

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

ZPĚVNÍK PRO MOBILNÍ ZAŘÍZENÍ S POKROČILÝMI FUNKCEMI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ KOMÁREK

BRNO 2015



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

ZPĚVNÍK PRO MOBILNÍ ZAŘÍZENÍ S POKROČILÝMI FUNKCEMI

ADVANCED SONGBOOK FOR MOBILE DEVICES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ KOMÁREK

VEDOUcí PRÁCE

SUPERVISOR

Doc. RNDr PAVEL SMRŽ, Ph.D.

BRNO 2015

Abstrakt

Tato práce se zabývá návrhem, implementací, vydáním a propagací aplikace zpěvníku pro operační systém Android. Největší důraz je kladen na schopnost aplikace přizpůsobit texty písní tak, aby byly velmi dobře čitelné i na malých obrazovkách mobilních zařízení. Aplikace je také doplněna dalšími funkcemi užitečnými zejména pro hráče na kytaru. Mezi ně patří například možnost stahování textů písní s akordy z webu, nebo automatický posun textu s nastavitelnou rychlostí pro různé části písně. Práce se dále zabývá ostatními důležitými aspekty vývoje aplikace, jako je její uživatelské testování a reálné nasazení.

Abstract

This Thesis describes all way from design and implementation to release and advertising of songbook application for Android operation system. The most important goal of the app is the ability to modify lyrics in such way, that they can be viewed flawlessly even on small screens of mobile devices. Application also features many other functions mostly useful for guitar players. User can download lyrics of song with chords directly from app and set up automatic scrolling with different speed for each part of the song. Thesis also describes other important aspects of application development, like user testing and final app release.

Klíčová slova

Android, Java, Android wear, materiální design, analýza textu, zpěvník, mobilní aplikace.

Keywords

Android, Java, Android wear, material design, text analysis, songbook, mobile application.

Citace

Ondřej Komárek: Zpěvník pro mobilní zařízení s pokročilými funkcemi, bakalářská práce, Brno, FIT VUT v Brně, 2015

Zpěvník pro mobilní zařízení s pokročilými funkcemi

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. RNDr. Pavla Smrže, Ph.D.

.....
Ondřej Komárek
17. května 2015

Poděkování

Děkuji Doc. RNDr. Pavlu Smržovi, Ph.D. za odborné vedení a za pomoc při tvorbě této práce. Dále bych chtěl poděkovat všem, kteří se podíleli na testování aplikace a podávali mi zpětnou vazbu.

© Ondřej Komárek, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
1.1 Motivace	3
1.2 Cíl práce	3
2 Analýza problému	4
2.1 Podobná stávající řešení	4
2.2 Zdroje dat	9
3 Použité technologie	10
3.1 Android	10
3.2 SQLite	13
3.3 Zpracování HTML	13
3.4 XML	13
4 Návrh řešení	14
4.1 Využití aplikace	14
4.2 Požadovaná funkcionalita	15
4.3 Design aplikace	18
4.4 Návrh uživatelských obrazovek	19
4.5 Databáze	22
4.6 Podporovaná zařízení	22
5 Implementace	24
5.1 Datové struktury	24
5.2 Aktivity, služby	25
5.3 Android wear	27
5.4 Pokročilé funkce	29
5.5 Další zajímavé funkce	35
5.6 Použité knihovny	36
6 Testování aplikace a zpětná vazba uživatelů	38
6.1 Úspěšnost analýzy a úpravy textu	38
6.2 Výkonnost	39
6.3 Uživatelské testování	40
7 Vydání aplikace a propagace	42
7.1 Uvedení do Google Play Store	42
7.2 Propagace	43

8 Závěr	45
8.1 Zhodnocení práce	45
8.2 Další rozvoj aplikace	45
A Obsah CD	48
B Plakát	49

Kapitola 1

Úvod

1.1 Motivace

Mobilní zařízení jsou pro mnoho uživatelů nejdůležitějším a často také jediným oknem do online světa. Využívají se k mnoha různorodým činnostem a počet aplikací pro mobilní platformy se vytrvale zvyšuje. Papírové knihy se prodávají stále méně a naopak prodeje jejich elektronických variant rostou. Zpěvníky patří mezi typy knih, u kterých se jeví jejich převedení do elektronické formy jako velmi vhodné. Uživatel takto může získat značné výhody jakými jsou zejména rychlé vyhledávání, označování oblíbených písní, nebo získávání nového obsahu. Mobilní zařízení má také člověk většinou u sebe, takže může elektronický zpěvník použít kdykoliv.

Hraji na hudební nástroj a několik mých přátel také. Přes všechny nesporné výhody elektronických zpěvníků používáme stále spíše ty papírové. Za touto volbou stojí zejména špatná čitelnost a nesprávné formátování na malých displejích u stávajících aplikací. V této práci se pokusím podobné problémy rozpoznat, analyzovat a na základě poznatků vytvořím aplikaci pro operační systém Android.

1.2 Cíl práce

Cílem práce je vytvořit dobře použitelnou aplikaci pro systém Android, která nebude opakovat chyby stávajících řešení a přinese nové funkce, které značně vylepší celkový dojem z používání. Mezi hlavní požadavky na aplikaci patří zejména její velmi dobrá optimalizace pro různé velikosti displejů, analýza a modifikace textu s akordy pro lepší čitelnost na menších zařízeních a také pěkný a přehledný design. V následujícím textu se pokusím tyto požadavky zhodnotit a postupně se dopracovat až k hotovému řešení. Vytvořenou aplikaci potom umístím do Google Play Store a budu se zabývat i její propagací.

V kapitole 2 je popsána analýza problému a srovnání plánů a cílů se stávajícími řešeními. Kapitoly 3 a 4 se zabývají návrhem řešení a výčtem použitých technologií s jejich popisem. Kapitola 5 popisuje implementaci aplikace, včetně změn v návrhu během řešení. Kapitoly 6 a 7 popisují ladění aplikace, zpětnou vazbu uživatelů, její vydání do Google Play Store a následnou propagaci.

Kapitola 2

Analýza problému

V této kapitole provedu zhodnocení stávajících aplikací a možnosti získávání textů písní. V následujícím textu je diskutován zejména význam jednotlivých funkcí.

2.1 Podobná stávající řešení

V současné době můžeme nalézt velké množství aplikací různé kvality s funkcí zpěvníku. Pokusím se obsáhnout všechny dostupné relevantní typy zpěvníků a zhodnotit jejich přínosy i negativa.

2.1.1 Klasický papírový zpěvník

Je nutné prohlédnout si běžné fyzické zpěvníky pro zjištění, co bude možné elektronickou verzí vylepšit a jaké jejich aspekty bude problém do aplikace promítnout. Pro tento účel jsem si vybral zpěvník Já písnička 1 [2].

Prvních několik stran zpěvníku zabírá obsah, který je rozdělený do kategorií podle žánrů a jednotlivé písně v nich jsou abecedně seřazeny. Ihned po obsahu následují samotné písně. U písní je proveden notový zápis společně s zobrazeným držením akordů na kytaru. Na začátku písně je uvedena její rytmika. Takto je nejčastěji označena pouze první sloka, zbytek textu je uveden bez značení a not. Na konci samotného zpěvníku je přehled akordů a rytmů, které jistě ocení mnoho začínajících hráčů. V aplikaci nejspíše nebude možné zobrazovat plný notový zápis. Nevýhodou klasického zpěvníku je jeho neskladnost a fakt, že jej příliš často nenosíme u sebe.

TISÍC MIL (FIVE HUNDRED MILES)
 Hudba a text H. West, český text Ivo Fischer (Doprovod: 6)

1. V nohách mám už ti-síc mil, stopy děm a ví-tr
 smyl a můj kůň í já jsme cestou zna-ve - ni.

R1. Těch ti-síc mil, těch ti-síc mil má jeden směr a jeden
 cíl, bí-lej dům, to ma-ly bí - ly stave - ní.

2. Je tam stráž a příkrej sráz, modrý tůň a bobří kráz,
 táta s mámnou, který věřj dětskejím snám.

R2. Těch ti-síc mil, těch ti-síc mil, má jeden směr a jeden cíl,
 jeden cíl, ten starcj, známěj, bílěj dům.

3. V nohách mám už ti-síc mil, teď mi zbyvá jen pár chvíl,
 cestu znám a ta se tam k nám nemění.
 R1.

4. Koušek dál a já to vím, uvidím už sto-pat dým,
 šikmej štít nad střechy ční k nebesám.
 R2.

(a) Stránka s textem písně

PŘEHLED AKORDŮ

	DUR (C)	MAJ I (Cmaj7)	DUR 7 (C7)	DUR 9 (C9)	ZVĚTŠENÝ (C+)	MOLL (Cm)	MOLL 7 (Cm7)	ZMĚNĚNÝ (Cdim)
C								
Cis Dim								
D								
Dis Em								
E								
F								
Fis Dim								
G								
Gis A								
A								
Ais B								
B								
Bis H								

(b) Přehled akordů

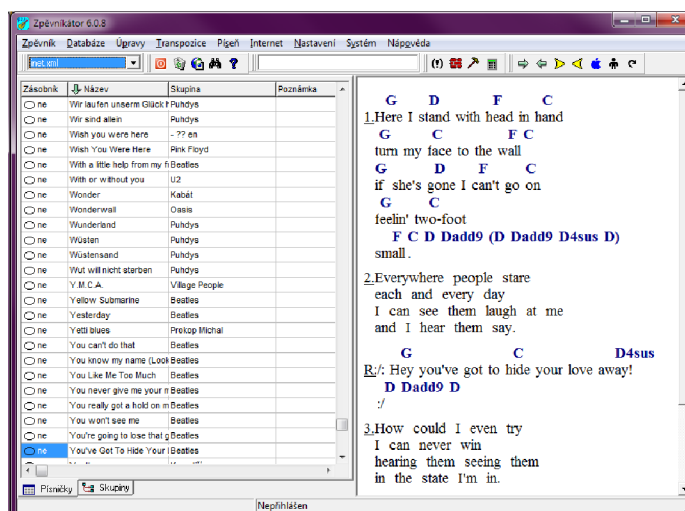
Obrázek 2.1: Elektronická verze papírového zpěvníku

2.1.2 Desktopové a webové aplikace

Tento typ aplikací sice nepůsobí jako přímá konkurence, je však vhodné prozkoumat jejich možnosti a funkce. Zvažím, které z funkcí jsou relevantní i pro mobilní aplikaci.

Zpěvníkátor

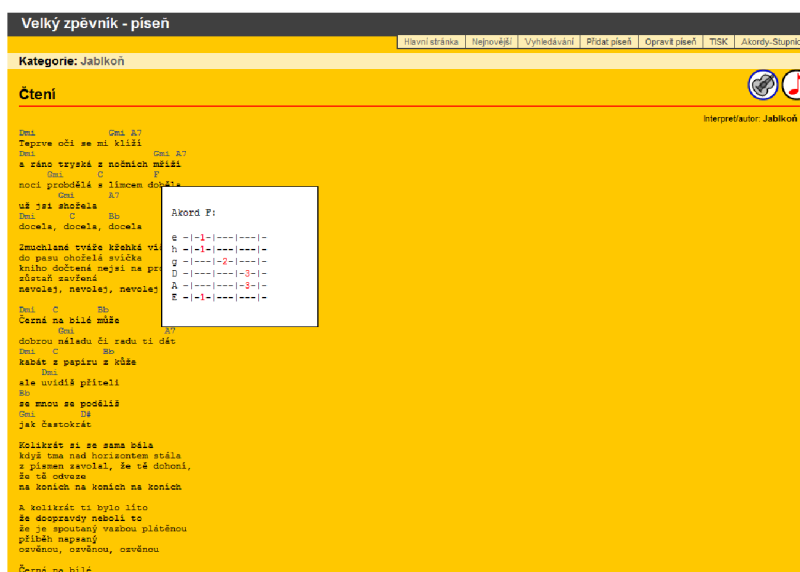
Desktopová aplikace s mnoha funkcemi a možnostmi nastavení zobrazení. Neobsahuje žádný základní seznam písní, do programu je tedy nutné dohledat databázi textů. Jako drobný nedostatek vnímám nemožnost vyhledání a stažení pouze konkrétní písně z některého serveru. Program obsahuje pokročilé možnosti transpozice akordů, včetně nalezení jejich podobného přepisu, který bude snadnější pro zahrání méně zkušeným hráčem. Zajímavou funkcí je možnost tisku zvolené písně. Program v levé části obrazovky zobrazuje seznam písní a v pravé vybraný text.



