



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE PRO PŘÍJEM Z AUDELIVER.COM

APPLICATION FOR AUDELIVER.COM DOWNLOADING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN PRISTAŠ

VEDOUcí PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2016

Abstrakt

Cielom tejto práce je vytvorenie aplikácie pre sťahovanie a prehrávanie podcastov. Zameral som sa hlavne na jednoduchosť návrhu dizajnu aplikácie a vytvorenie príjemného prostredia pre jej používanie. Aplikácia je naprogramovaná v jazyku Swift a využíva moderné prvky grafického rozhrania aplikácií v systéme iOS. Aplikáciu som mohol testovať na používateľoch a optimalizovať jej hardwarové nároky. Podarilo sa mi úspešne optimalizovať hardwarové nároky aplikácie a rozšíriť jej využitie na viacej zariadení Apple. Prínosom tejto práce sú nové poznatky vývoja aplikácií, ale aj dodržiavanie pokynov pre ľudské rozhrania. Keďže aplikácia využíva moderný jazyk a prináša moderné spôsoby realizácie, je prínosom hlavne pri ukážke toho, akým smerom sa vývoj aplikácií uberá.

Abstract

The aim of this work is to create an application to download and play podcasts. I focused mainly on the simplicity of the application design and creating a pleasant user environment. The application is programmed in Swift and uses modern elements of graphic application interface in iOS. I was able to test the application on users and optimize its hardware requirements. I managed to successfully optimize hardware requirements of the application and expand its use to more Apple devices. The benefit of this work is new knowledge of application development, but also respect for human interface guidelines. Since the app uses modern language and brings modern means of implementation, its main benefit is that it shows the direction the application development is headed.

Klíčové slová

Singlecast, Audeliver, Podcast, Sketch, Xcode, iOS, Swift, Alamofire, SWXMLHash, KD-CircularProgress, TLYShyNavBar

Keywords

Singlecast, Audeliver, Podcast, Sketch, Xcode, iOS, Swift, Alamofire, SWXMLHash, KD-CircularProgress, TLYShyNavBar

Citácia

PRISTAŠ, Martin. *Aplikace pro příjem z Audeliver.com*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Herout Adam.

Aplikace pro příjem z Audeliver.com

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána prof. Ing. Adama Herouta, Ph.D. Uviedol som všetky literárne zdroje a publikácie, z ktorých som čerpal.

.....
Martin Pristaš
16. mája 2016

Podakovanie

Chcel by som veľmi pekne poďakovať pánovi prof. Ing. Adamovi Heroutovi, Ph.D., ktorý mi odporučil pravidelné konzultácie a konzultoval so mnou celý vývoj aplikácie od jej návrhu po riešenie problémov pri odosielaní aplikácie do AppStore. Taktiež chcem poďakovať pánovi Ing. Igorovi Szókemu, Ph. D., ktorý mi umožnil použiť meno aplikácie Audeliver a krátko zhodnotil moje riešenie aplikácie Audeliver na osobných konzultáciách.

© Martin Pristaš, 2016.

Táto práca vznikla ako školské dielo na FIT VUT v Brně. Práca je chránená autorským zákonom a jej využitie bez poskytnutia oprávnenia autorom je nezákonné, s výnimkou zákonne definovaných prípadov.

Obsah

1 Úvod	2
2 Podcasty a služba Audeliver.com	3
2.1 Služba Audeliver	3
2.2 Špecifikácia Podcastov	3
2.3 Aplikácie na príjem podcastov dostupné v AppStore	6
3 Tvorba aplikácií v prostredí Apple	10
3.1 Programovanie pre systém iOS	10
3.2 Programy na návrh GUI	13
4 Návrh aplikácie Singlecast	15
4.1 Vývojový diagram aplikácie a jej využitie	15
4.2 Návrh UI	17
4.3 Tvorba databázového modelu v Xcode	21
4.4 Vhodné knižnice pre aplikáciu	21
5 Tvorba aplikácie Singlecast	24
5.1 Sťahovanie a parsovanie XML feedu a ukladanie dát v aplikácii	25
5.2 Zobrazenie zoznamu dostupných Podcastov v aplikácii	26
5.3 Informácie o Podcaste	27
5.4 Zoznam všetkých nahrávok a prehrávač v aplikácii – hlavné okno aplikácie	28
6 Testovanie aplikácie a uverejnenie na AppStore	31
6.1 Hardwarové nároky aplikácie	31
6.2 Testovanie na používateľoch	33
6.3 Zverejnenie aplikácie na AppStore	33
7 Záver	35
Literatúra	36
Prílohy	38
Zoznam príloh	39

Kapitola 1

Úvod

Cieľom mojej bakalárskej práce bolo vytvoriť aplikáciu pre systém iOS, ktorý je dostupný na mobilných zariadeniach a tabletoch značky Apple. Apple odporúča používať v aplikáciách svoje pokyny pre ľudské rozhranie. Popisujú, ako má aplikácia vyzeráť a ako sa majú pokyny pre posudzovanie aplikácie, ktoré boli odoslané na schválenie do AppStore dodržiavať.

V bakalárskej práci sa venujem systému iOS a programu Xcode, ktorý slúži na vývoj aplikácií pre operačné systémy iOS, OS X, watchOS, tvOS. Popíšem aplikáciu Sketch, ktorá mi pomohla pri grafickom návrhu aplikácie. Následne popisujem postup návrhu aplikácie Singlecast, ktorý sprevádza vývojový diagram a špecifikácia Podcastov, ktorými som sa musel riadiť, alebo aj knižnice, ktoré som pri návrhu a realizácii aplikácie využil. Preberiem tvorbu aplikácie Singlecast postupne pre každé okno aplikácie tak, ako som ju aj navrhol v grafických návrhoch v aplikácii Sketch a na záver zhodnotím testovanie aplikácie a neúspešný pokus odovzdania aplikácie do AppStore.

Súčasťou zadania bakalárskej práce je aj porovnať dostupné aplikácie na príjem Podcastov, ktoré som vyhodnotil na začiatku a aj vytvorenie prezentačného videa a plagátu, ktoré sú dostupné v prílohách.

Kapitola 2

Podcasty a služba Audeliver.com

Podcasty sú zvukové záznamy, ktoré autor umiestňuje na internet v podobe súborov, najčastejšie vo formáte MP3, na ktoré následne odkazuje na webových stránkach vo forme podcastu. Najčastejšími záznammi sú rádio nahrávky, nahrávky audio kníh, alebo zvukové záznamy rôznych divadelných hier či predstavení. Pre vytvorenie podcastu je potrebné vygenerovať XML kód, ktorý jednoznačne definuje podcast so všetkými jeho epizódami.

2.1 Služba Audeliver

Služba Audeliver¹ je webová služba, ktorá umožňuje vytvorenie vlastného Podcastu. Vytvorenie vlastného podcastu spočíva v tom, že si používateľ vytvorí vlastné konto Audeliver, do ktorého bude vytvárať podcasty. Po registrácii má používateľ možnosť vytvoriť si nový podcast, ktorý môže obsahovať všetky spomenuté položky v sekcii 2.2. Po vytvorení podcastu si používateľ môže nahráť vlastné audio súbory, alebo vložiť súbor pomocou odkazu na YouTube. Po nahratí súborov ich môže používateľ hromadne alebo osobitne pridať do svojho podcastu / podcastov. Takto vytvorený podcast obsahuje odkaz na jeho XML kód, ktorý sa pridáva v aplikáciách na prehrávanie a sťahovanie podcastov.

2.2 Špecifikácia Podcastov

Špecifikácia Podcastov je daná jednoznačne a určuje tak pravidlá, podľa ktorých sa má XML kód podcastu vytvárať. Presná špecifikácia XML kódu je dôležitá najmä pre to, aby všetky programy, ktoré podporujú prehrávanie Podcastov a ich odoberanie nemali problém XML kód Podcastu spracovať a čítať z neho jednotlivé položky. V základnom tvare obsahuje XML kód tagy obalené v hlavnom tagu `channel`, a to sú niektoré z nasledujúcich:

- `title` – Názov Podcastu
- `description` – Detailný popis Podcastu
- `link` – URL adresu odkazujúcu na stránku Podcastu
- `language` – Jazyk v ktorom je Podcast
- `copyright` – CopyRight podpisujúci tvorcu Podcastu

¹<http://www.audeliver.com/>

- `lastBuildDate` a `pubDate` – Slúžia na identifikáciu zmeny Podcastu a dátum jeho vytvorenia
- `item` – Epizóda Podcastu je obalená v tomto tagu

Okrem základných tagov, ktoré môže obsahovať XML Podcast, sa môžu vyskytnúť aj rozširujúce tagy pre iTunes, kde sa zadávajú špecifickejšie údaje Podcastu. Tieto rozširujúce tagy sú: *itunes:author*, *itunes:subtitle*, *itunes:summary*, *itunes:owner*, *itunes:explicit*, *itunes:image*, *itunes:category*.

Jednotlivé položky podcastu sú obalené v tagu `item` a ich parametre sú:

- `title` – Názov jednotlivej položky umiestnenej v Podcaste
- `description` – Popis položky
- `link` – URL adresa odkazujúca na viac informácií o položke
- `enclosure` – Obsahuje tieto atribúty:
 - `url` – URL adresa odkazujúca na súbor Podcastu
 - `length` – Veľkosť položky v bajtoch
 - `type` – Druh súboru – môže byť audio/mpeg, video/mp4 a podobne, viď [9]
- `pubDate` – Slúži na identifikáciu dátumu vytvorenia položky

XML kód Podcastu nieje limitovaný množstvom položiek a ani pridávaním nových označení. Vyššie som vypísal iba základné označenia, ktoré budem potrebovať pri práci s XML kódom v aplikácii.

2.2.1 Vzor XML štruktúry podcastov

Vzor XML kódu je vygenerovaný službou Audeliver, ktorá umožňuje vytvorenie vlastného Podcastu z vlastnej audio tvorby.

```
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd" xmlns:atom="
  http://www.w3.org/2005/Atom" xml:lang="de" version="2.0">
<channel>
<title>Podcast FIT</title>
<link>https://audeliver.com/</link>
<atom:link href="https://audeliver.com/podcast/ba70NTaj2jU0/feed.xml" rel="
  self" type="application/rss+xml"/>
<description>temporary podcast for Bachelor thesis</description>
<generator>Audeliver - https://audeliver.com</generator>
<lastBuildDate>Sun, 10 Apr 2016 13:05:02 +0200</lastBuildDate>
<language>en</language>
<copyright/>
<itunes:image href="https://audeliver.com/imagecloud/
  podcastdf86c3c5ceb8d2860f84d834bf1f61b83e6da3f0.jpg"/>
<image>
<url>
https://audeliver.com/imagecloud/
  podcastdf86c3c5ceb8d2860f84d834bf1f61b83e6da3f0.jpg
</url>
<title>Podcast FIT</title>
<link>https://audeliver.com/</link>
</image>
<itunes:summary>temporary podcast for Bachelor thesis</itunes:summary>
<itunes:subtitle>My first podcast</itunes:subtitle>
<itunes:author>Martin</itunes:author>
<itunes:owner>
<itunes:name>Martin</itunes:name>
<itunes:email>pristasm@gmail.com</itunes:email>
</itunes:owner>
<itunes:category text="Business">
<itunes:category text="Business News"/>
</itunes:category>
<itunes:category text="Education">
<itunes:category text="Educational Technology"/>
</itunes:category>
<itunes:explicit>yes</itunes:explicit>
<item>...</item>
<item>...</item>
<item>...</item>
</channel>
</rss>
```


Jednotlivé položky podcastu majú takúto štruktúru:

```
<item>
<title>01 - my own audio.mp3</title>
<itunes:subtitle/>
<description/>
<link>https://audeliver.com/mp3cloud/FOR8mBgs3FVL3W.mp3</link>
<enclosure url="https://audeliver.com/mp3cloud/FOR8mBgs3FVL3W.mp3" length="
  90312" type="audio/mpeg"/>
<guid>https://audeliver.com/mp3cloud/FOR8mBgs3FVL3W.mp3</guid>
<itunes:duration>0:00:05</itunes:duration>
<itunes:keywords/>
<itunes:explicit>no</itunes:explicit>
<pubDate>Tue, 29 Mar 2016 21:25:40 +0200</pubDate>
</item>
```

