objavljeni posnetki iz aplikacije in opis aplikacije ter njenih funkcionalnosti. Nekateri se poslužujejo tudi video posnetkov aplikacije v uporabi, kar pa se v praksi ne izkaže kot zelo zanimiva opcija - obiskovalci navadno nimajo časa, da bi si ogledali posnetek; po večini imajo raje posnetke iz aplikacije.

Za preusmerjanje obiskovalcev spletne strani na App Store za nakup aplikacije ima razvijalec na voljo dva gumba oz. znački. V primeru, da je to gumb, ki vodi do aplikacije na App Storu, se uporablja značka za prenos aplikacije (angl. *Download on the App Store*), če pa želimo uporabniku le sporočiti, da je aplikacija na voljo za prenos na App Storu, pa se uporablja značka, ki sporoča prav to (angl. *Available on the App Store*). Obe znački



Slika 4.2: Znački za trženje aplikacije.



Slika 4.3: Izsek Facebook spletne strani za aplikacijo Nail Designs.

sta prikazani na sliki 4.2.[10]

4.4 Integracija socialnih omrežij

Za oglaševanje in širjenje besede o aplikaciji se uporabljajo tudi socialna omrežja, najbolj pogosto sta to Facebook in Twitter. Nail Designs ima ustvarjeno Facebook stran, preko katere se uporabnike, ki so stran označili kot *všečno* (angl. *liked*), obvešča o prihajajočih novostih. Izsek Facebook spletne strani za aplikacijo Nail Designs je prikazan na sliki 4.3.

Povezava za označitev Facebook strani kot *všečne* je vsebovana znotraj aplikacije. Uporabniku se ob *všečkanju* tudi odklene dodaten paket manikur -

s tem se doseže ogromno število *všečkanj* strani in s tem večja izpostavljenost aplikacije.

S pomočjo Facebook strani se izvaja tudi oglaševanje za nove prihajajoče aplikacije. Tako le-te pridobijo že ob izidu veliko število uporabikov in se zato umestijo na višja mesta na lestvicah.

4.5 Začasno zmanjšanje cene aplikacije

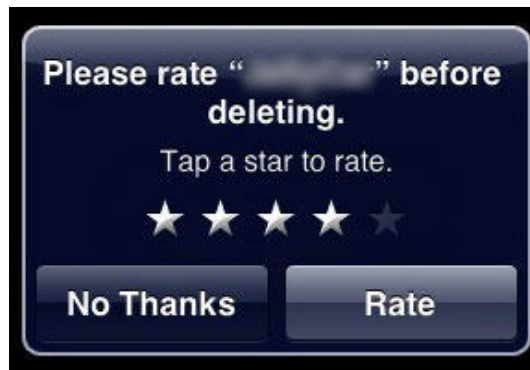
Trik, ki se ga poslužuje večina razvijalcev, je začasno zmanjšanje cene aplikacije. Aplikaciji se za nek kratek določen čas zmanjša cena, lahko tudi na brezplačno, kar močno vpliva na število prenosov aplikacije. V primeru Nail Designs, ki je bila sprva plačljiva aplikacija in imela v tem času v povprečju 35 prenosov na dan, je bilo ob prehodu na brezplačen prenos v prvem tednu 50,000 prenosov aplikacije.

S tem se pridobi ogromen skok na lestvicah in tako večjo izpostavljenost tudi po tem, ko se aplikaciji cena povrne na prvotno velikost (v primeru Nail Designs je sicer aplikacija ostala brezplačna zaradi spremenjenega načina pridobivanja zaslužka).

Ta trik je mogoč zaradi številnih spletnih strani, ki sledijo padcem v cenah vseh aplikacij. Te strani tako oglašujejo vse aplikacije, katerim se je znižala cena in s tem vodijo kupce do teh aplikacij.

4.6 Pridobivanje ocen zadovoljnih uporabnikov

Apple je imel v začetku izredno slab sistem za pridobivanje ocen uporabnikov. Uporabnika tekom rabe aplikacije nikoli ni zmotil, ob brisanu aplikacije pa se je uporabniku prikazal poziv, da naj aplikacijo oceni. Poziv je prikazan na sliki 4.4. Ta način pozivanja uporabnikov je pridobil le nezadovoljne uporabnike, ki so aplikacijo želeli izbrisati, ne pa tudi zadovoljne uporabnike, ki so aplikacijo uporabljali. Tako je aplikacija hitro lahko prišla na slab glas.



Slika 4.4: Poziv uporabnika k oceni aplikacije ob brisanju.