Obrázek 2.2: Ukázka programu Zpěvníkátor

velkyzpevnik.cz

Jednoduchá webová stránka s objemnou databází převážně českých textů. Služba umožňuje vyhledávání i prohlížení pomocí kategorií. Jako obrovské plus vnímám drobnou ikonku v seznamu písní znázorňující, zda text obsahuje akordy, nebo nikoliv. Tato zdánlivě nevýznamná funkce může výrazně zrychlit vyhledání vhodného textu pro konkrétní potřebu. Zobrazení textu je pěkné, pokud není limitováno malým displejem. Stránka bohužel nenabízí variantu webu pro mobilní zařízení. Prostor obrazovky je využit docela neefektivně, smysl by například dávalo zobrazit text ve více sloupcích. Zajímavou funkcí je zobrazení držení akordu na krku kytary nebo jeho zahrání na klávesách. Uživatel si může dokonce přehrát zvuk vybraného akordu.



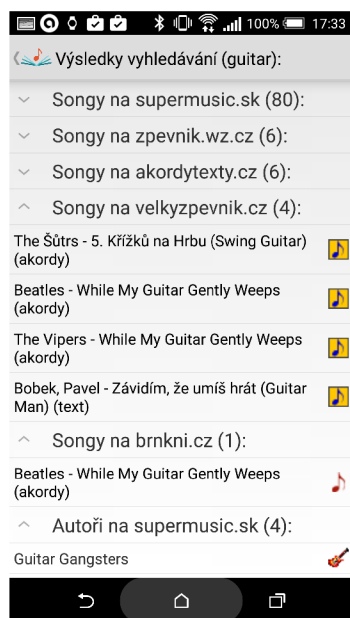
2.1.3 Mobilní aplikace

Nyní se pokusím zhodnotit provedení existujících mobilních aplikací. Vybral jsem pouze aplikace, které jsou zdarma, jsou obecnými zpěvníky a přijatelně fungují na zařízení s Androidem verze 5.0.1. Zvolil jsem aplikace na předních místech vyhledávání v Google Play Store podle klíčových slov “zpěvník” a “songbook”. Vybíral jsem takové aplikace, které mají nejvíce stažení a jsou nejlépe hodnoceny, takže se dají považovat za největší konkurenci.

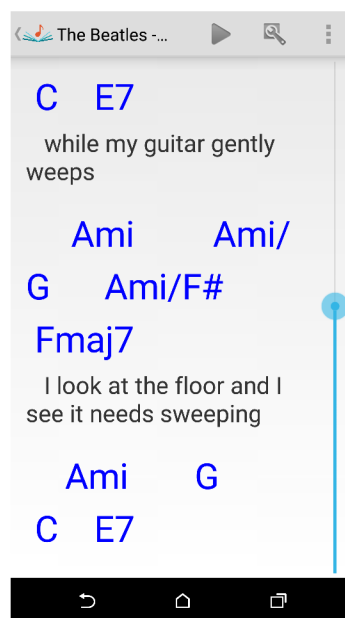
Zpěvník

Aplikace se zobrazí jako první ve vyhledávání v aplikačním obchodě po zadání výrazu “zpěvník”. U aplikace je zobrazeno přes 5000 stažení a má hodnocení 4,5/5, což jsou velmi dobré výsledky vzhledem k zaměření na české uživatele. Aplikace po spuštění zobrazí možnost vyhledání a přidání písně, takže nový uživatel nebude muset tuto často využívanou funkci hledat. Navigace v aplikaci je jednoduchá, logická, dojem však kazí velmi starý a strohý design. Zpěvník po spuštění neobsahuje žádné písně, takže si uživatel nemůže rychle prohlédnout základní funkce. V nastavení je k nalezení dohromady 7 zdrojů, ze kterých je možné texty stahovat. Stažené písně jsou potom označeny ikonkou zdroje, ze kterého pochází. Velmi pozitivně vnímám možnost zakázat některé servery, nebo jejich výsledky po

vyhledávání minimalizovat. Není však bohužel možné vkládat vlastní texty písní. Zobrazení textu písně je docela přehledné, ale při větší velikosti písma a zalomení řádku se akordy zobrazují v několika řádcích nad sebou, což je matoucí. Pro každou jednotlivou píseň jsou dostupné relativně pokročilé možnosti nastavení zobrazení, to je ale specifické pouze pro konkrétní text, takže jej uživatel musí pokaždé zadávat znova.



(a) Vyhledávání písní ke stažení



(b) Zobrazený text s akordy

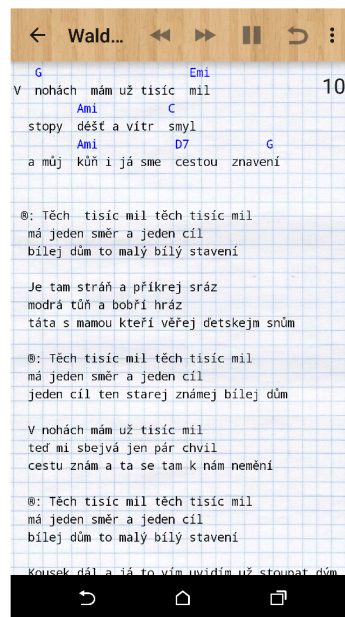
Obrázek 2.4: Ukázka aplikace Zpěvník, na obrázku (b) je zřetelné nevhodné zobrazení akordů na více řádcích nad sebou.

Guitar Songs

Tato aplikace se zobrazí jako šestá po vyhledání anglického výrazu “Songbook”. Předchozí aplikace jsou buď placené, nejsou obecnými zpěvníky nebo jsou tak nepovedené a plné chyb, že nemá smysl posuzovat jejich funkce. Aplikace má více než 500 000 stažení a hodnocení 4/5. Aplikace po spuštění vyzve k zvolení jazyků a následně se stáhne celá databáze textů dostupných pro vybrané jazyky. Reálnou použitelnost aplikace snižuje fakt, že si uživatel bohužel musí vystačit s tím, co aplikace sama stáhne. Pokud nějaká píseň chybí, má smůlu. Rozhraní aplikace je trochu matoucí, logice v navigaci nejvíce škodí takzvané “hamburger menu” (ikona třech čárek pod sebou, která po stisknutí vysune boční navigační panel). Uživatel má tendenci používat systémové tlačítko zpět a tím opouští aplikaci. Velmi pozitivně hodnotím možnost prohlédnutí držení akordů na kytáře a posuvník se zobrazením písmene abecedy. Dřevěné téma designu aplikace příliš nesvědčí a zřetelné jsou i nestandardní prvky používané spíše na platformě iOS. Není bohužel možné vkládat vlastní texty písní. Samotné zobrazení textu písně nemá žádné zvláštní funkce, kromě automatického plynulého posuvu. Akordy nad textem se nepřírozeně formátují po zvětšení velikosti písma a zalomení řádku.



(a) Seznam dostupných autorů



(b) Obrazovka textu písně s akordy

Obrázek 2.5: Ukázka aplikace Guitar Songs

2.1.4 Shrnutí

Ačkoli se aplikace svou povahou určením a funkcemi často velmi lišily, bylo možné rozpoznat klíčové funkce i nedostatky.

Za zajímavé vlastnosti bych označil:

- Možnost prohlédnutí držení akordů na krku kytary
- Transpozice akordů
- Zpřehlednění zobrazení podle ikonky zdroje
- Rychlý posun v seznamu písní podle pořadí v abecedě
- Automatické posouvání textu písně

Jako nejčastější nedostatky jsem identifikoval:

- Zastaralý design
- Chaotická a nestandardní logika ovládání
- Nepraktické zobrazení akordů nad texty při zvětšení velikosti písma (viz obrázek 2.4)
- Nemožnost hrát celou píseň podle akordů, pokud jsou uvedeny jen u první sloky písně
- Nízká výkonnost v případě většího množství textů
- Automatický posun textu je často příliš rychlý, nebo naopak pomalý a požadovaný text tak není vidět

V aplikacích mi nejčastěji chybělo:

- Stahování nových textů přímo z aplikace
- Filtrování, řazení
- Pokročilé nastavení
- Vkládání vlastních písní
- Vyhledávání bez diakritiky (pokud bylo obsaženo)

Těmito zjištěními se budu dále zabývat v kapitole 4 – Návrh řešení, kde navrhnu koncept aplikace, která využije přednosti současných řešení a zároveň odstraní jejich nedostatky.

2.2 Zdroje dat

Pro plánovanou aplikaci je nutné vhodně zvolit spolehlivé zdroje písňových textů, protože aplikace bude distribuována jen s malým počtem základních písní. Hlavní inspirací pro výběr serverů byla aplikace Zpěvník, která jich podporuje mnoho. Bohužel žádný z uvedených serverů neposkytuje žádné API na stahování textů, takže bylo nutné hledat jiný postup. Další možný způsob získání dat bylo načtení HTML kódu stránky a vyparsování potřebných informací z těchto dat. Tato metoda je bohužel závislá na změně kódu stránek. Přesto jsem si tuto variantu nakonec zvolil, protože se zdála jako jediná reálně možná.

Vybral jsem si dva české servery: `velkyzpevnik.cz` a `pisnicky-akordy.cz`. Oba obsahují značnou databázi záznamů a jejich HTML kód je pro stránku se seznamem vyhledaných písní i pro zobrazení detailu písně navzájem podobný, takže nebude třeba mít parser příliš odlišný. Oba dva také podporují vyhledání výrazů, které byly zadány bez diakritiky.

Rozhodl jsem se také pro podporu nějakého zahraničního serveru, protože bych aplikaci rád publikoval celosvětově. Oba zmíněné české servery sice obsahují určité množství textů zahraničních kapel, ale zejména moderní písně tam často chybí. Jako vhodný zástupce zahraničního serveru se jevil `ultimate-guitar.com`, který má navíc velmi aktivní komunitu.

Kapitola 3

Použité technologie

Po analýze problému je o něco jasnější, jaké technologie budu pro vypracování projektu potřebovat. V této kapitole popíšu jejich funkci, plánované použití a vysvětlím související pojmy.

3.1 Android

Android je v současné době nejrozšířenější mobilní operační systém. V roce 2014 měl na trhu podíl 81.5% mezi aktivními chytrými telefony [4]. Je tedy platformou, na které má aplikace potenciál velkého množství stažení, ale musí se i potýkat se značnou konkurencí. Následující text volně vychází z online dokumentace Googlu a knihy Pro Android 4 [3].