2.3 Aplikácie na príjem podcastov dostupné v AppStore

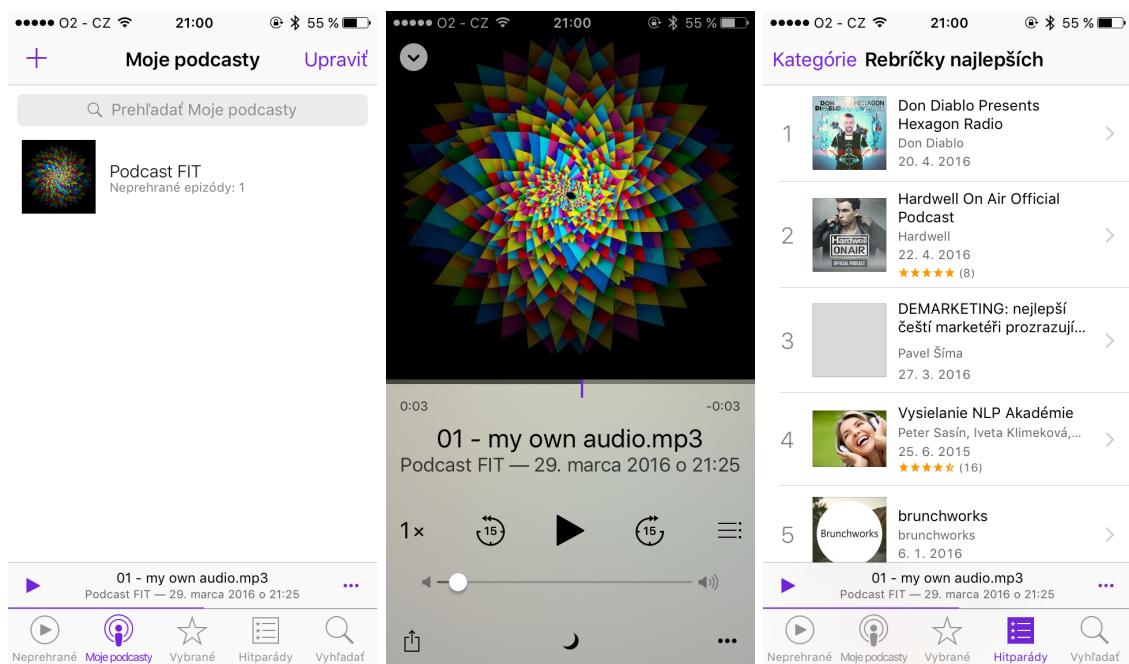
Hudobných aplikácií a aplikácií na sťahovanie podcastov nie je zrovna málo, no veľa vývojárov si za ne nechávajú zaplatiť. Cena pri aplikáciách na príjem Podcastov sa pohybuje medzi úrovňami **0.00 € - 9.99 €**, pričom tie, ktoré sú zdarma obsahujú množstvo platených funkcií pomocou in-App Purchases (nákupov v Aplikácii). Výber troch aplikácií, ktoré sú zdarma je popísaný nižšie.



Obr. 1: Ikony vybraných aplikácií pre prehrávanie Podcastov. Zľava: Podcasts, Overcast: Podcast Player, RSSRadio Podcast Downloader App

Vstavaná aplikácia – Podcasty

Táto vstavaná iPhone/iPad aplikácia od spoločnosti Apple je doslova nabitá funkciami a občas budí dojem, že je prehnane zložitá. Aplikácia úzko spolupracuje s iTunes a ich dostupnými Podcastmi, takže prakticky má prístup ku všetkým Podcastom, ktoré sa v iTunes objavujú. Podporuje synchronizáciu Podcastov spolu s aplikáciou iTunes – dostupnou ako na OS X, tak aj na Windows. Pri synchronizácii telefónu s iTunes sa vždy synchronizujú aj položky Podcastov.



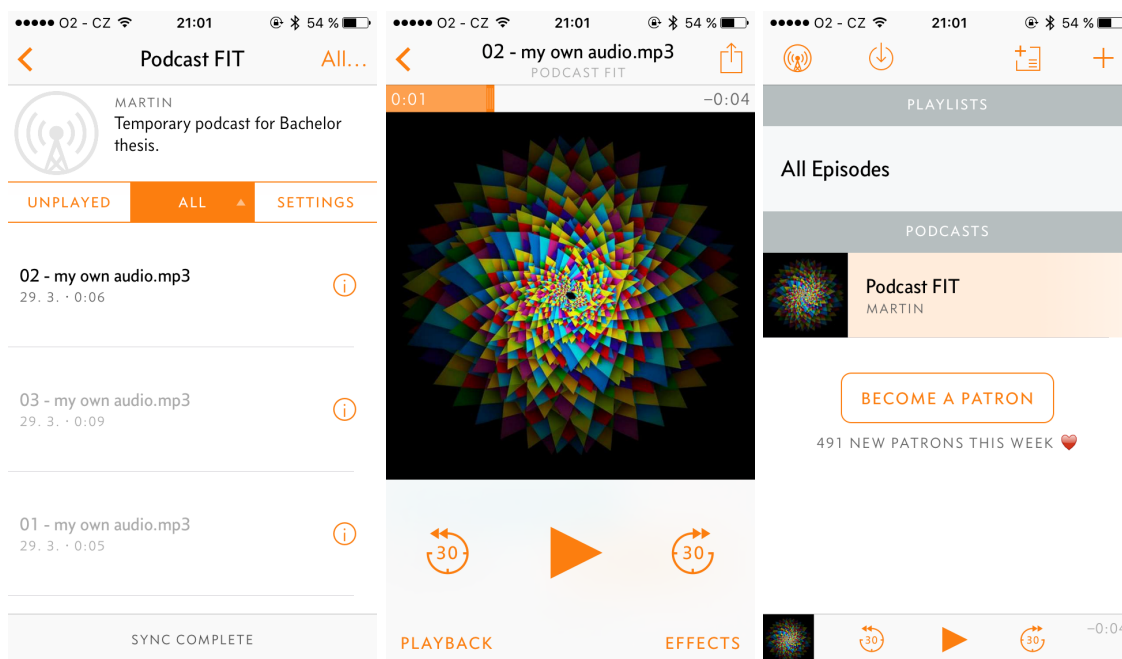
Obr. 2: Screenshots aplikácie Podcasty

Výhody a nevýhody aplikácie Podcasty

- | | |
|----------------------------------|--------------------------------|
| + Rýchla synchronizácia | – Príliš neprehľadné |
| + Podpora viacerých Podcastov | – Nemá iCloud zálohy |
| + Podpora podcastov z URL adresy | – Nemá podporu pre Apple Watch |

Aplikácia Overcast

Aplikácia Overcast je jedna z niekoľkých aplikácií, ktoré sú zdarma a slúžia na sťahovanie a prehrávanie Podcastov. Táto aplikácia je veľmi šikovná a ľahko ovládateľná. Obsahuje veľké množstvo funkcií, ktoré sú však platené ako in-App nákupy, čo nemusí každému vyhovovať, no môže si aspoň kúpiť to, čo potrebuje. V platenej verzii podporuje priame nahrávanie (upload) epizód alebo aj zmenu témy. Aplikácia obsahuje úplne iný druh písma ako všetky ostatné Apple aplikácie, čo nemusí každému používateľovi vyhovovať.



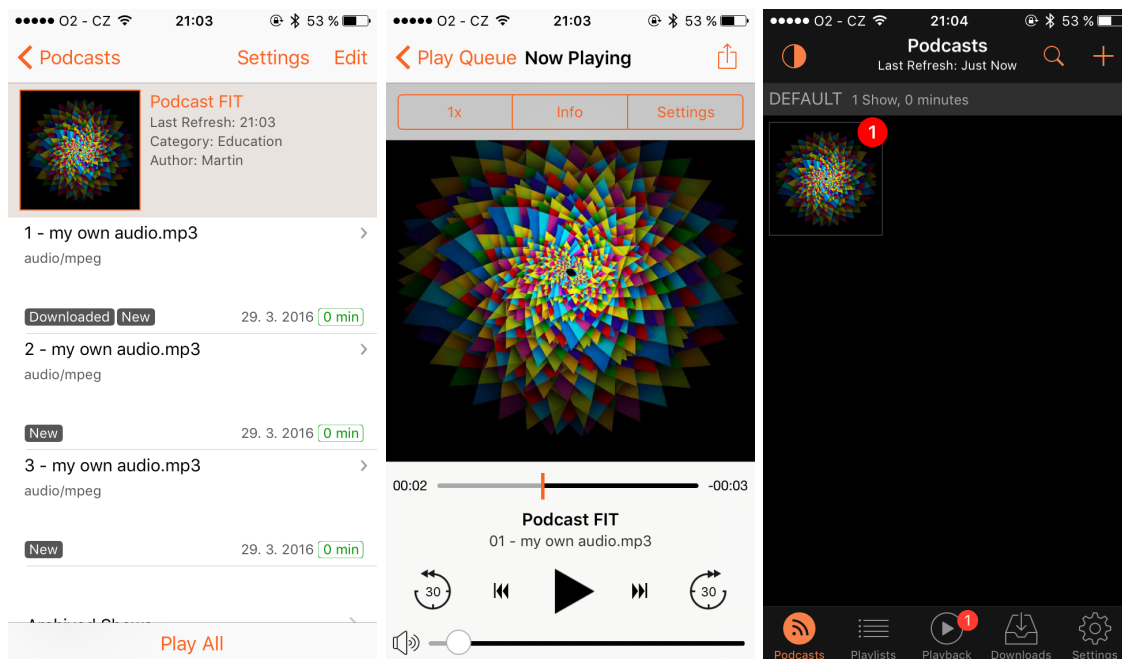
Obr. 3: Screenshoty aplikácie Overcast

Výhody a nevýhody aplikácie Overcast

- + Množstvo funkcií
- + Podpora pre Apple Watch
- + Podpora streamu
- Iné písmo v aplikácii
- Niektoré funkcie sú dostupné až po zaplattení
- V playliste vždy iba nové epizódy

RSSRadio Podcast Downloader App

Aplikácia RSSRadio Podcast Downloader App je dostupná v dvoch variantách, platená a zdarma. Aplikácia zdarma obsahuje reklamy a obmedzený počet odoberania Podcastov. Plná verzia aplikácie za cenu 3.99 € je zbavená všetkých reklám a môžeme v nej využiť neobmedzený počet odoberania Podcastov. Táto aplikácia je veľmi prehľadná a pôsobí dosť jednoducho, má obrovský počet nastavení a rôznych utilít, ktoré spríjemnia jej využívanie. Je možné ju prepnúť na tmavú tému, ak používateľovi svetlá téma nevyhovuje. Takisto podporuje aj iCloud synchronizáciu a zálohu všetkých dát.



Obr. 4: Screenshoty aplikácie RSSRadio Podcast Downloader App

Výhody a nevýhody aplikácie RSSRadio Podcast Downloader App

- + Množstvo nastavení
- + Podpora pre Apple Watch
- + Podpora iCloud
- Neprehľadný zoznam Podcastov
- Vo verzii zdarma reklamy a obmedzený počet odoberania Podcastov
- Nepodporuje stream

Z vybraných aplikácií vyplýva to, že všetky dodržiavajú viac-menej jednotné UI rozhranie a líšia sa skôr funkciami. V niektorých prípadoch sú podľa môjho uváženia aplikácie až príliš zložité a ponúkajú veľké množstvo funkcií, ktoré sú v mnohých prípadoch zbytočné. Niektoré aplikácie fungujú tak, že po pridaní nového podcastu sa automaticky začne sťahovať jedna jeho posledne pridaná epizóda, čo v mojom prípade nebolo vôbec pohodlné a tento krok je zbytočný. Reklamy v aplikáciách dosť často pôsobia otravne a kazia celkový dojem. Niektoré aplikácie obsahujú aj platené funkcie a ponúkajú možnosť doplatiť si za niečo, čo by sme chceli navyše využívať oproti neplateným verziám.

Kapitola 3

Tvorba aplikácií v prostredí Apple

AppStore obsahuje niečo cez 1 500 000 aplikácií, medzi ktorými nájdeme populárne aplikácie ako Facebook, Twitter, Youtube atď, alebo aj množstvo hier, utilít, aplikácií pre zdravie, objednávanie jedla a iné. Toto množstvo je však o niečo menšie ako aplikácie dostupné na Google play, no treba podotknúť, že Apple má pomerne zložitý proces schvaľovania aplikácií a doslova nepustí každú aplikáciu do svojho AppStore.

3.1 Programovanie pre systém iOS

Operačný systém iOS je čoraz viac populárnejší pre jeho stabilitu a jednoduchosť. Nachádza sa na zariadeniach iPhone, iPod a iPad. Oblúbený je aj vďaka jeho podpore pre zariadenia s vždy aktuálnymi aktualizáciami. Spoločnosť Apple predstavila operačný systém iOS v9.3 dňa 21.3.2016, pochválila sa jeho vylepšeniami a takisto aj tým, že viac ako 90 % ich zariadení funguje na najnovšom systéme iOS. Operačný systém iOS je vždy zdarma a na aktuálny model telefónu / tabletu trvá podpora najnovšieho operačného systému podľa môjho odhadu približne 6 rokov. Takáto podpora zariadení síce nemusí byť priaznivá na vývoj aplikácií, ale je podstatné to, že aplikácia sa môže dostať na väčší počet zariadení.