Appirater

Za implementacijo pridobivanja ocen zadovoljnih uporabnikov je v Nail Designs uporabljen brezplačen odprtokoden razred Appirater, dosegljiv preko spletne strani <https://github.com/arashpayan/appirater/>. Uporablja algoritem, ki prikaže poziv le uporabnikom, ki so aplikacijo zagnali n krat in jo imajo nameščeno že vsaj k dni. Ob izpolnitvi obeh pogojev se uporabniku prikaže poziv, ki ga prosi k oceni aplikacije. S tem pridobimo le ocene uporabnikov, ki so očitno zadovoljni z našo aplikacijo in so ocene temu primerne. V Nail Designs sta določena $n = 10$ in $k = 10$.

4.7 Oglaševanje preko ostalih aplikacij

Dobra praksa pri izdelavi iOS aplikacij je oglaševanje svojih preostalih aplikacij v aplikaciji. Je brezplačna oblika oglaševanja, kjer se že obstoječe uporabnike trenutne aplikacije prenese še na preostale aplikacije.

V Nail Designs je implementiran gumb, ki uporabnika vabi k ogledu svojih preostalih aplikacij. Ob kliku ga aplikacija preusmeri na App Store, kjer se mu prikažejo vse moje izdane aplikacije.

Poglavje 5

Sklepne ugotovitve

Aplikacija, predstavljena v tem diplomskem delu, se trenutno nahaja v samem vrhu svoje kategorije, kar odraža vso vloženo delo. Trenutno ima preko pol milijona uporabnikov, število pa se iz dneva v dan veča.

Uspeh Nail Designs pa ni prišel čez noč, bilo je potrebno veliko število iteracij, skozi katere se je aplikacija izboljševala. Veliko je k samim izboljšavam pripomoglo poslušanje odziva samih uporabnikov. Le oni namreč vedo, kaj so šibke točke aplikacije in kakšne lastnosti bi si še želeli.

Prav tako pa za sam uspeh iOS aplikacije ni dovolj le dobro napisana programska koda, temveč tudi kaj se dogaja v ozadju. Že ob samem pričetku projekta je potrebno začeti razmišljati o trženju aplikacije in pridobivanju morebitnih uporabnikov. Poleg načinov, omenjenih v tem diplomskem delu, obstaja še mnogo več trikov ter sistemov trženja in oglaševanja, ki pripomorejo k izboljšanju dobičkonosnosti aplikacije.

Prav to pa so tudi moje smernice za nadaljno delo - raziskati čim večje število novih načinov izboljšanja dobičkonosnosti aplikacije. Nekateri se namreč obnesejo, nekateri pa ne. Vsekakor pa mislim še naprej poslušati odzive uporabnikov in delati Nail Designs boljši iz iteracije v iteracijo.

Literatura

- [1] Apple, “Keynote video,” The Apple Worldwide Developers Conference, June 2013. [Online]. Available: <http://www.apple.com/apple-events/june-2013/>
- [2] Forbes, “The World’s Biggest Companies,” June 2013. [Online]. Available: <http://www.forbes.com/global2000/>
- [3] H. Conway and A. Hillegass, *iPhone Programming: The Big Nerd Ranch Guide*, 2nd ed. Big Nerd Ranch, Inc., August 2010.
- [4] iOS Developer Library, “iOS Technology Overview,” June 2013. [Online]. Available: <http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>
- [5] C. Hockenberry, *iPhone App Development*, 1st ed. O’Reilly Media, Inc., Maj 2010.
- [6] iOS Developer Library, “Internationalization and Localization,” June 2013. [Online]. Available: <https://developer.apple.com/internationalization/>
- [7] —, “In-App Purchase Guidelines,” June 2013. [Online]. Available: <https://developer.apple.com/in-app-purchase/In-App-Purchase-Guidelines.pdf>

- [8] Flurry, “Flurry Analytics Overview,” June 2013. [Online]. Available: <http://www.flurry.com/flurry-analytics.html>
- [9] A. Burkepile, C. Fulton, M. Galloway, J. Gundersen, K. Hafizji, M. Hollemans, F. L. Marsetti, M. Todorov, B. Trebitowski, and R. Wenderlich, *iOS 6 By Tutorials*. CreateSpace Independent Publishing Platform, 2012.
- [10] iOS Developer Library, “App Store Marketing and Advertising Guidelines for Developers,” June 2013. [Online]. Available: <https://developer.apple.com/appstore/AppStoreMarketingGuidelines.pdf>