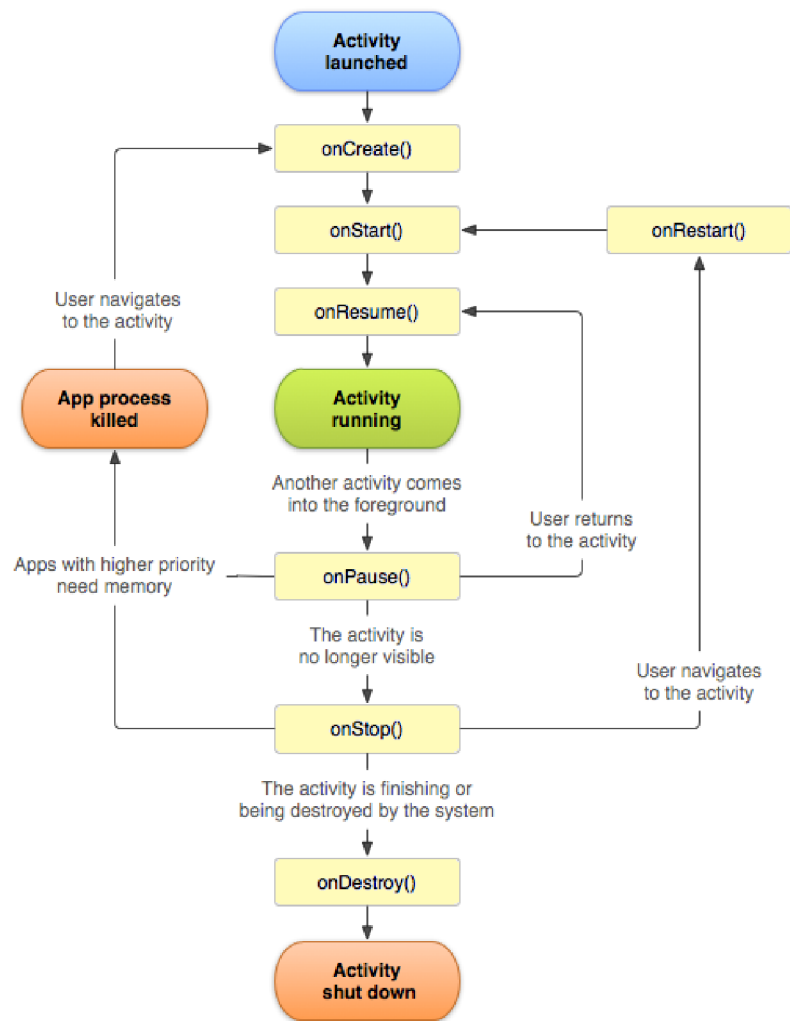
3.1.1 Prvky aplikace

Zde si představíme základní komponenty aplikace a také si definujeme české ekvivalenty. Názvy komponent a prvků se v praxi téměř nepřekládají, takže je možné, že uvedené názvy nebudou v českém kontextu zcela přesné.

Aktivity

Přeloženo z angl. Activity. Aktivity jsou prvek, který slouží k zprostředkování zobrazení, se kterým může uživatel interagovat. Data a programová logika vázaná k vykreslování obsahu jsou obsažena právě v aktivitě. Obvykle aktivita odpovídá jedné určité obrazovce v aplikaci. Uživatelské rozhraní může být tvořeno programově v aktivitě, běžně se spíše používá rozvržení prvků definované v souboru XML. Aktivity si mohou navzájem vyměňovat data různými způsoby. Mohou například používat databázi, zapisovat si navzájem hodnoty do statických členů tříd, nebo si posílat data při spouštění nebo ukončení aktivity pomocí záměrů. Na obrázku 3.1 je zobrazen životní cyklus aktivity¹.

¹Převzato z <http://developer.android.com/guide/components/activities.html>



Obrázek 3.1: Životní cyklus aktivity

Záměry

Přeloženo z angl. Intents. Záměry jsou objekty pro předávání dat mezi aktivitami, potažmo i různými aplikacemi. Potřebná data jsou k záměru přidána v serializované podobě. Existují dva základní typy záměrů – explicitní a implicitní. Explicitní určují přímo cílovou komponentu pro doručení zprávy a v praxi se využívají na spuštění jiné aktivity. Implicitní typy pouze zadávají obecnou akci k vykonání. Tato akce je zpracována operačním systémem, který v závislosti na nastavení zobrazí možné varianty vhodné pro zpracování požadavku, nebo vybere zvolenou standardní. V praxi může jít například o požadavek načtení webové stránky a dotaz k uživateli ze strany systému na výběr webového prohlížeče.

Služby

Přeloženo z angl. Services. Služba může být vnímána podobně jako aktivita, má však určité odlišnosti. Nejpodstatnějším rozdílem je to, že přímo neposkytuje uživatelské rozhraní. Dále má jiné možnosti a omezení ze strany operačního systému. Je například možné, aby se služba sama spustila po startu zařízení. Systémový manažer paměti také zohledňuje služby

jako prioritní a umožňuje jejich běh po velmi dlouhou dobu a to i při nedostatku paměti. Služby se tak nejvíce hodí pro určité úkony, které musí probíhat na pozadí, i když uživatel s aplikací neinteraguje. V praxi se často využívají pro implementaci různých časovačů, nebo jako posluchače pro komunikaci mezi více zařízeními.

Rozložení prvků

Přeloženo z angl. Layouts. Rozložení prvků definuje strukturu uživatelského rozhraní. Tato struktura může být specifikovaná odlišně pro zařízení s různými parametry, jako je rozlišení a velikost displeje, typ zařízení a další. Programátor může definovat prvky uživatelského rozhraní pomocí souboru XML, který potom předá aktivitě při jejím spuštění. Za běhu aplikace je potom možné ovlivňovat zobrazené prvky programově. V praxi se většina rozhraní definuje prostřednictvím XML souboru a programové změny slouží jen pro drobné úpravy.

3.1.2 Verze systému, Android SDK a Android Studio

Systém Android je známý svou fragmentací verzí fungujících na aktivních zařízeních. Tento fakt velmi znesnadňuje vývoj moderních, ale stále kompatibilních aplikací. Nyní se již nikdy nepodporují prvotní a zastaralé verze systému, zejména kvůli jejich nízkému zastoupení a také značným omezením. Pokud vývojář použije pouze prvky a funkce dostupné i ve starších verzích systému, není problém se zpětnou kompatibilitou z hlediska spuštění kódu. Potíže však činí různá grafická podoba prvků uživatelského rozhraní, která je odlišná pro různé verze systému a je také ovlivněna grafickou nastavbou od výrobce telefonu. Pokud je použita nějaká funkce dostupná pouze v novějším systému, je nutné zvýšit jeho minimální podporovanou verzi, nebo s využitím programového větvení funkci implementovat jiným způsobem, který je dostupný i na starších zařízeních. Problémem pak je nekonzistence funkcionality napříč verzemi systému, horší údržba a čitelnost kódu, ztížení návrhu a častější chyby.

Android SDK je softwarový vývojářský balík pro vývoj aplikací. SDK obsahuje API příslušné verze systému, dokumentaci, virtuální obrazy různých zařízení a další nástroje pro vývoj. V tomto projektu počítám i s vývojem pro mladou platformu Android wear. Tento upravený systém Android je určený pro použití především v nositelných technologiích, v dnešní době jsou to hlavně chytré hodinky.

V nynější době se nejčastěji používá vývojové prostředí Android Studio [5] postavené na IntelliJ IDEA od JetBrains, které je modernější alternativou k vývojářskému prostředí Eclipse. Nativním jazykem pro tvorbu Android aplikací je Java.

3.1.3 Distribuce aplikací

K distribuci aplikací je možné použít dva hlavní kanály. Používanější z nich je oficiální distribuce přes Google Play Store. Tuto základní variantu mohou využít jen zařízení používající služby Google Play, kterých je však drtivá většina. Aplikace z tohoto kanálu jsou prověřené, takže mají určitou minimální úroveň kvality. Většina uživatelů instaluje aplikace tímto způsobem.

Dalším způsobem distribuce aplikace je šíření samotného souboru .apk vlastními kanály. Aplikace sice nemusí projít přísnými kritérii Googlu, ale uživatelé k její instalaci budou mít menší důvěru a pravděpodobně ji ani nenaleznou. Mým cílem je distribuovat vytvořenou aplikaci prostřednictvím oficiálních zdrojů Google Play Store.

3.2 SQLite

SQLite je open source relační databáze. Podporuje běžnou SQL syntaxi a umožňuje tvořit dotazy do databáze programově. Záchází pouze se třemi základními datovými typy – TEXT, INTEGER a REAL. Databázi budu využívat pro perzistentní uložení textů písní a dalších informací.

3.3 Zpracování HTML

HTML je značkovací jazyk pro vytváření webových stránek. HTML se skládá z množiny značek (tagů) a jejich atributů. Mezi značkami je pak obsažen samotný text. Pro získání potřebných dat z HTML dokumentu se používají parsery. V tomto projektu bude parsování HTML potřeba zejména k získání seznamu písní po vyhledávání na serveru a také ke stažení textu a akordů z vybraného detailu písně.

3.4 XML

XML je obecný značkovací jazyk určený zejména pro výměnu informací v serializované podobě. XML se podobně jako HTML skládá ze značek a jejich atributů. Pro XML existují výkonné nástroje ke čtení tohoto formátu a zápisu do něj. V tomto projektu budu XML využívat pro zápis rozvržení prvků uživatelského rozhraní a také jako formát pro export databáze písní.

Kapitola 4

Návrh řešení

Tato kapitola se zabývá návrhem řešení aplikace. Přebírá poznatky a zjištění z analýzy problému a rozvíjí je do plánování implementace.

4.1 Využití aplikace

Cílová aplikace by měla obsahovat takové funkce, aby byla vhodná pro různé typy použití. O předpokládaných funkcích má smysl uvažovat pouze, pokud budeme očekávat jejich smysluplnost v různých situacích. Představme si tedy několik pravděpodobných scénářů využití aplikace:

Uživatel uslyší z rádia píseň a bude ho zajímat její text:

- Bude očekávat zejména rychlost, jednoduchost a přesnost vyhledání
- Nezajímají ho akordy, spíše jej obtěžují
- Píseň si nejspíše nebude chtít uložit
- Akce bude tak rychlá, že by mělo smysl uvažovat i o aplikaci pro Android wear

Skupinka přátel si bude chtít večer zazpívat u táboráku:

- Písně budou nejspíše již staženy
- Uživatelé budou zejména vyhledávat ve stažených písních
- Uživatelé mohou chtít využít funkce importu a exportu pro sdílení databáze mezi zařízeními
- Často využívána bude nejspíše funkce označení písně jako oblíbené

Amatérský hráč chce vyhledat píseň s akordy, kterou by si zahrál:

- Uživatel bude chtít vyhledat píseň, zjistit, zda obsahuje akordy
- Hráč bude nejspíš často písně přepínat, vyhledávat, filtrovat a řadit
- Uživatel si bude chtít najít a poslechnout originální znění skladby
- Klíčové bude správné zobrazení akordů a jejich pohodlné čtení po celou dobu hraní písně
- Uživatel ocení funkci metronomu a automatického posuvu

4.2 Požadovaná funkcionalita

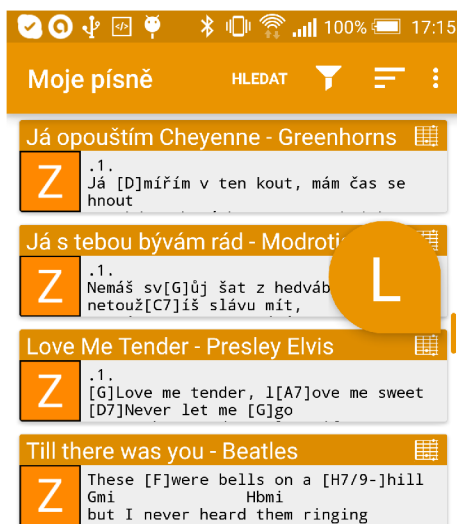
Podle příkladových využití z předchozího textu, analýzy celého problému a zadání práce bude třeba navrhnout specifické funkce zpěvníku. Zkusme si nastínit nejdůležitější z nich a vysvětlit si jejich smysl.