3.1.1 Vývojové prostredie Xcode

Xcode je IDE¹ spoločnosti Apple, ktoré obsahuje balíček profesionálnych vývojárskych nástrojov pre vývoj softwaru pre operačné systémy iOS, OS X, tvOS, watchOS. Apple tento program poskytuje zdarma k stiahnutiu z Mac App Store, ale iba pre systémy OS X. Xcode podporuje programovacie jazyky C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, Rez a po novom Swift.

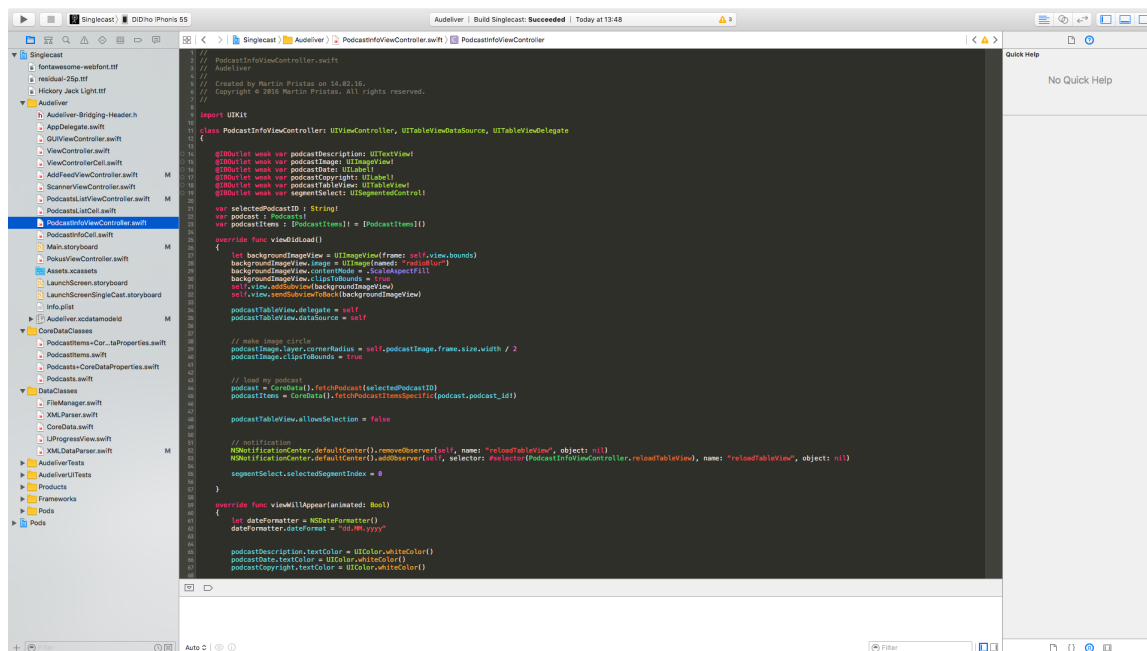
Rozhranie

Rozhranie Xcode integruje písanie kódu, návrh užívateľského rozhrania, správu aktív, testovanie a ladenie aplikácie. Všetko je navrhnuté v rámci jedného okna. Či sa aplikácia vyvíja v jazyku Swift, Objective-C, C, C++ alebo kombináciou týchto jazykov, Xcode kontroluje syntax kódu a indikuje chyby alebo varovania.

Xcode tiež obsahuje Interface Builder, pomocou ktorého môžeme jednoducho navrhovať rozhranie pre iOS, watchOS, tvOS alebo OS X zostavovaním okien, pohľadov, ovládacích prvkov a iných konfigurovateľných elementov z knižnice. Použitie StoryBoardu umožňuje

¹Integrated Development Environment

určenie toku aplikácie a prechody medzi scénami. Následne sa dajú graficky prepojiť objekty a prechody do kódu v aplikácii. Pre vytváranie univerzálneho dizajnu aplikácie slúži vlastnosť Auto Layout. Pomocou tejto vlastnosti môžeme definovať konštanty a rôzne obmedzenia pre jednotlivé elementy tak, aby sa automaticky prispôbili veľkosti obrazovky, veľkosti okna a lokalizácie.



Obr. 5: Aplikácia Xcode v hlavnom okne pri editovaní kódu.

Po tom, ako Xcode spustí aplikáciu v ladiacom móde, okamžite zahájí ladenie aplikácie. V prípade vývoja iOS, watchOS alebo tvOS aplikácie, xCode spustí aplikáciu v simulátore zariadenia alebo v zariadení pripojenom do počítača. Xcode tiež ponúka množstvo funkcií na ladenie a analýzu aplikácií, ktorý funguje na frameworku DTrace² vyvinutý spoločnosťou Sun Microsystems. Pomocou DTrace si používateľ môže sledovať bežice procesy, zbierať dáta a následne skúmať zozbierané dáta. Tieto *Instruments* (nástroje na testovanie a analýzu aplikácie) poskytujú možnosti sledovania jedného alebo viacerých procesov, zaznamenávať sekvenciu používateľových akcií a následne ich prehrať, profilovať aplikácie na OS X a iOS a ukladať záznamy užívateľského rozhrania.

3.1.2 Jazyk Swift

Jazyk Swift [5] je univerzálny programovací jazyk postavený na modernom prístupe k bezpečnosti, výkonu a programovacích návrhových vzorov. Cieľom jazyka Swift je vytvoriť najlepší dostupný jazyk, ktorý bude určený pre systémové programovanie, mobilné a desktopové aplikácie, škálovanie a cloudových služieb, ktoré sú čoraz viac populárnejšie. Dňa 3. decembra 2015 bol Swift uvedený ako OpenSource spoločnosti Apple. Tento programovací jazyk sa stáva čoraz viac populárnejší pre tvorenie aplikácií a má tak tendenciu nahradiť jazyk Objective-C, ktorý sa doteraz používa na vývoj aplikácií pre Apple produkty. Cieľom

²<https://wiki.freebsd.org/DTrace>

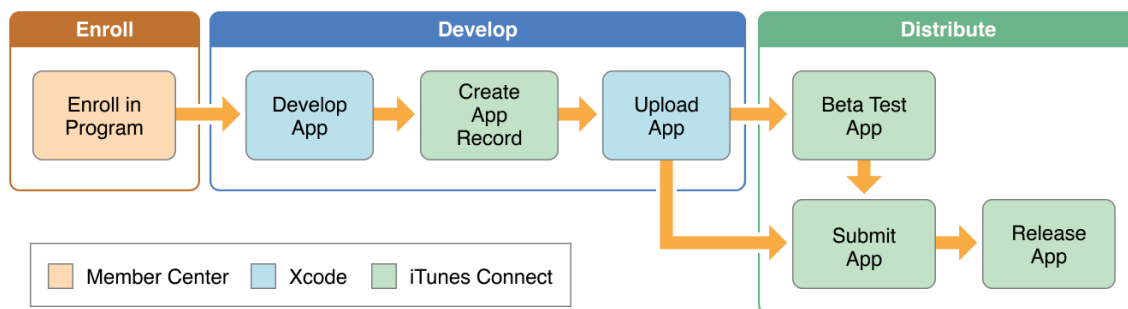
Swiftu nie je len nahradiť súčasný Objective-C jazyk pre aplikácie, ale nahradiť aj jazyky vychádzajúce z jazyka C (C, C++, a Objective-C). Swift je veľmi podobný skriptovacím jazykom, najmä programovaciemu jazyku Python a to pre jeho jednoduchosť, efektivitu a syntax.

3.1.3 Návrh aplikácií

Pri návrhu aplikácií pre systém iOS a ostatných treba dbať na dodržiavanie pokynov pre ľudské rozhranie (iOS Human Interface Guidelines) [2], kde popisujú ako by mala aplikácia správne vyzerat. Od návrhu aplikácie po finálny produkt je potrebné vykonať niekoľko krokov, ktoré sa odporúčajú dodržiavať. Je potrebné vypísať si všetky moduly, ktoré by mala aplikácia obsahovať, a ktoré by mohli jej používatelia využiť. Treba si určiť, pre koho bude aplikácia určená a kto je koncový používatel. Ďalej treba vyfiltrovať moduly z prvého kroku podľa toho, či spadajú do definície používateľov. Potom nasledujú kroky ako návrh UI, kde je potrebné držať sa pokynov pre ľudské rozhranie a následne vytvoriť prototyp aplikácie. Po prototypu sa vývoj iteruje do vtedy, až kým nedojdeme k správne mu „výsledku“.

3.1.4 Publikovanie aplikácie na AppStore

Pre publikovanie aplikácie je v provom rade potrebné mať zakúpený vývojarský účet, ktorý sa platí ročne. Pomocou zakúpeného vývojárskeho účtu môžeme publikovať aplikácie na App Store alebo aj ladiť aplikácie priamo na našom zariadení pripojenom k Macu. Každá aplikácia musí obsahovať svoj Bundle IDs [1], ktorý slúži ako identifikátor aplikácie. Pomocou tohoto identifikátoru môžeme pristupovať k službám ako Game Center alebo In-App nákupy a ďalšie. Pre zaslanie aplikácie do App Store musíme vlastniť certifikáty, ktoré podpisujú aplikáciu. Certifikáty sú určené na vývoj a distribúciu produktu. Po tom ako vytvoríme ID aplikácie, môžeme vytvoriť aplikáciu v iTunes Connect [3], čo je webový nástroj na publikáciu a spravovanie aplikácií pre App Store a Mac App Store. Potom je možné priamo z Xcode nahrať aplikáciu do iTunes Connect a dať ju na testovanie, prípadne poslať na posudok Applu. Ak posudok bude pozitívny, aplikáciu je možné uvrejniť v App Store.

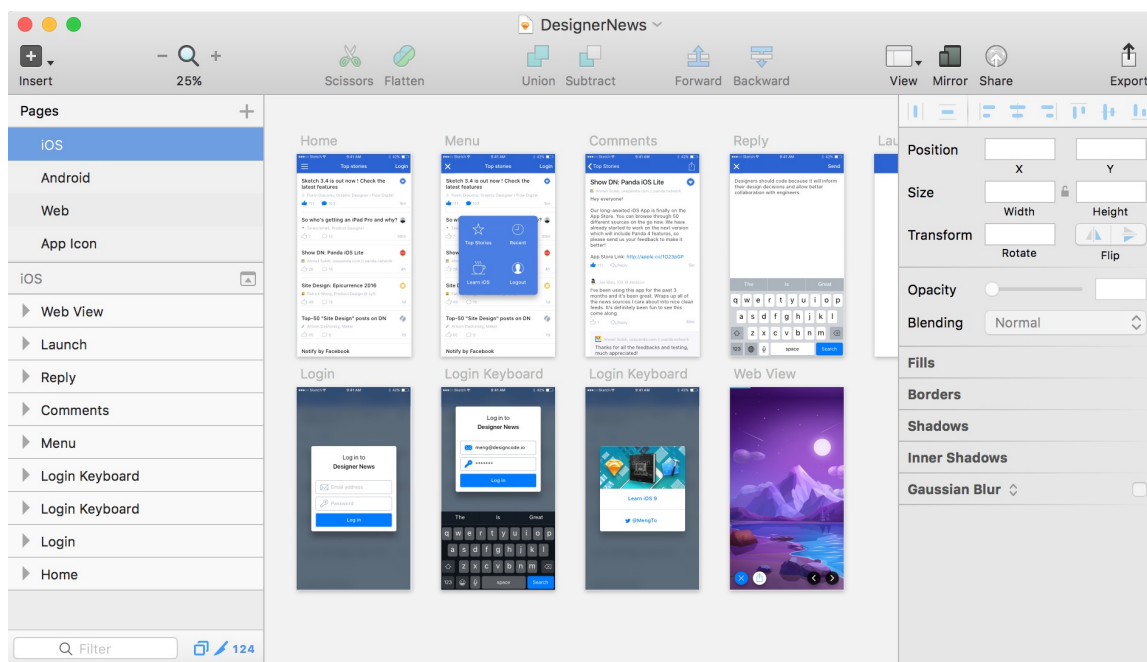


Obr. 6: Postup vývoja a publikovanie aplikácie [3].

3.2 Programy na návrh GUI

Tvorba GUI je veľmi dôležitou časťou vývoja aplikácie. V prípade, že chceme navrhovať GUI v programe ako napríklad Photoshop, Illustrator, Corel Draw a podobne, mali by sme sa riadiť pokynmi pre ľudské rozhranie. Tieto pokyny obsahujú aj zoznam odporúčaných grafických elementov, ktoré sa v systéme iOS používajú. Je dôležité dbať na používanie týchto elementov nakoľko sa aplikácie snažia byť čoraz viac jednotné z hľadiska rozhrania. Ak by sme rôzne prvky menili a dali im úplne iný tvar, bolo by to máťuce pre používateľov.