Základní databáze písní a možnost stahovat nové

Obě tyto části jsou důležité pro zaujetí uživatele při prvním spuštění a zvýšení motivace používat aplikaci i v budoucnu. Uživatel si bude chtít vyzkoušet funkce aplikace hned po spuštění, takže je dobré, aby obsahovala několik základních textů. Stahování dalších písní musí být jasné a přehledné. Tuto funkci bude uživatel pravděpodobně používat často, takže bude vhodné ji umístit přímo na obrazovku se seznamem písní pro rychlý přístup.

Filtrování, řazení a vyhledávání

Tyto funkce budou nutné zejména pro časté uživatele z důvodu rychlejšího nalezení písně. Vyhledávání bude podporovat nalezení textu s diakritikou i pokud bude zadán výraz bez ní. Řazení bude umožněno buď podle jména autora, nebo názvu písně. Filtrování bude vhodné zavést pro skrytí písní z různých zdrojů. Takto si uživatel bude moci snadno zobrazit třeba jen naimportované písně. Pro velmi velké množství položek je také dobré zavést funkci rychlého posuvníku s ukazatelem písmene abecedy – viz obrázek 4.1



Obrázek 4.1: Výřez obrazovky používající rychlý posuvník se zobrazením písmene abecedy

Zobrazení ikony zdroje

K rychlejší orientaci také poslouží ikona typu zdroje. Uživatel tak může i bez filtrování snadno rozpoznat, jaký text je ze kterého serveru, zda ho vložil manuálně, importoval a podobně. Tato ikona také bude sloužit pro označení zdrojového serveru při vyhledávání nových písní ke stažení.

Přidání do oblíbených

Uživatel by měl mít možnost označit text jako oblíbený. Takové texty by měly být graficky zvýrazněny a řazeny v seznamu položek nahore pro snadnější dostupnost. Tato funkce by měla ušetřit nutnost hledání často hraných písní a předpokládám její hojné využívání. Označit text jako oblíbený bude možné z detailu textu pomocí plovoucího akčního tlačítka.

Znázornění obsažení akordů

Položka textu písně v seznamu by měla zřetelně zobrazovat, zda text písně obsahuje akordy, nebo jde pouze o samotný text. Uživatel, který chce hrát na kytaru, vyhledává spíše jen ty s akordy. Kdo hledá jen texty písní, pravděpodobněji vybere text bez akordů.

Podpora Android Wear

Aplikace by měla mít možnost zobrazovat písně přímo na hodinkách s Android Wear. Toto může mít využití pro hráče na kytaru, kteří mohou text prohlížet přímo při držení krku kytary. Jde především o podporu funkce odeslání konkrétního textu do hodinek po akci vyvolané uživatelem. Jako zajímavá možnost se jeví i nalezení textu písně přímo z hodinek hlasovým povelům. Toto by využili lidé pro rychlé vyhledání textu písně, kterou právě slyší. Tato funkcionalita je však velmi obtížná na implementaci a vzhledem k tomu, že aplikace je určena především pro hráče na hudební nástroj, ji nejspíše přenechám až na příští verze aplikace.

Manuální časování posunu textu

Automatický posuv textu je sice zajímavou funkcí, ale ve většině případů nefunguje správně. Uživatel často nevybere vhodnou rychlost, nebo se některá část písně má hrát vícekrát a požadovaný text se tak ocitne mimo hranice obrazovky. Zpěvník by proto měl obsahovat možnost načasovat si posun textu podle přání hráče. Uživatel si například může pustit originální píseň a zvolit si režim časování posuvu. Potom v průběhu písně posouvá obrazovkou podle svých potřeb. Po nastavení časování může píseň hrát s textem posouvaným přesně podle jeho preferencí.

Jako vhodná velikost jednotky posunu se jevila polovina výšky obrazovky. Posun o větší část obrazovky by mohl vyvolat skrytí prohlížené části textu při nedokonalé synchronizaci a kratší posuny by zase musely probíhat častěji, takže by hráče více obtěžovaly. Posun bude prováděn rychlou animací pro lepší možnost udržení pohledu na jednom místě v textu. Časové známky posunů budou uloženy v databázi. Uživatel bude při změně velikosti textu upozorněn, že toto jiné nastavení poničí správnou synchronizaci posunu textu. Tato funkcionalita by měla být dostupná i na hodinkách Android wear, kde bude praktická zejména z důvodu jejich drobného displeje.

Metronom

Tato funkce bude pro spoustu hráčů příjemná, protože nebudou muset pro hru používat dvě aplikace současně. Metronom bude dostupný z obrazovky detailu textu a bude obsahovat jen základní nastavení.

Optimalizace pro malé obrazovky

Tato oblast je poměrně obsírná – zahrnuje všechny aspekty, které ovlivní finální zkušenost uživatele s hlavní částí aplikace – prohlížením textu písní. Jako největší problém existujících aplikací jsem vnímal nesprávné zalamování řádků textů s akordy. U většiny z nich se při větší velikosti textu delší řádky zalamovaly například na tři viditelné řádky. Akordy nad nimi bohužel udělaly to samé, takže uživatel viděl nepřehledné uskupení několika řádků s akordy následované třemi řádky textu. Tento jev je dobře vidět na obrázku 2.4 z aplikace Zpěvník. Správné řešení by mělo vždy zobrazovat jeden řádek akordů a jeden řádek textu, nehledě na zalamování. Tato drobná změna zajistí výrazně větší čitelnost textu a schopnost hráče správně přiřadit změnu akordu k místu v něm. Tento problém bude řešen vhodnou úpravou zobrazení textu.

Pokročilá analýza a úprava textů

Největší překážkou správného zobrazení textu na mobilních zařízeních je nevhodná forma vkládání akordů do textu přímo uživateli. Velká část textů obsahuje akordy třeba jen v první sloce a prvním refrénu. Na velké obrazovce počítače nebo v papírovém zpěvníku to není velký problém – hráč se může podívat na akordy v první sloce, protože je text na velké obrazovce viditelný celý. Podobný problém je ve zobrazení refrénů. Ty jsou často uvedeny jen jednou a v dalším textu se na ně odkazuje jen značkou “Ref:”, popřípadě v anglických textech značkou “Chorus:”. Hráč, který text písně příliš dobře nezná, nemůže reálně hrát a zpívat refrény, protože je zkrátka na menším displeji nevidí. Tím klesá celá využitelnost aplikace na mobilních zařízeních, protože se uživateli nechce manuálně všechny texty upravovat, aby byly dobře čitelné.

Je tedy na místě vytvořit nějaký systém, který bude tyto nedostatky rozpoznávat a automaticky opravovat. Největší překážkou v takovém snažení jsou však velmi rozdílné formáty, ve kterých uživatelé písně vkládají. Je tedy logické jeden z nejčastějších a nejprehlednějších formátů označit jako základní a pokoušet se všechny ostatní typy rozpoznat a převést právě do něj. Systém úpravy textů by se dal rozdělit do třech modulů:

- **Normalizace textu** – Texty, které mají nestandardní, nebo žádné značení, je potřeba upravit, aby byly vhodné pro další analýzu. Pokud nemá text žádné značení, je třeba pokusit se odhadnout hranice slok a rozlišit refrény. Na konci této fáze by všechny texty měly mít podobný formát a značení.
- **Analýza a úprava textu** – Tato část se zaměřuje na odstranění hlavních nedostatků, které způsobují špatnou čitelnost. Funkcí analýzy je tedy rozpoznat, zda má píseň akordy jen v první sloce a zda jsou refrény uvedeny jen jako odkaz. Pokud text tyto problémy obsahuje, provede se jejich úprava. Akordy jsou přepsány i do ostatních slok a značky refrénů jsou nahrazeny skutečným textem. Cílovým stavem je správně naformátovaný a označený text písně, který má akordy obsaženy ve všech slokách a všechny refrény jsou popsány celým jejich textem i s akordy.
- **Modifikace zobrazení textu** – Tento modul zajišťuje převod z interní reprezentace textů s akordy do reálné čitelné formy. Měl by zajišťovat korektní zobrazení a zvýraznění akordů nad textem. Dále by měl zařídit formátování textu tak, aby nedocházelo k vrstvení akordů nad sebe při zalomení řádku textu s akordy.

Pro tento modulární způsob zpracování jsem se rozhodl po zkoušení jiných přístupů, jakými bylo například vytváření specifických metod určených k modifikaci textu pro každý

typ vstupního formátu zvlášť. Těch však bylo příliš mnoho a navíc by se mnoho úprav v programovém kódu opakovalo. Proto jsem nakonec zvolil obecnější metodu, u které jsem předpokládal i větší úspěšnost a menší chybovost (nesprávné rozpoznání a poničení textu).

U této funkce je také důležité vnést do nastavení možnost jejího vypnutí, pokud by uživatele obtěžovala, nebo nebyla pro jeho případy užití příliš úspěšná. Každý z modulů by měl navržen tak, aby při nemožnosti splnit jeho funkci text nijak neupravil a neponičil. Uživatel by měl mít možnost upravit již modifikovaný text, takže je vhodné, aby se úprava provedla pouze jednou a poté se její výsledek uložil. Jediná část, která by se měla vykonávat pokaždé, je modifikace zobrazení textu, aby byl text správně naformátován i po změně velikosti písma.

Bohaté možnosti nastavení

Při vytváření nastavení aplikace je nutné dodržovat určitá specifická pravidla¹, aby se uživatel necítil zmaten a byl si přesně vědom, co vlastně nastavuje. Není například vhodné vkládat do obrazovky nastavení možnost, kterou bude potřebovat uživatel měnit poměrně často – taková položka by měla být umístěna do akční nabídky. Dále je dobré se vyhnout možnostem nastavení, které většina uživatelů pravděpodobně nikdy nezmění. Při popisu každé položky nastavení je nutné zvolit velmi jasný, přímočarý a jednoduchý jazyk, protože velká část uživatelů používajících aplikaci v anglickém jazyce nemusí být rodilým mluvčím.

Aplikace bude obsahovat nastavení:

- Volba velikosti písma, odděleně pro telefon a hodinky
- Možnost povolit, nebo zakázat automatické vypínání obrazovky při neaktivitě
- Výběr zobrazení akordů běžně, nebo tučně, barva jejich zvýraznění
- Možnost povolit, nebo zakázat analýzu a vylepšování textů
- Výběr serverů používaných pro stahování písní
- Filtr zobrazených stažených písní podle jejich zdroje, volba řazení písní
- Zobrazení nebo skrytí akcí pro Android wear
- Export databáze

4.3 Design aplikace

Jako základní styl designu jsem si zvolil moderní Materiální design [9]. Tento čistý a přehledný způsob zobrazování grafických prvků byl představen společně s verzí Androidu 5 s kódovým označením Lollipop. Design se vyznačuje výraznými barvami, čistými jednolitými plochami (flat design), stíny počítanými v reálném čase a propracovanými animacemi připomínající reálný pohyb. Aplikace navržená v tomto designovém stylu by měla působit dojmem přítomnosti reálných objektů. Například dialog pro interakci s uživatelem vypadá jako skutečný lístek s poznámkou, který vystupuje nad rozhraní aplikace a skutečně upoutá pozornost.

¹<http://developer.android.com/design/patterns/settings.html>

Design bude nutné v aplikaci využívat takovým způsobem, aby naplňoval přísné směrnice jeho správného použití. Jen tak zajistíme, že uživatel hned po prvním spuštění bude znát navigaci a rozumět významu prvků. Zároveň však bude nutné používat programové techniky [6] pro zajištění kompatibility se zařízeními fungujícími na starší verzi systému.

Z novinek tohoto krásného designového stylu využijí plovoucí akční tlačítko na přidávání nových textů písní a na označení písně jako oblíbené. K zobrazení seznamu textů použijí kartičkový seznam CardView. Jako hlavní barvu aplikace jsem zvolil tmavě oranžovou. Mým osobním cítěním se barva k povaze aplikace dobře hodí a navíc ji mají uživatelé spojenou s hudebním obsahem, protože je také použita v aplikaci Hudba Play od Googlu.

Pro oživení aplikace jsem se rozhodl vytvořit systém stahování obalů alb, které odpovídají zobrazené písni. Díky získanému obrázku bude možné vytvořit mnohem zajímavější zobrazení detailu písně.

4.4 Návrh uživatelských obrazovek

V této sekci si popíšeme rozložení obrazovek aplikace a předávání dat mezi nimi. Vysvětlení a popis zamýšleného návrhu obrazovek provedu na snímcích skutečně dokončené aplikace, protože takto bude celé schéma snáze pochopitelné. Na schématu jsou vyznačeny akce použití nejdůležitějších navigačních prvků a obrazovky zobrazené po jejich zvolení. Celý diagram z obrázku 4.2 je okomentován v dalším textu.

4.4.1 Základní obrazovka – Moje písně

Po spuštění aplikace se zobrazí obrazovka **Moje písně**. Zde uživatel nalezne seznam stažených textů, nástroje pro vyhledávání, filtrování a řazení. Je odtud možné vstoupit do nastavení, nebo rozbalit nabídku akčního tlačítka pro získání dalšího obsahu. V rozbalené nabídce je možné zvolit možnost **Importovat**, **Stáhnout z webu**, nebo **Vložit manuálně**. Uživatel může také provést akci smazání nebo úpravy nad položkou ze seznamu textů. Samozřejmě je také možné některou píseň vybrat klepnutím a zobrazit si tak její detail.

4.4.2 Import

Tato obrazovka má velmi prostou funkci. Uživatel je pouze vyzván, aby vybral soubor s daty pro import. Ihned po zvolení souboru dojde k importu a zobrazí se opět úvodní stránka **Moje písně**.

4.4.3 Stáhnout z webu

Na této obrazovce se uživatel ocitne po zvolení akčního tlačítka **Stáhnout z webu**. Zde může zadat vyhledávaný výraz a zobrazit si nalezené výsledky. Položky jsou odlišeny ikonou zdroje serveru, ze kterého daná píseň pochází, a jsou seřazeny podle relevance tak, jak určil server. Uživatel si zde vybere některý ze zobrazených textů a ocitne se v **detailu písně**.

4.4.4 Detail písně

Tato obrazovka je využita pro zobrazení textu již stažené, nebo právě vyhledávané písně. Obsahuje akční menu s několika funkcemi, stažený obal alba, akční plovoucí tlačítko na označení písně jako oblíbené a upravený text s akordy. Akční menu zobrazuje dostupné ikony funkcí, pouze pokud mají pro daný kontext smysl. Při prohlížení zatím nestažené

písně se například zobrazí ikona diskety pro uložení, nebo po vytvoření časování posunů se objeví ikona spouštějící automatický posun. Dále zde nalezneme tlačítko pro **odeslání** textu do hodinek, možnost automatického vyhledání hudebního videa na YouTube, ikonu **Nastavení časování** a možnost **Upravit**.

4.4.5 Úprava písně

Na tuto obrazovku může uživatel vstoupit více způsoby. Bude vyvolána prázdná, pokud na obrazovce **Moje písně** zvolí možnost **Vložit manuálně**. Ze stejného místa může být upravena existující píseň dlouhým podržením položky seznamu a zvolením **Upravit**. Poslední možností příchodu je volba **Upravit** z **detailu písně**.

4.4.6 Časování posuvů

Časování posuvů se zobrazí po stisknutí ikony přesýpacích hodin v **detailu písně**. Uživatel je upozorněn na zacházení s funkcí pomocí přechodného systémového upozornění (toast). Po prvním dotyku obrazovky se zobrazí časovač. Dalšími dotyky se obsah posune přibližně o polovinu své výšky směrem nahoru, takže se zobrazí další text. Pro ukončení a uložení časování stačí stisknout systémové tlačítko navigace zpět.

4.4.7 Nastavení

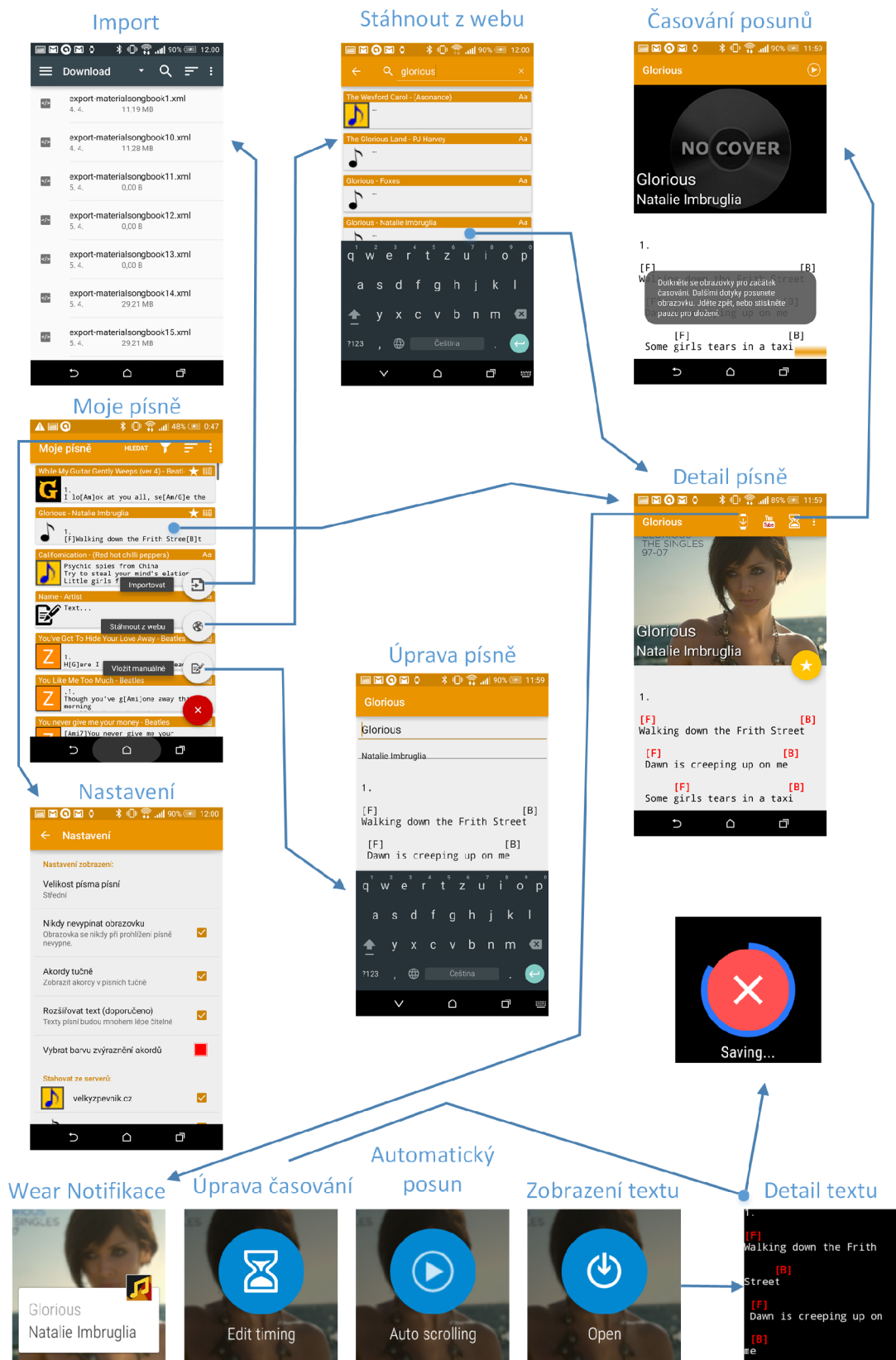
Do nastavení se uživatel dostane z obrazovky **Moje písně** volbou nastavení.

4.4.8 Android Wear – upozornění

Po stisknutí ikonky zobrazující odeslání do hodinek se na tomto zařízení zobrazí systémové upozornění (notifikace). Možnosti zobrazení prvků na hodinkách s Android wear jsou poměrně omezené, zejména kvůli malému displeji a směrnicím² uživatelského rozhraní od Googlu. Zvolil jsem systém zobrazení, kdy je upozornění ve stylu malé kartičky s ikonou aplikace.

Na kartičce je název písně se jménem autora a na pozadí se zobrazuje obrázek obalu alba. Při posunutí zprava doleva se odkrývají jednotlivé funkce, které jsou zobrazeny od kartičky napravo. Textové popisky funkcí jsou v anglickém jazyce, protože Android wear zatím nepodporuje český jazyk. První v pořadí je možnost **úprava časování**. Tato funkce je velmi podobná jako **časování posuvů** na telefonu – uživatele taktéž provede celým postupem, který je srovnatelný. Rozdíl je pouze v ukládání časování, kdy uživatel může proces zrušit, pokud akci na malinkém displeji vyvolal omylem. Další možností je **automatické posouvání**. Tato funkce je zobrazena pouze, pokud již bylo provedeno časování. Tato varianta zobrazení po prvním dotyku obrazovky provede posuny přesně v intervalech, jaké si uživatel zadal. Poslední volbou je možnost běžného **zobrazení** textu bez dalších speciálních funkcionalit.

²<https://developer.android.com/design/wear/patterns.html>



Obrázek 4.2: Návrh jednotlivých obrazovek, šipky znázorňují přepnutí zobrazení.

4.5 Databáze

SQLite databáze bude v aplikaci sloužit pro uložení textů písní a jejich načasování pro telefon i hodinky. Bude tak obsahovat celkem tři tabulky:

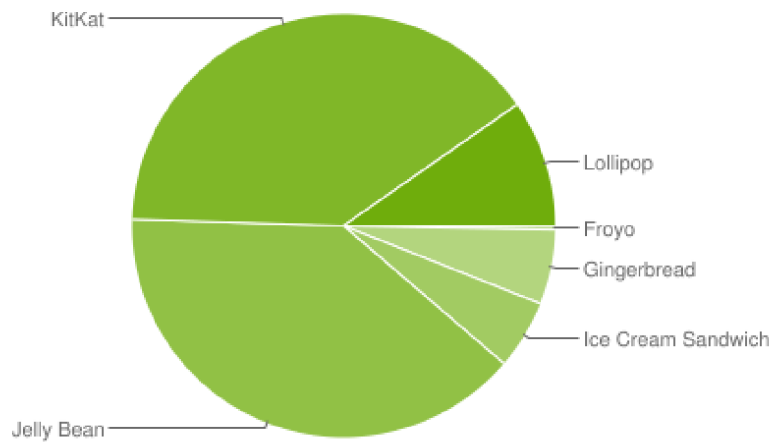
- **songs** – Tato tabulka reprezentuje píseň a všechny její údaje. Pro uložení potřebných dat využijí sloupce a datové typy:
 - id** INTEGER – jednoznačný identifikátor písně
 - name** TEXT – název písně
 - artist** TEXT – jméno umělce
 - text** TEXT – text písně
 - language** TEXT – jazyk
 - favicon** TEXT – název obrázku reprezentující zdroj, rozlišení zdroje
 - favorite** INTEGER – určení, zda je píseň favorizována, či nikoliv
 - text_expanded** INTEGER – určení, zda byl text již expandován
- **watch_timestamps** – V této tabulce jsou uloženy údaje o časování písně pro hodinky. Kromě identifikátoru a odkazu do tabulky songs obsahuje jen časovou známku typu INTEGER.
- **phone_timestamps** – Tato tabulka má stejnou strukturu, jako watch_timestamps, ale je určena pro data časování na telefonu.