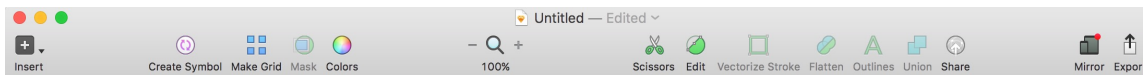
3.2.1 Program na kreslenie GUI – Sketch



Obr. 7: Program Sketch pri návrhu aplikácie a vlastnom rozložení viacerých kresliacich plátien.

Sketch³ je vektorový nástroj pre navrhovanie používateľského rozhrania, ktorý je dostupný len pre systém OS X. Je jednoduchý na prácu a zaberie na disku oveľa menej miesta ako napríklad Photoshop, teda približne 50MB. Sketch je skvelý nástroj pre navrhovanie iOS a Android aplikácií alebo Webových stránok. Celé rozhranie programu Sketch je podobné všetkým OS X aplikáciám s podobným účelom. Po otvorení obsahuje panel nástrojov, navigáciu a panel nastavovania prvkov. V celom programe sa pracuje len s pixelmi a ostatné jednotky ako napríklad centimetre a palce nie sú podporované. Panel nástrojov je plne konfigurovateľný a umožňuje tak prispôsobenie obľúbených prvkov.

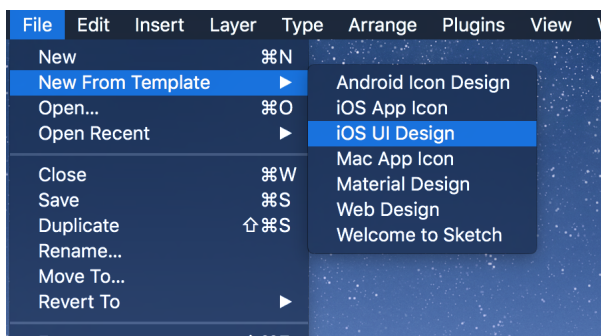
³<https://www.sketchapp.com/>



Obr. 8: Panel nástrojov v programe Sketch. Je očividné, že program Sketch je navrhnutý tak, aby bol minimalistický a používateľ sa do jeho ovládania rýchlo dostal.

Sketch podporuje rozdeliť svoj projekt na viac kresliacich plátien. Tieto plátna sú obrazovky aplikácie. Každé kresliace plátno predstavuje obrazovku alebo interakcie aplikácie v rámci obrazovky. Tento efektívny pracovný postup má zásadný význam pre moderného návrhára. Program tiež obsahuje stránky, ktoré určujú napríklad platformu alebo úplne iné rozlíšenie aplikácie. Pri väčšine projektov tak môžeme navrhnuť v programe Sketch pomocou viacerých stránok a kresliacich plátien aplikáciu pre Andorid, iOS, Web, alebo aj watchOS a podobne. Týmto spôsobom môžeme mať všetky tieto plátna a stránky v jednom súbore Sketch, čo ušetrí veľa času pri prepínaní medzi obrazovkami a platformami, a vytvorí oveľa lepší prehľad projektu.

Šablóny používateľského rozhrania sú súčasťou programu Sketch a umožňujú tak dodržiavať pokyny pre ľudské rozhranie už pri návrhu aplikácie. Program Sketch obsahuje najnovšie šablóny systému iOS 9, alebo aj šablóny pre material dizajn.



Obr. 9: Výber šablón UI v programe Sketch.

Pomerne dosť dôležité je to, že program Sketch má prispôbené exportovanie obrázkov a prvkov UI presne pre vývojový nástroj Xcode, čo umožňuje exportovať obrázky do 3 rôznych rozlíšení s pridaným suffixom @2x, @3x, ktoré Xcode podporuje pre Retina displeje. Ďalej obsahuje aj verzovací systém a vie sa tak vrátiť k predchádzajúcim úpravám projektu. Samozrejmosťou je aj automatické ukladanie a záloha projektu priamo do cloudového úložiska iCloud.

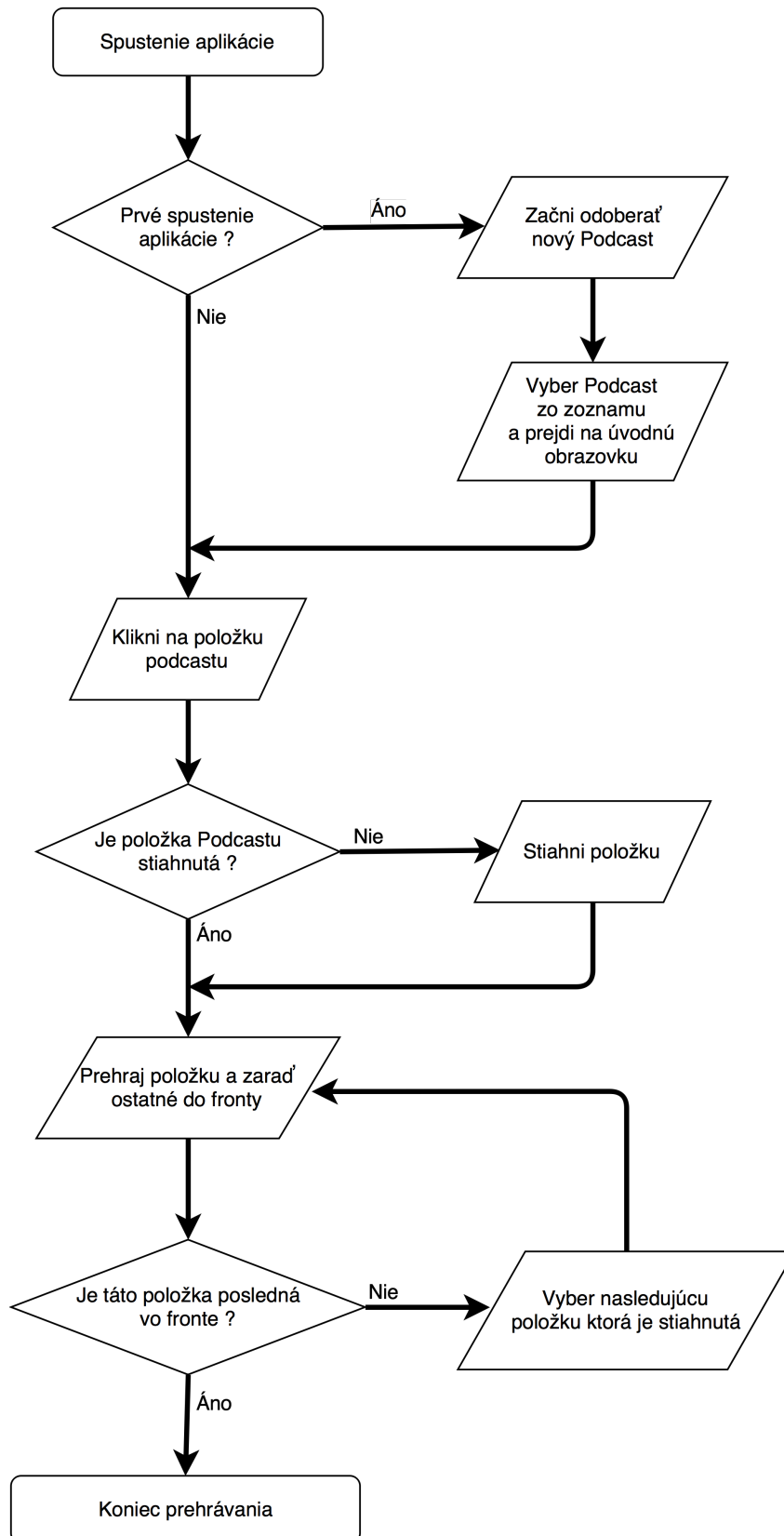
Kapitola 4

Návrh aplikácie Singlecast

Aplikácia by mala byť určená pre ľudí, ktorí obľubujú počúvanie audio nahrávok a jednoduchosť používania. Pri návrhu aplikácie som sa zamerlal hlavne na jednoduchosť zobrazenia prvkov v rozhraní a minimalizáciu okien a prechodov aplikácie. Používateľ nemusí mať skúsenosti s tvorením podcastov alebo ich odoberaním, no po začatí používania aplikácie by mu malo byť všetko jasné. Mal by patriť do mladšej vekovej skupiny, ktorá má skúsenosti s používaním operačného systému iOS, keďže aplikácia využíva systémové prvky používateľského rozhrania. Orientácia používateľa v aplikácii bude jednoduchá a nemusí sa učiť novým praktikám. V aplikácii sa počíta so základným používateľským rozhraním bez doplnkových funkcií, filtra epizód a samostatným nastavovaním odoberania podcastov. Používateľ by mal mať základné nároky na aplikáciu.

4.1 Vývojový diagram aplikácie a jej využitie

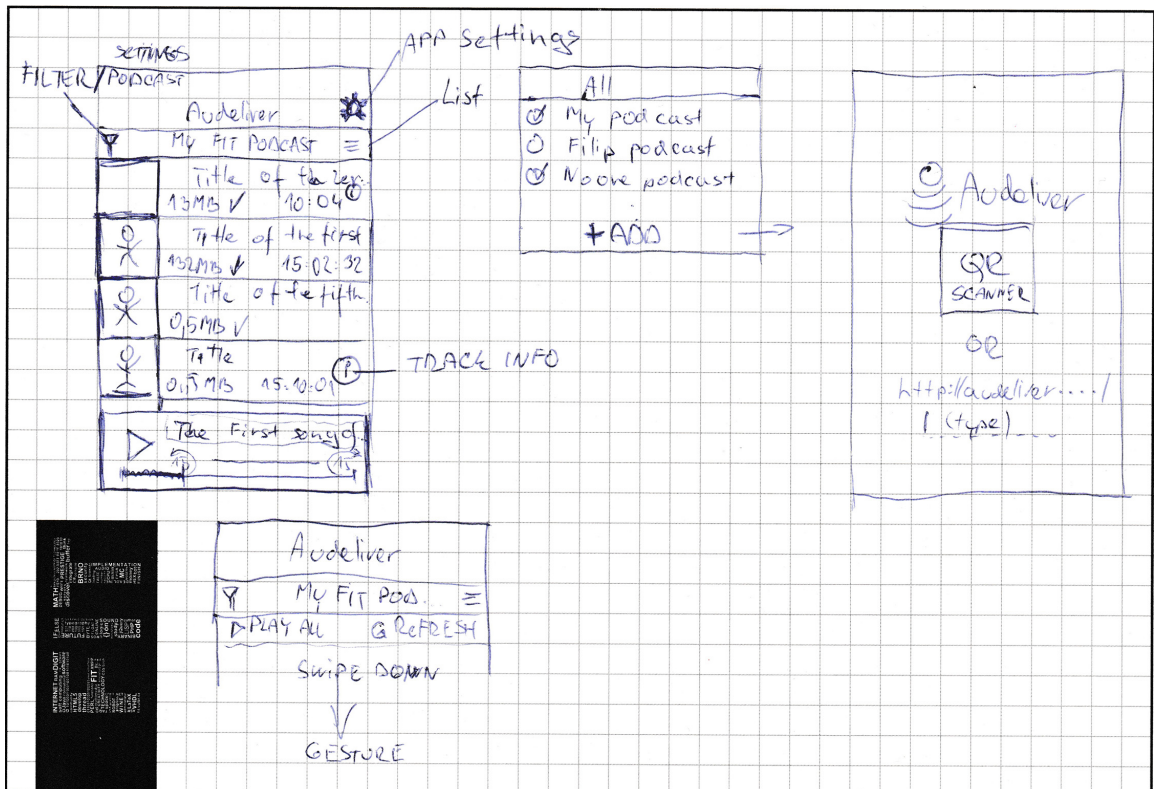
Vývojový diagram aplikácie (obr. 10) Singlecast je pomerne zjednodušený a ukazuje základnú činnosť aplikácie. Vytvoril som návrh aplikácie, ktorá pôsobí jednoducho a je zbavená zložitosti, ktorú musí užívateľ skúmať pri ostatných aplikáciách. Po návrhu vývojového diagramu bolo jasné, čo má aplikácia robiť a ako má slúžiť. Po prvom zapnutí aplikácie má užívateľ možnosť okamžite prejsť k odoberaniu Podcastu. Po pridaní nového Podcastu do aplikácie sa otvorí zoznam Podcastov, v ktorom si môžeme zapnúť a vypnúť podcasty, ktoré sa nám zobrazia na úvodnej obrazovke. Úvodná obrazovka preto narába už len so zapnutými Podcastmi a ukazuje ich položky. Ak sa klikne na stiahnutú položku, spustí sa prehrávanie, a ak epizóda stiahnutá nieje, začne sa sťahovať.



Obr. 10: Zjednodušený vývojový diagram Podcastovej aplikácie.

4.2 Návrh UI

UI som sa rozhodol navrhovať najskôr na papier, aby som si premyslel štruktúru návrhu celej aplikácie. Pri návrhu som však jemne vychádzal už z navrhnutých aplikácií, ktoré sú dostupné a udávajú tak to, ako má aplikácia na príjem Podcastov vyzerať. Prvotný návrh UI zobrazoval úvodnú obrazovku so zoznamom položiek jednotlivých Podcastov. Dôležitý bol aj návrh pridávania Podcastov, pri ktorom som pridal možnosť pridávanie Podcastu pomocou QR kódu, ktorý snímala automaticky kamera mobilu alebo tabletu. Po niekoľkých návrhoch na papier som pristúpil k návrhu celého GUI v programe Sketch, aby bolo jednoduchšie si predstaviť jednotlivé elementy aplikácie a zároveň tak dodržať HIG¹.



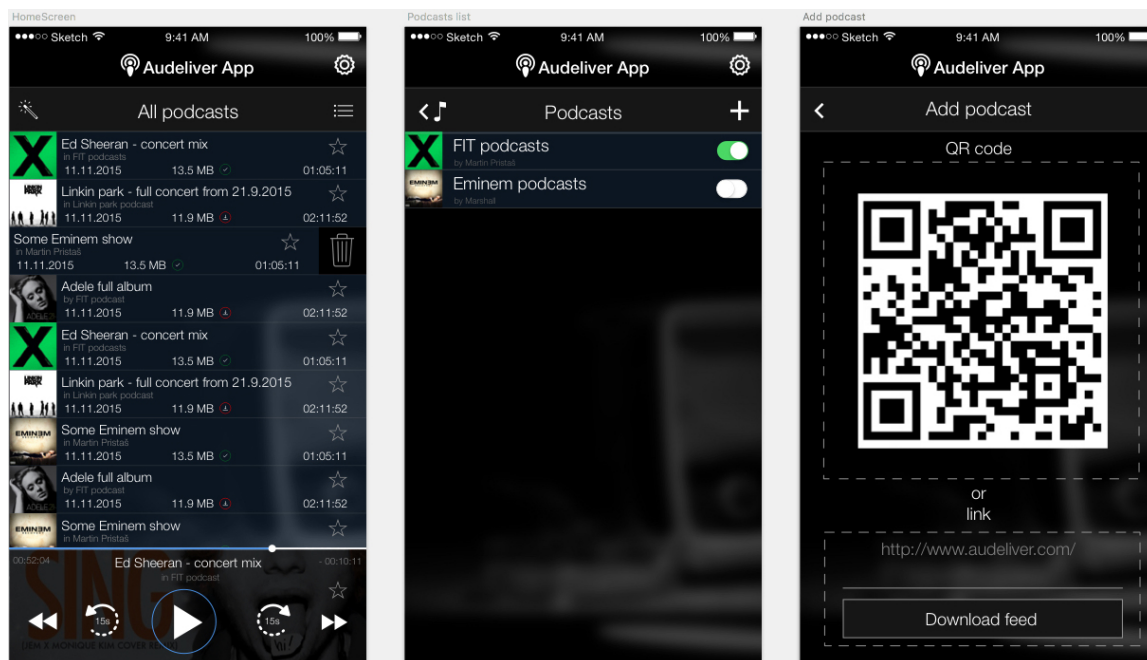
Obr. 11: Prvý návrh UI aplikácie.

4.2.1 GUI v programe Sketch

Aplikáciu Sketch som predstavil v Kapitole 3.2.1. Táto aplikácia mi umožnila pohodlne navrhnuť grafické rozhranie mobilnej aplikácie pomocou jej nástrojov. Tipy a triky som čerpal z knihy [14], ktorá popisuje postup správneho návrhu grafického rozhrania aplikácií. Pri návrhu grafického rozhrania som si vytvoril projekt pre návrh aplikácie na iPhone 6S. Podľa predlohy malo plátno návrhu 375px × 667px, čo však ale nebolo až tak podstatné, pretože Sketch je program, ktorý pracuje s vektormi. Návrh aplikácie som realizoval pomocou UI šablón systému iOS 9, ktoré sú dostupné už priamo v Sketchi a vkladajú sa ako symboly prvkov, ktoré sú editovateľné.

¹Human Interface Guidelines

Keďže som pri návrhu mock-upov nedbal na to, aký farebný štýl bude mať aplikácia, zvolil som tmavé pozadie s tmavým obrázkom a zafarbenie niektorých prvkov UI do sýtej modrej farby.



Obr. 12: Návrh vlastného GUI v programe Sketch.

Výhodou v programe Sketch bolo exportovanie obrázkov a ikon pre použitie v Xcode. Program umožňuje exportovať obrázky v rozlíšení 1x, 2x a 3x, čo je použiteľné pre Apple zariadenia s vyšším rozlíšením. Všetky prvky, ako ikony aplikácie alebo tlačidlá pre prehrávač som si vyexportoval hromadne z programu Sketch.

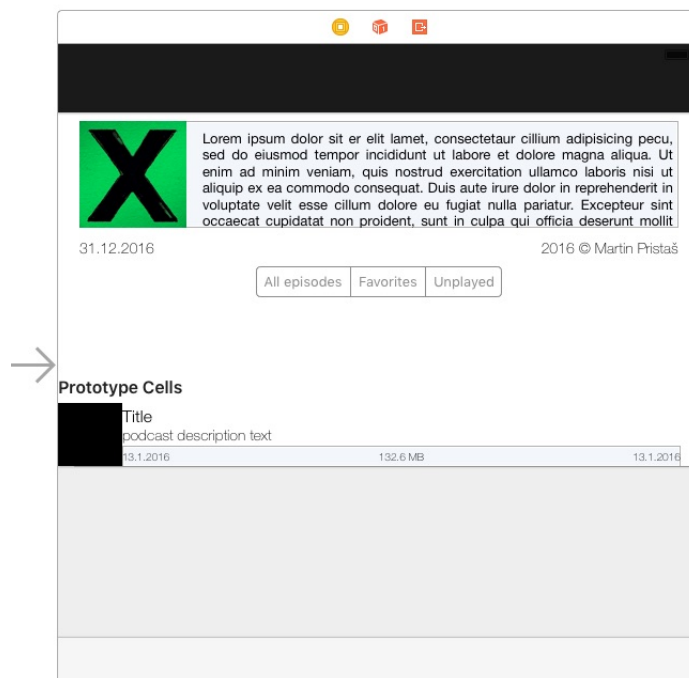
4.2.2 Tvorba GUI pomocou StoryBoardu v programe Xcode

V StoryBoarde sa tvorba grafického rozhrania líši s programami, v ktorých sa grafické rozhranie navrhuje. StoryBoard obsahuje nástroj AutoLayout, ktorý slúži na automatické prispôbovanie prvkov v závislosti od veľkosti okna a teda aj rozlíšenia. Pre moju aplikáciu som potreboval navrhnuť jednotné grafické rozhranie, ktoré by fungovalo správne ako pre iPhone, tak aj pre iPad. Pre každé okno aplikácie som využil vlastný View Controller [6], ktorý obsahoval vlastnosti tried jednotlivých častí aplikácie. Ďalej som využil nástroj AutoLayout a všetkým prvkom prideloval rôzne konštanty, ktoré zabezpečovali ich vzhľad ako na iPhone, tak aj na iPade. Na správne nastavovanie konštant v AutoLayout som využil knihu [16], ktorá je síce staršia a neobsahuje aktuálne trendy AutoLayout a úpravy v iOS 9, no na jednoduchší projekt postačila dostatočne.

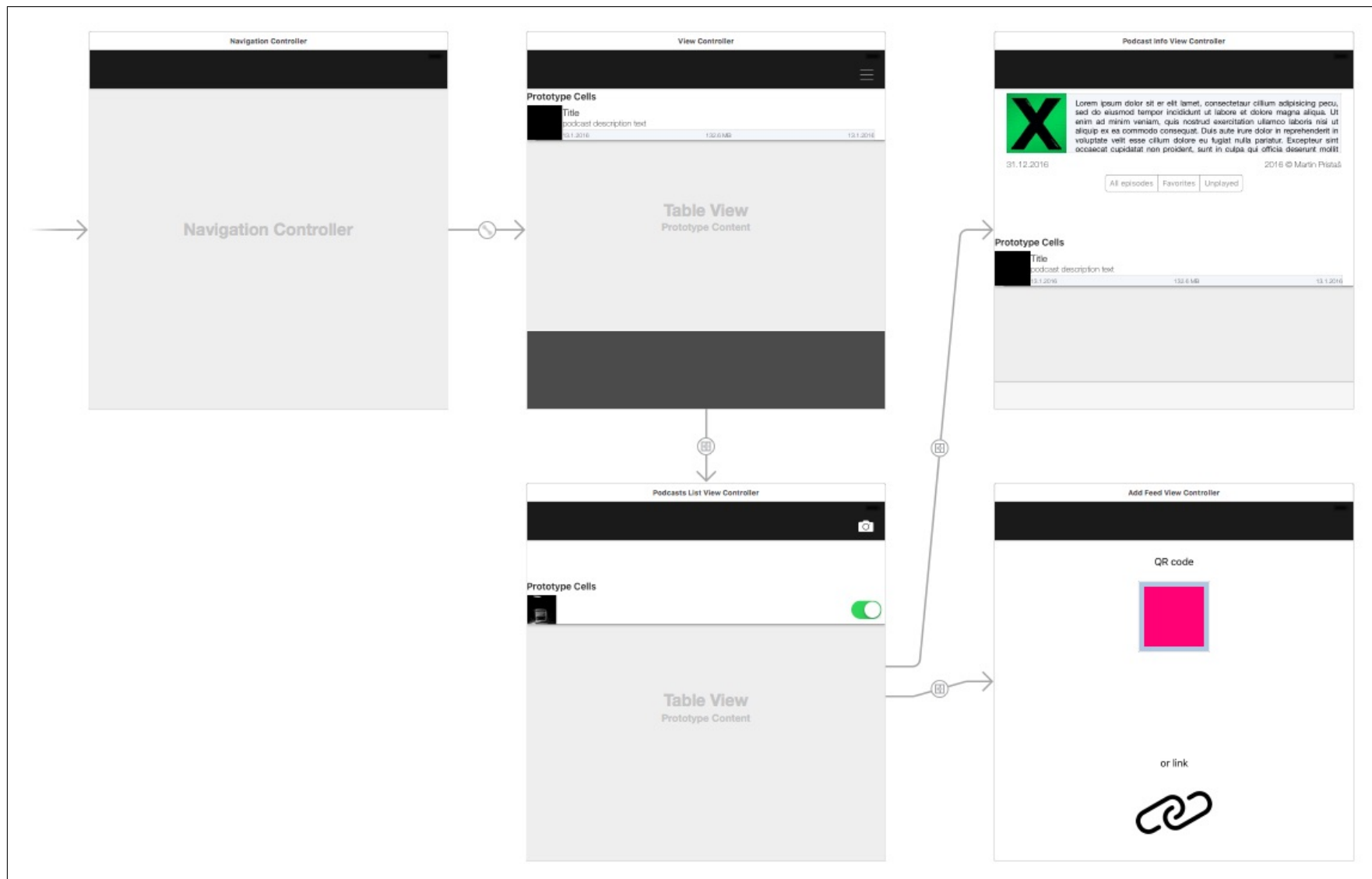
Celý StoryBoard riadi jeden Navigation Controller, ktorý slúži na navigáciu medzi oknami aplikácie. Následne každé okno aplikácie má vlastný View Controller, ktoré sú medzi sebou prepojené.

StoryBoard v programe Xcode umožňuje navrhnuť prototypový ViewController, ktorý bude buď univerzálny alebo prispôsobiteľný pre iPhone alebo iPad. Ja som zvolil univerzálny návrh pre iPhone a iPad. Všetky elementy z návrhu StoryBoardu sa dajú napojiť priamo

do jeho triedy, v ktorej sa môžu meniť ich vlastnosti. Toto platí pri elementoch, ktoré patria do triedy ViewController. Ďalej je možnosť vytvoriť prototyp bunky pre TableView (tabuľkové zobrazenie), kde sa budú generovať bunky pre všetky riadky v tabuľke, teda zoznamy Podcastov alebo zoznamy epizód v hlavnom okne aplikácie.



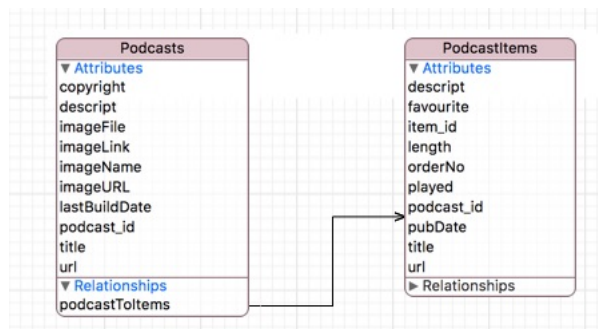
Obr. 13: Prototyp okna pre zobrazenie informácií o Podcaste.