4.6 Podporovaná zařízení

Na základě plánovaných funkcí bylo nutné určit podporované vrze systému Android. Několik prvků, jako třeba plovoucí akční tlačítko vyžaduje verzi 5.0 a vyšší, což je značně omezující. V současné době je totiž jen minimum zařízení s touto verzí operačního systému, takže by aplikace jen velmi těžko hledala svoje uživatele. Proto jsem se rozhodl vytvořit kompatibilní prvky, které budou zobrazeny jen na nižších verzích systému. Takto bylo možné veškerou pokročilou funkcionalitu podporovat i na zařízeních s verzí 4.4 KitKat. Touto úpravou se rozšíří poměr zařízení, na které půjde aplikace nainstalovat asi na polovinu³ všech aktivních. Na obrázku 4.3 je znázorněn poměr jednotlivých verzí systému ke dni 14.5 2015.

Aplikace pro hodinky není plánována příliš komplikovaně, takže bude možné podporovat všechny existující verze systému. Aplikace bude fungovat na všech hodinkách s Android wear s kulatým i čtvercovým displejem. Prakticky půjde o verzi 4.4 a vyšší, protože tato verze je nejstarší dostupná na Android wear.

³Převzato z <https://developer.android.com/about/dashboards/index.html>



Obrázek 4.3: Podíl různých verzí systému android na aktivních zařízeních (14. května 2015)

Kapitola 5

Implementace

V této kapitole rozeberu implementační detaily zajímavých částí aplikace a popíšu aplikaci z pohledu uživatele. Věnovat se také budu použitým knihovnám a jejich významu.

5.1 Datové struktury

V této části textu si představíme často používané datové struktury a interní reprezentaci některých dat.

5.1.1 Song

Tato třída je používána pro uložení všech potřebných údajů o písni. Využívá se při operacích s databází, pro udržení a reprezentaci stavu, vytvoření seznamu položek nebo předání dat do spouštěné aktivity v serializovaném formátu. Obsahuje stejné atributy, jaké má tabulka `songs` 4.5 v databázi. Má však navíc proměnnou `linkToDetail` typu `String`. Tento atribut slouží k uchování vyparsované URL stránky detailu písně při vyhledávání nového obsahu z podporovaných serverů a také odlišení, zda je zobrazená píseň již lokálně stažena, nebo jen v režimu prohlížení po vyhledání. Tento údaj není ukládán do databáze, jako ostatní informace o písni, protože není potřebný pro další zobrazení. Stažené písně tedy mají tuto proměnnou prázdnou.

5.1.2 Timestamp

Obecná třída časové známky používaná jak pro údaje o časování na telefonu, tak na hodinkách. Obsahuje tyto atributy:

- **long id** – Jednoznačný identifikátor používaný v databázi
- **long id_song** – Identifikátor určující, ke které písni časová známka patří
- **long timestamp** – Samotná časová známka, prakticky jde o časovou vzdálenost tohoto posunu od předchozího v milisekundách

5.1.3 DatabaseHelper, Datasource

`DatabaseHelper` je třída, kterou využívám pro uložení statických a konstantních názvů tabulek, jejich atributů a podobně. Je zde také obsažena rutina pro inicializaci celé databáze.

`Datasource` obsahuje samotnou databázi `SQLiteDatabase database`. Je určena pro všechny funkce zapouzdřující přístup k databázi. Jsou zde základní metody pro úpravu nebo smazání konkrétních položek, ale i komplexní operace pro rozsáhlejší práci s daty.

5.1.4 Interní reprezentace textů

V průběhu vytváření aplikace bylo nutné se rozhodnout pro jednotný interní formát uchovávání akordů v textech písní. Akordy jsou vsazeny přímo do místa v textu, do kterého patří. Jsou uzavřeny do hranatých závorek, aby bylo možné je později rozpoznat a upravit zobrazení textu tak, aby byly akordy zobrazeny nad textem a ne v něm. Při zkoumání ostatních existujících aplikací jsem často narazil právě na tento formát, který mi připadal pro toto využití vhodný. Hlavní výhodou je jednodušší zpracování a analýza textu oproti variantě, kdy by akordy byly uloženy v samostatném řádku nad textem. Toto formátování je velmi jednoduché a zároveň přehledné pro uživatele. Ten se s ním může setkat například v aktivitě `EditSongActivity`, kde je text zobrazen právě v této interní podobě.

5.2 Aktivity, služby

V předchozím textu jsme si v sekci **4.4 návrh uživatelských obrazovek** vytvořili představu o budoucí podobě aplikace. Nyní popíšu důležité součásti aplikace z pohledu programátora. Zajímat nás budou zejména datové toky mezi aktivitami a implementační detaily jednotlivých funkcí. Tento text se bude částečně prolínat s další podkapitolou **5.4 pokročilé funkce**.

5.2.1 MainActivity

`MainActivity` je hlavní aktivitou celé aplikace. V rámci aplikace je spouštěna jako první. V této aktivitě je inicializován statický objekt `datasource`, který je obsažen ve službě `MyWearableListenerService`. Tento objekt slouží pro přístup k databázi a je pak využíván v rámci celé aplikace. Pro zobrazení položek seznamu písní je v této aktivitě `RecyclerView recList`¹, modernější varianta klasického `ListView`. Tento zobrazovací prvek pro seznamy má vlastní upravený adaptér `adapter` sloužící k propojení s daty. V této aktivitě je možné seznam písní prohledat, seřadit, nebo vyfiltrovat podle zdroje. Vyhledávání probíhá v rámci názvu písně, autora i textu.

Při vstupu do aktivity se obnovuje seznam písní kvůli možné změně v databázi. Tato operace probíhá na pozadí s využitím `AsyncTask`, protože při několika tisících písní může trvat nízké jednotky sekund a v případě běhu v hlavním programovém vlákne by se uživatelské rozhraní zaseklo.

Při spouštění dalších aktivit, které vyžadují informace o vybrané písni, jsou tato data předávána v serializované podobě společně se záměrem ke spuštění aktivity.

5.2.2 DownloadSongActivity

Tato aktivita je funkčně srovnatelná s `MainActivity`. Obsahuje stejný zobrazovací prvek pro seznamy položek a disponuje také vyhledávacím polem. Po zadání a odeslání vyhledávacího výrazu se na pozadí načtou všechna potřebná data z povolených serverů. Data se ze serveru získávají načtením HTML kódu stránky vyhledávání, jako by návštěvník hledal

¹<https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>

na webu. Tato data jsou pak zpracována specificky podle každého serveru, protože jsou na každém z nich formátována jinak. Po zvolení nějaké položky se načtou a zpracují data konkrétní písně a text se zobrazí v `SongDetailActivity`. Celý tento proces bude mnohem podrobněji popsán v sekci [5.4.1 Pokročilé funkce – stahování textů z internetu](#).

5.2.3 `SongDetailActivity`

Tato aktivita nás bude zajímat nejvíce, protože je určena k zprostředkování hlavní funkce celého zpěvníku – zobrazování textu písní. Do aktivity je možné vstoupit buď při výběru již uložené písně, nebo při vyhledávání písně ke stažení.

Jako první uživatele upoutá zajímavé rozvržení prvků a obrázků obalu alba, do kterého píseň patří. Tato funkce automaticky zpracuje název písně a umělce a společně s dalšími klíčovými slovy zadá na pozadí vyhledávání obrázků v Googlu. Po načtení výsledků se pokusí stáhnout a zobrazit první nalezený obrázek. Této funkci se podrobněji věnuji v sekci [5.4.4 Pokročilé funkce – automatické stažení obalu alba](#).

Pro optimální zobrazení textu na obrazovce dochází k několika stupňům analýzy a úpravy textu. Tyto modifikace zajistí, že akordy budou zobrazeny nad textem pouze v jednom řádku a budou expandovány i do sluk, které je neobsahovaly. Dále dojde k úplnému rozepsání refrénů tam, kde byl uveden pouze odkaz. Provedené změny zajistí mnohem lepší praktickou použitelnost celého zpěvníku. Tyto funkce více rozeberu v části [5.4.2 Pokročilé funkce – analýza a úpravy textu](#).

Uživatel zde také může odeslat text do připojeného zařízení se systémem Android wear. To provede stisknutím ikony hodinek s šipkou. Na hodinkách se zobrazí upozornění s informacemi o písni a uživatel pak může na tomto drobném zařízení text prohlížet a používat další doprovodné funkce. Celý proces odesílání dat mezi hodinkami a telefonem spolu s dalšími detaily popíšu v sekci [5.4.3 Pokročilé funkce – Android wear](#).

Pokud již bylo nastaveno časování posunu textu pro mobilní zařízení, bude v akčním menu zobrazena šipka přehrání. Tato ikona reprezentuje spuštění funkce automatického posunu textu. Po jejím zvolení bude textem posouváno přesně v takových intervalech, jaké si uživatel nastavil. Tato funkce společně s nastavením časování je podrobněji popsána v části [5.4.5 Pokročilé funkce – uživatelsky časovaný posun textu](#).

Jednou z dalších praktických funkcí je okamžité vyhledání písně na serveru YouTube jediným stisknutím tlačítka. Tuto možnost ocení zejména začínající hráči, kteří si chtějí rychle vyhledat a poslechnout originální znění skladby. Tato funkce je relativně jednoduchá – v její implementaci je pouze potřeba vyslat systému nový záměr pro načtení webové stránky, jejíž URL odpovídá vyhledání názvu písně a autora na službě YouTube. Systém Android tento záměr reprezentuje tak, že uživateli nabídne otevření příslušné mobilní aplikace a vyhledání provede přímo v ní.

Poslední zajímavostí této aktivity je integrovaný metronom, který umožní základní podporu udržení rytmu hráče. Uživatel si v dialogu nastaví počet úderů za minutu a počet opakovaných úderů v cyklu, na jehož konci bude zahrán úder odlišným zvukem. Pravidelné přehrání zvuku zajišťuje řešení pomocí `TimerTask`, který v určených časech vyvolává akce až do přerušení uživatelem.

5.2.4 `EditSongActivity`

Tato jednoduchá aktivita poskytuje rozhraní pro úpravu písně. Přijme serializovaný objekt reprezentující text a zobrazí jej k úpravám. Tato aktivita nic neukládá do databáze,

veškeré změny jsou provedeny po návratu do předchozích aktivit a vyčtení pozměněného serializovaného textu.

5.2.5 ExportSongActivity

Aktivita zajišťuje serializaci celé tabulky `songs` a její zápis na paměťové úložiště telefonu. Export tedy nepodporuje uložení uživatelských časování posunů, zejména kvůli jejich nekompatibilitě na jiných zařízeních. Pro serializaci využívám funkce třídy `XmlSerializer`. Při ukládání pole `favicon` se dosazuje fixní hodnota reprezentující importované písně neohledě na stávající zdroj, aby byl text po importu správně označen. Po zvolení akce exportu uživatel pouze vidí dialog o exportování a poté hlášku o úspěšném ukončení.