Obr. 14: Návrh StoryBoardu v programe Xcode.

4.3 Tvorba databázového modelu v Xcode

Tvorba jednoduché databázy v programe Xcode vychádza zo vzoru XML štruktúry Podcastov. Podľa tejto štruktúry som vybral najdôležitejšie tagy pre popis Podcastu a pre jeho položky. Tvorba databázového modelu sa robí vytvorením entít, v ktorých následne vytvorím všetky potrebné premenné spolu s názvami. Potom nadradenej entite `Podcasts` pridám vzťah k entite `PodcastItems`.



Obr. 15: Entity v Xcode databáze pre Aplikáciu Singlecast.

Atribúty ako *copyright*, *descript*, *imageLink*, *imageName*, *imageURL*, *podcast_id*, *title*, *url*, *item_id* boli definované ako textový reťazec (String), a pri *imageFile* sa jednalo o binárne data. Ďalšie atribúty *lastBuildData* a *pubDate* som nastavil ako typ `NSDate`, ktorý predstavuje objekt k časovému intervalu s referenčným dátumom (00:00:00 UTC 1. januára 2001).

4.4 Vhodné knižnice pre aplikáciu

Podpora externých knižníc pre Xcode je v zásade samozrejmosťou a umožňuje tak použiť akýkoľvek framework aký potrebujeme, a zároveň je kompatibilný s verziou systému, pre ktorý sa Aplikácia vytvára. Na pridávanie a aktualizáciu frameworkov som použil manažér `CocoaPods`, ktorý umožňuje inštaláciu akéhokoľvek frameworku, ku ktorému má prístup. `CocoaPods` umožňuje inštaláciu a aktualizáciu knižníc pomocou konzoly a používa na to len súbor `Podfile`, z ktorého číta, aké frameworky aplikácia obsahuje. `CocoaPods` mi zjednodušilo prácu pri prechode na novšiu verziu Swiftu a tiež umožnilo prepojiť knižnicu napísanú v jazyku `Objective-C` s mojou aplikáciou, ktorá je písaná vo `Swift`. V aplikácii som využil nižšie popísané frameworky:

Alamofire – HTTP komunikácia

`Alamofire` [10] je knižnica na HTTP komunikáciu napísaná v jazyku Swift, ktorá je dostupná pre platformy iOS, OS X, tvOS a watchOS. Túto knižnicu som využil pri sťahovaní obrázka Podcastov a aj pri sťahovaní mp3 súborov z položiek Podcastov. Výhodou tejto knižnice je sťahovanie dát so zistením progresu sťahovania. Implementácia knižnice je jednoduchá a využíva `CocoaPods` modul. `Alamofire` umožňuje sťahovanie ale aj odosielanie dát s využitím zistenia, koľko dát bolo stiahnutých/odoslaných z celkového množstva.

Zdrojový kód 1: Príklad sťahovania súboru pomocou Alamofire

```
Alamofire.download(.GET, "https://httpbin.org/stream/100", destination:
    destination)
    .progress { bytesRead, totalBytesRead, totalBytesExpectedToRead in
        print(totalBytesRead)

        // This closure is NOT called on the main queue for performance
        // reasons. To update your ui, dispatch to the main queue.
        dispatch_async(dispatch_get_main_queue()) {
            print("Total bytes read on main queue: \(totalBytesRead)")
        }
    }
    .response { _, _, _, error in
        if let error = error {
            print("Failed with error: \(error)")
        } else {
            print("Downloaded file successfully")
        }
    }
}
```

SWXMLHash – Jednoduché parsovanie XML súboru

SWXMLHash [13] je alternatívne riešenie parsovania XML súborov v jazyku Swift, ktoré prevádza XML súbor do slovníka polí. Narozdiel od NSXMLParser, čo je stavaná knižnica Apple sa k jednotlivým elementom pristupuje jednoducho a okamžite, ak samozrejme poznáme štruktúru XML kódu.

Zdrojový kód 2: Príklad použitia SWXMLHash

```
<root>
  ...
  <catalog>
    <book><author>Bob</author></book>
    <book><author>John</author></book>
    <book><author>Mark</author></book>
  </catalog>
  ...
</root>

xml["root"]["catalog"]["book"][1]["author"].element?.text
```

KDCircularProgress – Kruhový ukazovateľ progresu

Framework KDCircularProgress [8] slúži na zobrazovanie kruhového progresu, ktorý využijem v aplikácii na zobrazovanie stavu prehratia alebo sťahovania položky. Je napísaný v jazyku Swift a obsahuje veľké množstvo nastavení zobrazenia. Je možné nastavovať hrúbku čiary progresu, farbu a dokonca aj prechody medzi viacerými farbami. Súčasťou je aj počiatkový uhol, žiara stopy alebo zaoblené okraje stopy.

Zdrojový kód 3: Príklad použitia KDCircularProgress

```
let progress = KDCircularProgress(frame: CGRect(x: 0, y: 0, width: 300,
    height: 300))
progress.startAngle = -90
progress.progressThickness = 0.2
progress.trackThickness = 0.7
progress.clockwise = true
progress.center = view.center
progress.gradientRotateSpeed = 2
progress.roundedCorners = true
progress.glowMode = .Forward
progress.angle = 300
progress.setColors(UIColor.cyanColor() ,UIColor.whiteColor())
view.addSubview(progress)
```



Obr. 16: Vzhľad KDCircularProgress

TLYShyNavBar – Úprava navigačného panela

TLYShyNavBar [11] prináša veľmi jednoduché rozšírenie navigačného menu, ktoré obsahuje aplikácia a je vstavané v iOS aplikáciách. Rozšírenie spočíva v tom, že sa tento framework deleguje na navigačné menu aplikácie a aj na rolovanie obrazovky. Zachytáva systémové hlásenia o pohybe rolovania a následne animuje skrytie a zobrazenie navigačného panela. Toto jednoduché a praktické riešenie prináša po aplikovaní väčší priestor zobrazovania dát na obrazovke, v mojom prípade to je zoznam položiek Podcastov. Tento framework je napísaný v Objective-C a na to, aby som ho mohol použiť vo svojom projekte stačí vytvoriť jednoduché premostenie môjho projektu s frameworkom, aby som ho mohol v kóde využívať.

Kapitola 5

Tvorba aplikácie Singlecast

Pri tvorbe aplikácie som vychádzal zo svojich počiatočných návrhov a samozrejme pokynov pre ľudské rozhranie od spoločnosti Apple. Jednotlivé prvky GUI rozhrania aplikácie som implementoval tak, ako boli navrhnuté v programe Sketch, kde som si celý vzhľad aplikácie pripravil.



Obr. 17: Navrhnuté logo pre aplikáciu SingleCast

Pred tvorbou aplikácie som si naštudoval niektoré veci z kníh [7, 15], aby som sa vyhol zlému návrhu aplikácie už od začiatku. Pred samotným vytváraním grafického rozhrania som najprv vyskúšal prácu s frameworkom Alamofire spomenutým v kapitole 4.4, ktorým som spočiatku sťahoval a triedil dáta z URL adresy Podcastu. Po sťahovaní a vytriedení XML kódu som mohol dáta pridávať do vnútornej databázy aplikácie alebo ich aktualizovať.

Po vytvorení aktuálneho zoznamu Podcastov z URL odkazu som prešiel na zobrazenie zoznamu Podcastov a ich výber, potom zobrazenie jednotlivých položiek Podcastov a následne tvorbu prehrávača spolu so sťahovaním jednotlivých položiek.

5.1 Sťahovanie a parsovanie XML feedu a ukladanie dát v aplikácii

Aplikácia obsahuje samostatné okno na pridávanie URL adresy Podcastu, a to pomocou vloženia linku do textového pola, alebo naskenovaním QR kódu, ktorý túto adresu obsahuje. Po zoskenovaní alebo vložení URL adresy sa táto adresa odovzdá triede nazwanej `XMLDataParserClass`, ktorá obsahuje metódu nazvanú `parseByUrl`. Táto metóda zaručí správne stiahnutie XML kódu z URL adresy. Po správnom stiahnutí XML kódu, ktorý sa kontroluje pomocou frameworku Alamofire, sa volajú funkcie `readPodcastData` a `readPodcastItems`. Tieto funkcie triedia data Podcastu a jednotlivých položiek, ktoré sa v ňom nachádzajú.

Funckia `readPodcastData` využíva k uloženiu dát štruktúru `podcastStruct`, ktorá obsahuje nasledujúce premenné: `title`, `description`, `url`, `lastBuildDate`, `copyright`, `imageURL`, `imageTitle`, `imageLink`, `podcast_id`, teda tie, ktoré sa priamo uložia aj v databáze. Podobne ako `podcastStruct` pri funkcii `readPodcastData` ukladá data o Podcaste, tak funkcia `readPodcastItems` prečíta data z XML kódu o položkách Podcastu. Toto vykoná v cykle pre všetky položky podcastu a vytvorí pole všetkých položiek, teda štruktúru `podcastItemStruct`, ktorá obsahuje tieto premenné: `title`, `description`, `url`, `length`, `pubDate`, `item_id` a `podcast_id`.

Premenné typu `String` reprezentujúce `podcast_id` a `item_id` sú generované vždy pomocou URL adresy Podcastu alebo jednotlivých položiek, kde sa znaky „. / :“ nahradia znakmi „-“. a reťazce obsahujúce „http://“ a „.xml“ sa zrušia úplne. Takto vytvorím unikátne ID pre Podcast a aj pre jeho položky. Pomocou tohoto ID sa vyberajú dáta z databázy a tak je potrebné, aby toto ID ostalo vždy rovnaké pre jeden konkrétny Podcast, napríklad v prípade keď aktualizujem dáta Podcastu.

Po prečítaní dát z XML kódu trieda `XMLDataParserClass` ich odovzdá triede `CoreData`, ktorá ich spracuje pomocou metód `addPodcastToCoreData` a `addItemToCoreData`. Trieda `CoreData` pri pridávaní Podcastu najprv overí, či sa v databáze aplikácie nachádza tento podcast. Ak áno, prebehne jeho aktualizovanie a obnovia sa jeho základné položky vrátane obrázka. Sťahovanie obrázku podcastu má na starosti tiež framework Alamofire. Pomocou tohto sťahovania sa stiahnu binárne data obrázka a uložia sa v databáze aplikácie. Podobný postup ako pri pridávaní podcastu sa vykonáva aj v metóde `addItemToCoreData`, kde sa pridávajú položky podcastu, avšak najprv je potrebné overiť to, či databáza neobsahuje položky, ktoré sa už v novom XML kóde nenachádzajú. Toto všetko overí rovnaká metóda pomocou vyvolania dát z databázy.

Po pridaní nového alebo aktualizovaní Podcastu, ktoré prebehne úspešne sa aplikácia vráti na zoznam Podcastov.

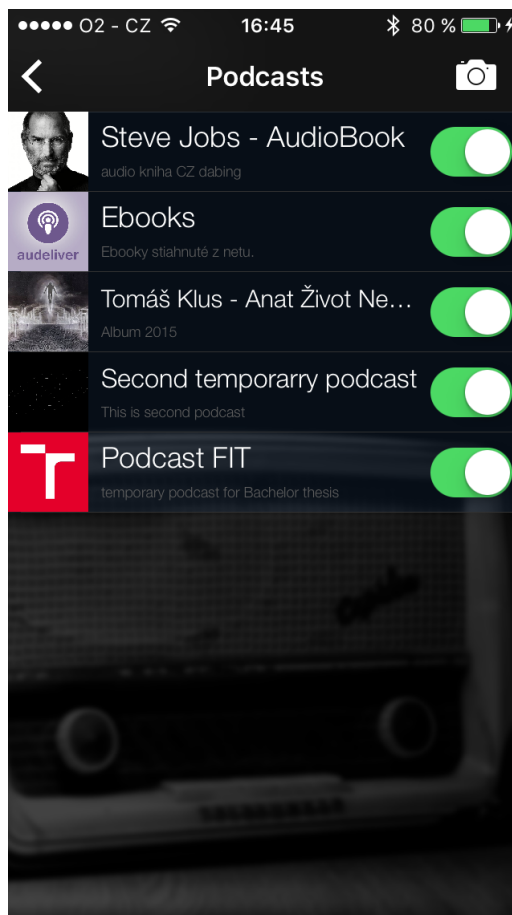


Obr. 18: Vzhľad pridávania Podcastu pomocou QR kódu alebo odkazu

5.2 Zobrazenie zoznamu dostupných Podcastov v aplikácii

Toto okno slúži na selektovanie Podcastov, ktoré chceme mať v hlavnom okne aplikácie. Hlavné okno obsahuje navigačný panel, ktorý je v celej aplikácii rovnaký a tabuľkové zobrazenie dostupných Podcastov. Po otvorení tohto okna sa v tabuľkovom zobrazení ukážu dostupné Podcasty, ktoré máme v aplikácii. Každá bunka tabuľky predstavuje jeden určitý Podcast, ktorý je definovaný názvom, popisom a obrázkom. Vedľa Podcastu je jednoduché prepínacie tlačidlo `UISwitch`, ktoré slúži na zapnutie/vypnutie Podcastu z výberu všetkých Podcastov, ktoré sa zobrazia v hlavnom okne aplikácie. Po kliknutí na bunku tabuľky je možné prejsť na Informácie o Podcaste.