5.2.6 ImportSongActivity

Tato aktivita umožňuje načíst vyexportovanou databázi této aplikace. Uživatel je vyzván k výběru souboru z úložiště zařízení a poté probíhá deserializace a ukládání do databáze. Je zde využíván xml parser `SAX` [1]. Bylo nutné správně pracovat s kódováním a počítat se speciálními znaky. Import umožňuje vytvoření duplicitních písní.

5.2.7 ScrollTimerActivity

V této aktivitě dochází k časování posunů textu pro danou píseň. Text je zobrazen naprosto stejným způsobem, jako při prohlížení v `SongDetailActivity`. Uživatel je upozorněn, že dotykem spustí záznam časování a dalšími dotyky obrazovky provede posunutí textu. Po spuštění záznamu časování je oznámeno, že se údaje uloží prostým návratem zpět. Při ukončování aktivity jsou hodnoty zapsány do databáze a předchozí aktivitě je oznámeno úspěšné nastavení časování. Tato funkce je podrobněji popsána v sekci 5.4.5 pokročilé funkce – uživatelsky časovaný posun textu.

5.2.8 SettingActivity

Tato aktivita zajišťuje správnou inicializaci celého systému preferencí v rámci aplikace. Jsou zde také různá rozšíření jako je například upozornění na nutnost změny časování při změně velikosti písma.

5.2.9 MyWearableListenerService

Tato služba je určena pro zachytávání zaslaných dat z hodinek. Nyní je používána především pro zpracování a uložení dat o časování písně na hodinkách. Ve službě je však také umístěna instance databáze. Tato služba a další související fakta jsou lépe popsána v sekci 5.4.3 pokročilé funkce – komunikace s Android wear.

5.3 Android wear

V této části textu si přiblížíme platformu Android wear. Budeme se zabývat hlavními odlišnostmi od běžné verze systému, zejména těmi v ovládání a zobrazování různých prvků.

5.3.1 Specifika platformy

Ačkoli na hodinkách běží systém Android v téměř plné podobě, některé jeho aspekty jsou velmi odlišné. Většina změn se však týká standardů ovládání a zobrazování prvků z důvodu drobné obrazovky. Není například možné použít zobrazovací prvek `WebView` pro načtení webové stránky. Zkusíme si nejvýraznější odlišnosti podstatné pro tuto aplikaci rozebrat a blíže popsat.

Prostředí systému a ovládání

Základní filozofií systému je především jednoduchost a čistota designu. Celý systém a jeho ovládání je navrženo v tomto smyslu. Základním prvkem prostředí je seznam kartiček upozornění, které se přeposílají z uživatelského telefonu, nebo se zobrazují jen na hodinkách. Tento seznam je po grafické i obsahové stránce inspirován službou `Google Now`², na kterou je většina uživatelů systému Android zvyklá. Uživatel se v kartičkách pohybuje posunutím prstu nahoru nebo dolů. Posunutím prstu zleva doprava dojde k smazání kartičky a tím k odstranění upozornění. Při posunu na opačnou stranu se zobrazí možné akce související s upozorněním. Akce jsou zobrazeny jako velká kulatá ikona přes celou obrazovku pro snadné zvolení. Úplně nad všemi kartičkami nahoře je potom zobrazen ciferník hodinek. Při dotyku ciferníku je možné zadávat hlasové povely, nebo vybrat žádanou funkci ručně. Systém je však spíše zatím zaměřen pro používání v reakci na nějaké upozornění, uživatel by měl být nucen spouštět minimum aplikací ručně.

Aplikace většinou obsahují velmi málo ovládacích prvků, nebo zobrazují jen samotný obsah. Celé rozhraní je navrženo tak, aby uživatel posouval prstem různými směry a odkrýval tak další obrazovky s dalšími možnostmi. Aplikace se běžně zavírají posunutím prstu zleva doprava, podobně, jako při mazání upozornění. Pokud však obsah umožňuje kontinuální posouvání (například mapa), nelze tuto variantu zavření použít. Potom je nutné po dlouhém podržení zobrazit velký červený křížek pro ukončení aplikace.

Upozornění

Upozornění mohou být duplikovány z telefonu, nebo jsou zobrazeny pouze na hodinkách. Je dobré, pokud jsou aplikace inteligentní a proaktivní – například po příchodu do restaurace je uživatel upozorněn, že může přátelům oznámit svou polohu a prohlédnout si menu. Při vyvolání upozornění je uživatel varován pouze vibracemi, protože hodinky nemají žádný reproduktor.

Různý tvar displejů

Programátor musí při návrhu aplikace počítat s různými typy obrazovek. V současné době se prodávají hodinky s kulatým nebo obdélníkovým displejem. Při tvorbě rozhraní je tedy potřeba vytvořit dva soubory rozložení prvků, každý pro jeden typ displeje. Při zobrazení textů je nutné dbát správného rozložení [7] a to zejména na kulatých displejích. Při použití běžných prvků zobrazení se text naformátuje do čtverce, kterému je kružnice displeje vepsaná. Tím jsou však osekány rohy textu a jeho části jsou nečitelné. Je tedy doporučeno použít prvek `BoxInsetLayout`, který zajistí, že se text zobrazí ve čtverci, kterému bude kruh displeje opsán, takže nedojde k žádnému ořezání a text bude stále dobře čitelný. Nevýhodou tohoto řešení je horší využití plochy displeje pro zobrazení hranatých útvarů.

²<http://www.Google.com/landing/now/>

Možné by také teoreticky bylo text zarovnávat dynamicky podle kulatých okrajů, to by však vedlo k nepřehlednému poskakování textu při posunu. Zvolil jsem tedy první možnou variantu přizpůsobení kulatému displeji.

5.3.2 Aktivity, služby

V této kapitole stručně popíšu jednotlivé aktivity a služby, které jsou součástí aplikace pro hodinky. Jejich funkce je však velmi podobná – všechny aktivity zobrazují text stejným způsobem, některé z nich však přidávají nějakou další funkcionalitu.

MainActivity

Tato aktivita je hlavní aktivitou, která se zobrazí po běžném spuštění aplikace přímo z hodinek. Zatím však nemá tato varianta žádný význam, takže je v tomto případě pouze zobrazena instrukce vyzývající odeslání písně z telefonu.

Další možností vstupu do aktivity je volba možnosti **otevřít** z akcí uvedených u upozornění. Tato možnost otevře text v jednoduchém režimu prohlížení bez speciálních funkcí. V aktivitě jsou zpracována serializovaná data připravená při vytváření notifikace. Text písně je do hodinek odeslán po prvních dvou fázích zpracování textu, takže je nutné text ještě přizpůsobit malým rozměrům displeje. Použijí se tedy funkce z třídy `AcordTextModifier`, které zajistí správné zalamování řádků a akordů. Text se dále naformátuje podle nastavení a je připraven k zobrazení uživateli. Tyto úpravy jsou provedeny ve všech dalších aktivitách a proto je tam již nebudu uvádět.

SetTimingActivity

Tato aktivita se spustí výběrem možnosti **upravit časování**. Funkcionalita je téměř shodná jako časování prováděné na telefonu popsané v sekci [5.2.7 ScrollTimerActivity](#). Jediným rozdílem je ukončení časování. Zde se provádí zavřením okna aktivity tažením prsem zleva doprava. Poté se zobrazí standardní dialog s prodlevou pro zrušení ukládání, pokud uživatel vyvolal akci omylem. Pokud časovač dialogu doběhne, nastavené časové známky se odešlou do telefonu a uloží.

AutoScrollActivity

Ke spuštění této aktivity dojde po zvolení možnosti **automatický posun**. Tato funkce posunu je shodná s funkcí popsanou v sekci [5.2.3 SongDetailActivity](#). Jediným rozdílem je spuštění automatického posunu, které se zde provádí dotykem obrazovky.

NotificationUpdateService

Tato služba zprostředkovává zachycení zaslaných dat a zajišťuje vytvoření notifikace na hodinkách. Služba bude s dalším kontextem přehledně popsána v sekci [5.4.3 komunikace s Android wear](#).

5.4 Pokročilé funkce

V této části textu popíšu funkce, které byly obtížnější na implementaci, nebo je vhodné je popsat, protože je žádná konkurenční aplikace nemá. Zaměřím se zejména na podstatné detaily a problémy, se kterými jsem se během implementace setkal.

5.4.1 Stahování textů z internetu

V současnosti jsou aplikací využívány tři zdroje pro stahování textů písní. Jde o servery velkyzpevnik.cz, pisnicky-akordy.cz a ultimate-guitar.com. Veškeré stahování dat se serverů probíhá pomocí načtení HTML kódu požadované stránky a jeho dalšího zpracování.

Pro každý server je v aplikaci uložena adresa jeho vyhledávací stránky, do které se později doplní vyhledávaný výraz zadaný uživatelem. Pro načítání stránky používám knihovnu Jsoup 5.6 určenou pro parsování HTML. Tento proces probíhá na pozadí, aby dlouhé síťové operace nezastavily vykreslování uživatelského rozhraní. Načtený HTML kód je však třeba zpracovat odlišně pro každý server, protože jeho struktura je u každého konkrétního z nich unikátní.

Zde bylo praktické využít principu dědičnosti. Vytvořil jsem základní třídu `AbstractSite`, která deklaruje nutné atributy a metody. Z této třídy pak dědí každá jednotlivá třída konkrétního serveru – například `UltimateGuitarSite`. V těchto konkrétních třídách dojde k vyparsování nalezených položek a jejich zápis do statického seznamu písní v `DownloadSongActivity`. Po načtení všech zdrojů jsou data zpracována a zobrazena podobně, jako by se jednalo o lokální seznam písní. Každá píseň má však v sobě uložené URL pro načtení detailu písně, pokud by byla zvolena.

Po zvolení konkrétní písně dojde k načtení a zpracování textu písně. Zde je také využit princip dědičnosti, základní třídou je zde `AbstractDetail`. Třída reprezentující logiku zpracování HTML detailu písně konkrétních serverů je vybrána na základě toho, z jakého serveru zvolená píseň pochází.

Největší výzvou bylo korektní zpracování zobrazovaných akordů tak, aby byly vždy dosazeny přesně na místo, kam patří. Všechny servery měly značky s akordy obsažené vždy na předchozím řádku před textem, do kterého patřily. Na obrázku 5.1 je část HTML kódu relevantního k pochopení struktury značení textu a akordů. Mezi značkami je určitý počet mezer, který akordy odděluje. Nebylo však přímo možné vyčíst jen veškerý text písně, včetně dceřiných značek, protože by bylo nemožné určit, do jakého místa akord přesně patří. Parser totiž všechny nadbytečné mezery zahazoval a toto chování nebylo možné zakázat. Navrhl jsem tedy metodu, kdy postupně najdu a uložím do pole všechny akordy a počet mezer mezi nimi. Text veškerých akordů pak doplňuji nějakým vlastním “escape” řetězcem, který mi je v jednoduším textu poté dovolí vyhledat. První akord na řádku byl doplněn znaky “B@”, ostatní potom “@@”. Poté proběhlo k vyčtení celého textu a všech dceřiných značek. Díky znalosti počtu mezer mezi akordy bylo později možné správné formátování opět obnovit a odstranit přidané rozlišovací znaky.

```
<h1 class="navez"><strong>Čtení</strong></h1><p class="autor">Interpret/autor: <b>Jablkoň</b>&nbsp;</p><pre
class="pisen"><span class='chord' onclick='sch2("Dmi",event)' onmouseover='sch("Dmi",event)'
onmouseout='hch("Dmi")'>Dmi</span>      <span class='chord' onclick='sch2("Gmi",event)'
onmouseover='sch("Gmi",event)' onmouseout='hch("Gmi")'>Gmi</span> <span class='chord' onclick='sch2("A7",event)'
onmouseover='sch("A7",event)' onmouseout='hch("A7")'>A7</span>
Teprve oči se mi kličí
<span class='chord' onclick='sch2("Dmi",event)' onmouseover='sch("Dmi",event)'
onmouseout='hch("Dmi")'>Dmi</span>      <span class='chord' onclick='sch2("Gmi",event)'
onmouseover='sch("Gmi",event)' onmouseout='hch("Gmi")'>Gmi</span> <span class='chord' onclick='sch2("A7",event)'
onmouseover='sch("A7",event)' onmouseout='hch("A7")'>A7</span>
a ráno tryská z nočních mříží
      <span class='chord' onclick='sch2("Gmi",event)' onmouseover='sch("Gmi",event)'
onmouseout='hch("Gmi")'>Gmi</span>      <span class='chord' onclick='sch2("C",event)' onmouseover='sch("C",event)'
onmouseout='hch("C")'>C</span>      <span class='chord' onclick='sch2("F",event)' onmouseover='sch("F",event)'
onmouseout='hch("F")'>F</span>
noci probdělá s límcem doběla
```

Obrázek 5.1: Fragment HTML kódu ze stránky detailu písně na velkyzpevnik.cz

Další fází bylo předzpracování textu do interní reprezentace aplikace. Všechny akordy byly sloučeny na stejný řádek s textem pomocí statické metody `TextAnalyzer.mergeDownChords`. Po dokončení všech operací této funkce byla aktuální podoba textu zaznamenána pro pozdější uložení. Následovalo další zpracování textu podle nastavení tak, jako probíhá běžně při zobrazení detailu stažené písně.