Vytiahnutie Podcastov z databázy má na starosti metóda `fetchAllPodcasts` ktorá je v triede `CoreData` a vracia pole Podcastov. Z tohto pola sa tvoria bunky pre tabuľkové zobrazenie, ktorá vracia bunku tabuľkové zobrazenia `UITableViewCell`.

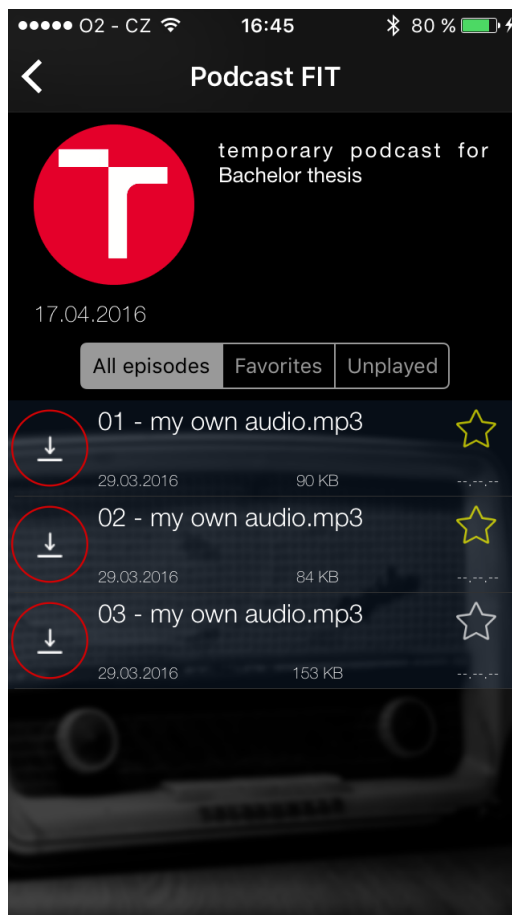


Obr. 19: Vzhľad zoznamu Podcastov v aplikácii

5.3 Informácie o Podcaste

Zobrazenie informácií o Podcaste som spočiatku nemal v pláne zapracovať do aplikácie, preto sa nenachádza ani v návrhu aplikácie. Rozhodol som sa, že je potrebné zobrazit základné informácie o Podcaste a epizódach, ktoré sa v ňom nachádzajú, tak som túto obrazovku spracoval už bez dodatočného navrhovania v programe Sketch. Obrazovka sa delí na dve časti, ktoré ponúkajú informácie o Podcaste.

Vrchná časť obrazovky ukazuje obrázok podcastu v kruhovom tvare, text popisu podcastu, copyright a dátum poslednej úpravy Podcastu. Pod touto časťou sa nachádza tabulkové zobrazenie jednotlivých položiek Podcastu, ktoré sú rozdelené do troch kategórií. V prvej kategórii je zoznam všetkých položiek, v druhej sa nachádza zoznam obľúbených epizód a v tretej prehraté epizódy. V tejto časti je možné kliknutím na hviezdičku označovať epizódy Podcastu za obľúbené.



Obr. 20: Vzhľad informácií o Podcaste

5.4 Zoznam všetkých nahrávok a prehrávač v aplikácii – hlavné okno aplikácie

Zoznam všetkých epizód z Podcastov, ktoré si používateľ zvolil predstavuje hlavné okno aplikácie a tak sa po spustení zobrazuje ako prvé. Po zapnutí aplikácie sa na tomto okno skryje prehrávač. Zoznam tvorí klasický výpis epizód Podcastov v tabuľkovom zobrazení `UITableView`, ktorý získava data z Databáze aplikácie.

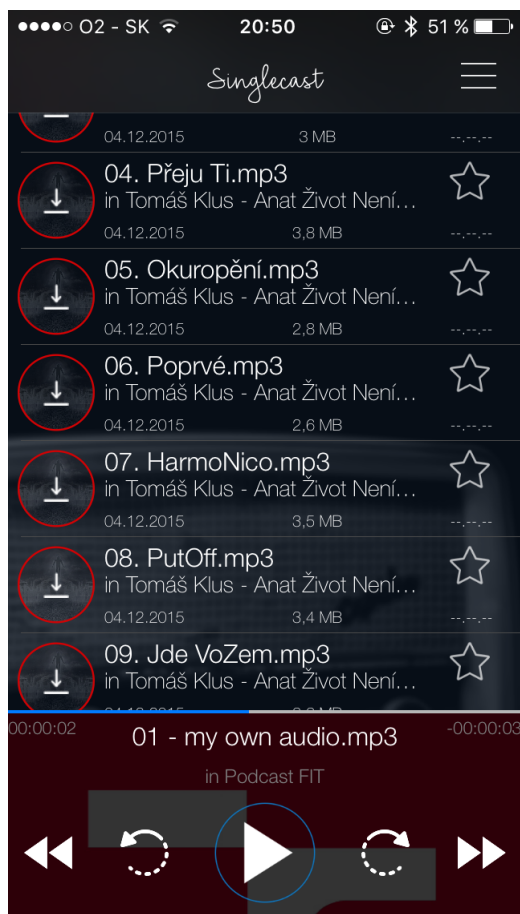
Zoznam umožňuje aj prediktívne vyhľadávanie v názve a popise epizód. Vyhľadávanie v názve a popise epizód funguje na princípe dopytovania databázy s prediktormi `NSPredicate` [4]. Po každej zmene textu vo vyhľadávacom poli sa volá delegátna funkcia `searchBar` (`searchBar: UISearchBar, textDidChange searchText: String`), v ktorej spracovávam zadaný text a predávam text triede `ClassData`, ktorá vo funkcii `fetchPodcastItemsBySearchBar` vráti nájdené epizódy.

V tomto okne aplikácie využívam aj framework `TLYShyNavBar` (sekcia 4.4), ktorý delegujem na `ScrollView` tabuľkového zobrazenia epizód a nastavím mu parametre. Pre jednoduchosť som využil iba obyčajné skrývanie navigačného menu vo vrchnej časti obrazovky pri rolovaní smerom nadol. Navigačné menu sa znova zobrazí pri rolovaní nahor. Navigačné menu má nadpis `Singlecast` v úhľadom `Fonte` typu `Hickory Jack` a pôsobí ako vpísané

ručne.

V zozname epizód je možné vyhľadávanie spomenuté vyššie, sťahovanie epizódy alebo pustenie prehrávania. Po kliknutí na bunku tabuľky sa volá delegátna metóda `tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath)`, v ktorej overujem, či je daná epizóda už stiahnutá alebo nie. Ak je epizóda stiahnutá, tak sa pustí prehrávanie kliknutej epizódy. V prípade, že epizóda má už prehratú nejakú časť, tak sa pokračuje v skončenom čase, ktorý bol uložený do databázy k epizóde. Ak epizóda dostupná nie je, je potrebné ju stiahnuť. Pre stiahnutie sa používa trieda `FileManager`, ktorá obsahuje metódu `addFileToDownloadQueue`. Tejto metóde odovzdávam položku Podcastu a v prípade, že sa aktuálne nič nestahuje, spustí sa jej sťahovanie. Ak momentálne aplikácia sťahuje nejakú epizódu, vloží sa nová epizóda na stiahnutie do fronty. Fronta sa postupne spracováva podľa poradia vloženia.

Prehrávač v aplikácii je jej dôležitá súčasť. Po zapnutí aplikácie sa prehrávač „skryje“ a používateľovi sa nezobrazuje. Určil som to tak preto, lebo som chcel, aby bol tento prehrávač jednoduchý a skrytý počas prehľadávania epizód Podcastov. Po kliknutí na epizódu, ktorá je stiahnutá sa vytvorí prehrávacia fronta z položiek v tabuľke, zo spodu obrazovky sa animáciou zobrazí prehrávač a spustí sa prehrávanie epizódy. Prehrávač využíva vstavaný Apple framework `AVAudioFoundation`, ktorý slúži na prácu s prehrávaním videa a audia. Tento framework obsahuje triedu `AVAudioPlayer`, ktorá má množstvo vlastností a používa sa primárne na prehrávanie audio súborov. Pred prehrávaním súboru sa overí, či súbor existuje, a či ho bude možné prehrať. Prehrávač na to využíva metódy `prepareToPlay` a `play`. Tieto metódy môžu vytvoriť výnimky v aplikácii a vypísať tak chybu ak sa súbor nepodarí prehrať. Prehrávač v aplikácii obsahuje časové údaje prehrávanej epizódy, koľko chýba do jej konca a koľko času je už prehratého, názov epizódy a názov Podcastu. Má ovládacie tlačidlá na spustenie a prerušenie prehrávania epizódy, tlačidlá na posunutie o niekoľko sekúnd a preskočenie epizódy na ďalšiu. V prehrávači sa automaticky preskakujú epizódy, ktoré nie sú stiahnuté a prehrávajú sa iba tie, ktoré sú dostupné v zariadení.



Obr. 21: Vzhľad hlavného okna aplikácie Singlecast so zoznamom epizód všetkých podcastov a prehrávačom.

Kapitola 6

Testovanie aplikácie a uverejnenie na AppStore

V poslednej kapitole v tejto bakalárskej práci sa budem venovať čiastočnej optimalizácii aplikácie a taktiež zhodnotením „boja“ s Apple Application Review, ktorí mi schvalovali aplikáciu, avšak neúspešne. S testovaním aplikácií som nikdy nepracoval a tak bolo potrebné si prečítať aspoň základy testovania aplikácií v programe Xcode, kde mi pomohla kniha [12], ktorá podrobne popisuje ako prebieha testovanie aplikácií. Podrobné testovanie aplikácie však nebolo treba vykonávať.

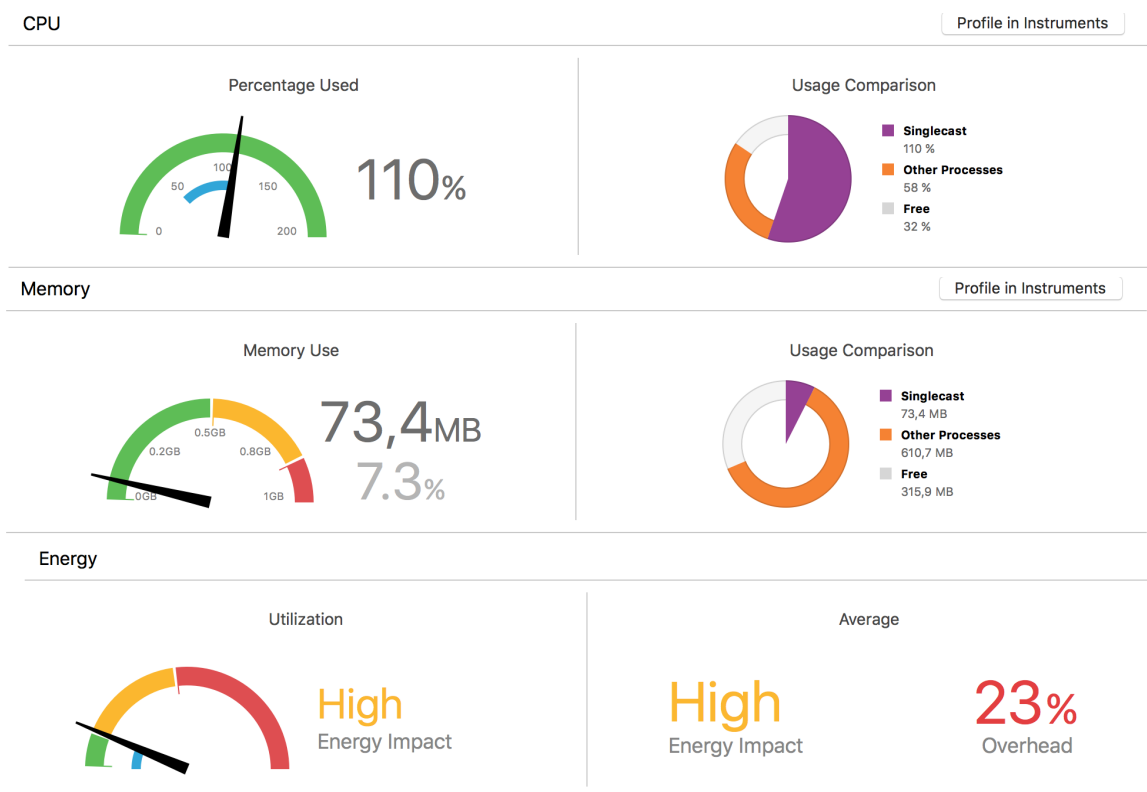
6.1 Hardwarové nároky aplikácie

Keďže aplikácia je tvorená pre mobilné telefóny iPhone a tablety iPad s operačným systémom iOS a minimálnou verziou 9.1, je dostupná pre iPhone 4S, 5, 5S, 6, 6 plus, 6S, 6S plus a iPad mini, mini 2, mini 3, mini 4, 2. generácie, 3. generácia, 4. generácia, Air, Air 2 a Pro. Podporuje teda obrovské množstvo zariadení. Každé z týchto zariadení má rozdielny výkon CPU, pričom najstaršie zariadenie ako napríklad iPhone 4S má procesor niekoľko násobne slabší oproti iPhone 6S plus. Tento výkon CPU sa odrazí na výkone aplikácie pri prehrávaní epizód spolu so sťahovaním Podcastov, kde je v režii najviac delegačných metód a súčasného spracovávaní rôznych procesov.

Ako testovací model som využil vlastný iPhone 5S, ktorý je už približne 3 roky starý. Neponúka najvyšší výkon CPU, a tak bolo možné reálne odhadnúť to, či aplikácia vie pracovať dostatočne dobre pri takomto výkone CPU, a či bude výkon postačujúci. V kapitole 3.1.1 som popisoval prostredie Xcode a aj spôsoby testovania a ladenia aplikácií. Aplikácia sama o sebe nieje náročná na chod a výkon CPU. Najvyšší výkon CPU je potrebný počas prehrávania epizódy a ešte vyšší keď aplikácia naraz sťahuje a prehráva epizódu. Pomocou módu ladenia v aplikácii Xcode som videl, koľko % procesora aplikácia využíva a taktiež aký to má dopad na celkovú spotrebu mobilného telefónu a využitie pamäte telefónu.

Podľa grafov využitia CPU, pamäte a spotreby energie môžem zhodnotiť, že aplikácia nemá žiadne veľké hardwarové nároky a bez problémov zvládne svoj chod aj na iPhone 4S. Pri sťahovaní a prehrávaní epizódy je využitie procesora na hranici v rozmedzí 80 % - 110 %¹. Navyše pri samotnom prehrávaní alebo sťahovaní aplikácie to bolo len niečo málo cez 80 %.

¹Režim TurboBoost umožňuje zvýšenie taktu procesora nad hranicu 100 %.



Obr. 22: Využitie CPU, pamäte a spotreba energie pri súčasnom prehrávaní a sťahovaní epizód

6.2 Testovanie na používateľoch

Pred uverejnením prvej verzie aplikácie Audeliver (neskôr premenovaná na Singlecast) som oslovil niekoľko ľudí aby moju aplikáciu skúsilo a povedali mi k nej nejakú kritiku alebo pochvalu. Webový nástroj iTunes connect, ktorý slúži na správu svojich aplikácií, umožňuje aj pozvať k aplikácii interných a externých testerov. Pre moju aplikáciu som využil interných testerov, čo boli moji známi, kolegovia z práce a spolužiaci, ktorým som odoslal pozvánku na testovanie aplikácie.

Používateľské dojmy boli pozitívne avšak vytkli mi pár vecí, ktoré by som mal opraviť alebo zjednodušiť. V prvej verzii aplikácie som po prvom spustení aplikácie neukazoval „navigačnú“ šípku, ktorá smerovala do pravého horného rohu aplikácie. Táto šípka bola navrhnutá používateľom, ktorému chýbala a mala úlohu určovať, čo treba robiť po prvom spustení aplikácie. Mala popis „Select a Podcast“, teda aby používateľ prešiel do okna určenom pre výpis Podcastov a zvolil si aspoň jeden. V prípade, že používateľ nemal k dispozícii žiadny Podcast pridaný, v aplikácii sa znova zobrazila šípka smerujúca do pravého horného rohu s textom „Add a Podcast“ a našepkávala používateľovi pridať Podcast pomocou URL adresy. Toto smerovanie pomocou šípok je nápomocné po prvom spustení aplikácie, aby sa v nej používateľ rýchlejšie zorientoval.

Ďalším vylepšením, ktoré bolo treba spraviť, bolo to, aby korektne fungovali ukazatele v hlavnom okne aplikácie, ktoré ukazovali buď % stiahnutej epizódy alebo % vypočítanej epizódy. V prvej verzii aplikácie sa to zobrazilo vždy iba prvýkrát pri zapnutí a ďalej sa tento výpis neobnovoval. Bolo potrebné vyriešiť problém vždy aktuálneho zobrazovania pomocou delegátnych metód.

Niektoré funkcie nefungovali správne, ako napríklad vyhľadávanie aj v popise epizódy alebo prehrávač v prípade, keď bolo použité vyhľadávanie. Všetky tieto úpravy zabrali minimum času a minimum zásahu do kódu. Samozrejme bolo potrebné aj optimalizovať dizajn aplikácie, ktorý nefungoval správne na každom zariadení, čo sa ukázalo pri samotnom testovaní aplikácie na používateľoch.

Takéto testovanie aplikácie je veľmi pohodlné a praktické v prípade, že mám dostatok testerov či už interných alebo externých. Xcode umožňuje aj výpis chybových kódov aplikácie alebo častí, kedy sa aplikácia používateľovi vypla alebo nefungovala správne. V prípade problémov môže tester aj reportovať chybu aplikácie a podobne. Uľahčí to tak celkovú prácu s testovaním veľkých aplikácií, ktoré majú fungovať spoľahlivo.

6.3 Zverejnenie aplikácie na AppStore

Ako som už spomínal, prvá verzia aplikácie sa volala Audeliver a niesla tak meno služby poskytujúcej vytvorenie vlastných Podcastov. Po prvom testovaní som chcel aplikáciu Audeliver uverejniť na AppStore, avšak narazil som na veľký problém. Pri overovaní aplikácie v App Review mi aplikáciu zamietli s niekoľkými poznámkami k jednotlivým problémom aplikácie. Prvý problém bol ten, že aplikácia obsahovala ikonu aplikácie podobnej aplikácii Podcasty od spoločnosti Apple, viď obrázok 23. Tento problém nebol taký závažný a dal sa jednoducho vyriešiť. Hlavný problém z App Review bol ten, že Apple označil službu audeliver.com za nelegálnu a podporujúcu tzv. „pirátsky obsah“ a aplikáciu, ktorá povolovala sťahovanie hudby od rôznych umelcov, ktorá neboli vlastníctvom Podcastu a samotnej služby audeliver.com. Taktiež jeden z problémov bol ten, že služba Audeliver musí mať zmluvu na poskytovanie hudobných nahrávok a jej šírenie aspoň s jednou nahrávacou spoločnosťou. Po dlhých rozhovoroch s centrálou Apple v Californii sa mi tento problém

nepodarilo nijako vyriešiť.

Po premýšľaní o riešení tohto problému som dospel k záveru, že by možno pomohlo to, aby sa aplikácia uverejnila pod univerzálnym menom a vôbec nespadala pod službu Audeliver.com. Rozhodol som sa aplikáciu prerobiť na univerzálne riešenie prehrávania Podcastov a tiež zmenil meno aplikácie na Singlecast. Po zmene mena aplikácie bolo potrebné vyriešiť aj zmenu ikony aplikácie, ktorá sa Applu nepáčila a tiež prispôbiť stahovanie a ukladanie XML kódu na univerzálny spôsob pre viac druhov Podcastov. Bohužiaľ, aj po týchto úpravách sa mi nepodarilo aplikáciu dostať do AppStore, pretože je potrebné vlastniť zmluvu s nahrávacou spoločnosťou. Položil som však otázku v Apple Review ako to robia ostatné aplikácie určené na prehrávanie hudby alebo Podcastov, no nedostal som žiadnu konkrétnu odpoveď.



Obr. 23: Ikona aplikácie Audeliver zamietnutá v App Review a ikona aplikácie Singlecast po zmene názvu aplikácie

Kapitola 7

Záver

Hlavným úspechom bakalárskej práce je vytvorenie fungujúcej aplikácie na príjem Podcastov, ktorá má jednoduchý dizajn a prehľadné rozhranie. Podarilo sa tak splniť zadanie bakalárskej práce, kde bolo potrebné vytvoriť funkčnú aplikáciu. Som rád, že som sa rozhodol pre vývoj na operačný systém iOS a taktiež pre jazyk Swift, ktorý je však ešte stále vo vývoji. Bohužiaľ, nepodarilo sa mi aj napriek veľkej snahe odoslať aplikáciu do AppStore, keďže bola zamietnutá v AppReview. Myslím si, že by bolo dobré časom tento problém vyriešiť, aby sa z tohoto projektu stala práca, ktorú bude možné priamo vyskúšať na vysokom počte zariadení.

Rád by som v budúcnosti zapracoval na nastaveniach aplikácie, ktoré by umožňovali nastaviť si príjem dát a podcastov tak, ako to používateľovi vyhovuje. Rozšíriť aplikáciu pre podporu Apple Watch, ktorá by podporovala prehrávanie Podcastov offline, priamo z hodínok. Zo začiatku som premýšľal nad možnosťou rozšírenia na Apple TV alebo systém OS X, no ukázalo sa, že vytvoriť jednoduchý a funkčný prehrávač Podcastov nie je až také jednoduché a vyžaduje si to množstvo detailov, na ktoré sa pri samotnom návrhu nemyslí.

Som presvedčený, že som zadanie bakalárskej práce splnil aj napriek tomu, že aplikácia neobstála pri schvaľovaní do Apple Store. Singlecast sa dá považovať za plnohodnotnú a funkčnú aplikáciu, ktorá by sa mohla využívať v zariadeniach iPhone, iPad a iPod.

Literatúra

- [1] Apple Inc. : App Distribution Guide. 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html>>
- [2] Apple Inc. : iOS Human Interface Guidelines. 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>>
- [3] Apple Inc. : iTunes Connect Developer Guide. 2016, [Online; navštívené 1.5.2016]. Dostupné z: <https://developer.apple.com/library/ios/documentation/LanguagesUtilities/Conceptual/iTunesConnect_Guide/Chapters/About.html>
- [4] Apple Inc. : NSPredicate Class Reference. 2016, [Online; navštívené 1.5.2016]. Dostupné z: <https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSPredicate_Class/>
- [5] Apple Inc. : Swift. 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://swift.org/>>
- [6] Apple Inc. : View Controller Programming Guide for iOS. 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/>>
- [7] BEUBURG, M. *Programming iOS 9*. 1. O'Reilly Media, 2015. ISBN 978-1-49193-685-6.
- [8] DEDEOGLU, K. : KDCircularProgress: A circular progress view with gradients written in Swift. GitHub, 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://github.com/kaandedeoglu/KDCircularProgress>>
- [9] FREED, N.; BORENSTEIN, N. S. : Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC 2046, RFC Editor, November 1996. Dostupné z: <<http://www.rfc-editor.org/rfc/rfc2046.txt>>
- [10] FULLER, K.; HARDWOOD, K.; NOON, C. : Alamofire: Elegant HTTP Networking in Swift. GitHub, 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://github.com/Alamofire/Alamofire>>
- [11] JALEEL, M. : TLYShyNavBar. GitHub, 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://github.com/telly/TLYShyNavBar>>
- [12] LEE, G. *Test-Driven iOS Development*. 1. Addison-Wesley, 2012. ISBN 978-0-3217-7418-7.

- [13] MOHUNDRO, D. : SWXMLHash: Simple XML parsing in Swift. GitHub, 2016, [Online; navštívené 1.5.2016]. Dostupné z: <<https://github.com/drmohundro/SWXMLHash>>
- [14] MORSON, S. *Designing for iOS with Sketch*. 1. Apress, 2015. ISBN 978-1-484214-59-6.
- [15] NAHAVANDIPOOR, V. *iOS 9 Swift Programming Cookbook*. 1. O'Reilly Media, 2015. ISBN 978-1-4919-3669-6.
- [16] SADUN, E. *iOS Auto Layout Demystified*. 2. Addison-Wesley, 2013. ISBN 978-0-321-96719-0.

Prílohy

Zoznam príloh

Obsah CD

- Projekt aplikácie Singlecast pre program Xcode
- Plagát bakalárskej práce
- Videouputávka pre aplikáciu Singlecast