Zobrazená píseň z vyhledávání ze serverů má všechny funkce fungující stejně jako u kompletně uložené písně. Žádný z pozměněných stavů jako je například editace písně nebo uložení automatických posunů však nevydrží opuštění detailu zobrazeného textu. Píseň lze uložit do zařízení stiskem ikony s motivem diskety.

5.4.2 Analýza a úpravy textu

V dřívějším návrhu jsem popsal rozdělení analýzy a úpravy textu na tři hlavní součásti. Nyní si tyto části rozebereme a popíšeme si dopodrobna jejich jednotlivé funkce. K těmto úpravám dochází při zobrazení detailu písně. Soubory s kódy provádějícími analýzu textu jsou vyčleněny v samostatné knihovně, aby je bylo možné využít i v aplikaci pro hodinky. Třídy obsahují další pomocné funkce související s úpravou textu, které jsou používány i mimo tyto tři části.

Normalizace textu

Vstupními daty tohoto modulu je syrový text písně, takže v nich může být v podstatě cokoli. Některé texty písní jsou zamýšleny jako běžné zpěvníkové texty s akordy, jiné mají pouze pomocí znaků zobrazovat správné držení akordů na krku kytary. V jiných jsou zase uvedeny naprosté nesmysly a text se ani nedá považovat za píseň. Ve všech případech však musí tento modul rozpoznat dodaný obsah a pokusit se rozeznat a označit jeho části a vylepšit formátování, nebo neprovádět žádné změny. Normalizace je podporována jen u běžných písní, které mají akordy zobrazené nad samotným textem.

Většina programové logiky související s tímto logickým modulem je obsažena ve třídě `TextNormalizer`. Samotný proces se pak spouští funkcí `normalizeText`, které se předá text písně a její návratovou hodnotou je upravený výsledek. Na počátku všech úprav se odstraní všechny vícenásobné mezery na začátku a konci celého textu. Pokud text neobsahuje číslované sloky nebo označené refrény, modul se pokusí jejich hranice odhadnout. To provede tak, že nejdříve projde text a určí standardní počet mezer mezi řádky v rámci sloky. Toto je potřeba z důvodu, že někteří uživatelé vkládají text na každý řádek, jiní ve sloce vynechávají řádky pro větší přehlednost. Díky této hodnotě je možné určit hranice jednotlivých bloků, protože bývají odděleny větším počtem vynechaných řádků. Pokud však text není strukturován ani pomocí vynechaných řádků, je rozpoznán pouze jeden blok. V takovém případě není možné spolehlivě rozpoznat jednotlivé části písně a text zůstane celým modulem nezměněn.

Po odlišení jednotlivých bloků se musí rozpoznat, jaké z nich jsou refrény a které slokami. Modul se pokusí rozpoznat podobnost mezi začátky jednotlivých bloků a ty podobné označí jako refrény. Podobnost je určována porovnáním jednotlivých množin slov bloků. Pokud je mezi nimi větší, než 80% shoda, tak jsou tyto bloky označeny jako refrén. Z těchto bloků je zachován v původním stavu pouze první a ostatní jsou nahrazeny značkami refrénu. Zbylé bloky se očíslovají jako sloky. Celá tato analýza probíhá pouze, pokud jsou v textu nalezeny akordy, protože jinak nemá valný význam. Podporovány jsou i jiné varianty, kdy jsou značeny například jen akordy, ale sloky nikoliv.

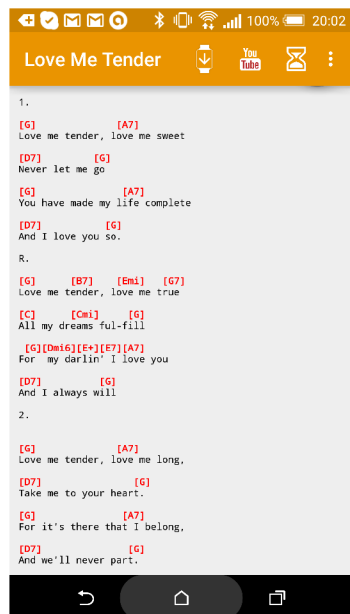
Výstupem tohoto modulu je upravený text, který má všechny bloky správně označené. Druhou variantou je selhání úpravy, kdy nedojde k modifikaci textu. Takový text je dalšími moduly ignorován.

Analýza a úprava akordů a refrénů

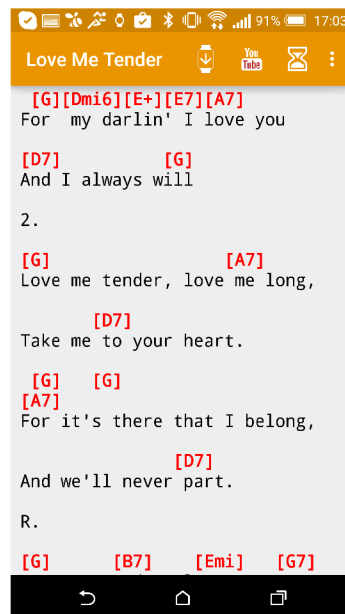
Tento modul na vstupu očekává správně naformátovaný a označený text. Jeho cílem je odstranit nejzásadnější nedostatky textu způsobující problémy při čtení na menších obrazovkách mobilních zařízení. Za tyto nedostatky považují formát, kdy jsou akordy uvedeny pouze u první sloky a prvního výskytu refrénu a kdy je refrén uveden pouze jednou a potom je na něj jen odkazováno. Tyto nedostatky jsou však u textů vkládaných uživateli velmi časté.

Většina kódu logického modulu je obsažena v třídě `TextAnalyzer`. Samotná analýza se pak vyvolá metodou `getFinalText`. Její návratová hodnota obsahuje řetězec s upraveným textem. Součástí této třídy jsou i dvě podtřídy – `FragmentHolder` a `AcordHolder`. Tyto třídy jsou využívány pro uchovávání manipulovaných dat. Za fragment se považuje sloka, refrén, nebo odkaz na něj. `FragmentHolder` slouží pro uložení informací o analyzovaném fragmentu. Mezi tyto informace patří indexy nalezené značky patřící k fragmentu, indexy začátku a konce textu a tento text v podobě řetězce. Fragment je dále označen typem (zda se jedná o sloku, refrén nebo odkaz na něj) a je určeno, zda obsahuje akordy, či nikoliv. `AcordHolder` slouží k uložení zpracovávaných akordů. Obsahuje text akordu, jeho příslušnost ke sloce nebo refrénu a informaci, ke kterému řádku a na jakou pozici na něm patří.

Během samotné analýzy dojde k rozpoznání částí textu a jejich uložení do výše popsaných struktur. Poté může dojít k samotnému odstraňování nedostatků. Nejdříve dochází k přepsání akordů z první sloky i do ostatních, pokud v nich nejsou obsaženy. Toho je dosaženo dosazením akordů do stejného relativního místa na řádku, jako v příslušném řádku v první sloce s akordy. Pozice nově vkládaného akordu tedy není určena absolutně např. počtem znaků od začátku řádku. Výpočet probíhá tak, že se určí poměrná pozice původního akordu mezi začátkem a koncem řádku. Do stejné vzdálenosti vzhledem k počtu znaků se pak umístí do upravované sloky. Tento přístup je mnohem vhodnější než pouhé vkládání na základě absolutního počtu znaků od začátku celé původní sloky, protože text upravované sloky je většinou trochu jiné délky a akordy se pak nesprávně přepíší na jiný řádek. Tuto jednodušší formu úpravy jsem používal dříve, rozhodl jsem se však přejít na metodu zohledňující každý řádek zvlášť, protože jsem od ní očekával mnohem lepší výsledky. Po dokončení rozšíření akordů se provede rozepsání refrénu místo všech značek, které za sebou text refrénu nemají. Srovnání textů zpracovaných těmito různými metodami můžeme vidět na obrázku 5.2 Varianta (a) při expanzi uvažuje o každém řádku zvlášť, zatímco méně vhodná metoda (b) přepisuje akordy pouze na základě vzdálenosti ve znacích od začátku sloky.



(a) Výsledek expanze zohledňující řádky



(b) Expanze zohledňující počet znaků

Obrázek 5.2: Výsledky dvou různých metod přepsání akordů. Akordy byly původně obsaženy jen v první sloce a refrénu.

Po úpravách tímto modulem je na výstupu text písně, který je řádně označen a má akordy v celém objemu textu. Refrény jsou v něm rozepsané a dobře označené, takže při zobrazení jakékoli části textu je možné bez problémů hrát i zpívat. Po průchodu tímto modulem se text ukládá do databáze a označuje jako upravený. Tyto fáze se při zobrazení již upraveného textu později neopakují.

Optimalizace pro různá zařízení

Tento logický modul zajišťuje správné naformátování výsledného textu pro konkrétní obrazovku. K jeho spuštění dojde pokaždé při zobrazení textu a výsledek této části úprav se nikam neukládá, pouze se zobrazuje. Cílem modulu je upravit text tak, aby se předešlo znepráhlednění textu několika řádky akordů nad sebou, ke kterému dochází zejména při větší velikosti textu u konkurenčních aplikací. Modul dále převádí text z interní reprezentace do zobrazitelné formy a provádí další úpravy jako je nastavení velikosti a typu písma, obarvení textu akordů apod. Většina logiky modulu je obsažena ve třídě `AcordTextModifier`.

Nejdříve musí dojít ke korektnímu rozdělení řádků tak, aby byly přidány znaky konce řádků “\n” na místa, kde by normálně došlo vzhledem k velikosti obrazovky k zalomení textu. Tuto úpravu provádí metoda `splitBrokenLines`. V této metodě se k původně prázdnému řetězci cyklicky přidávají slova z řádku a je kontrolováno, zda by se text na obrazovce zalomil. Pokud ano, je před další slovo vložen znak konce řádku. Tímto se zajistí, že při následující extrakci akordů z interní reprezentace do zobrazení nad textem nedojde k navrstvení více řádků akordů nad sebe, což by značně snížilo čitelnost celého textu.

V další fázi úprav dojde k vyjmutí akordů z textu a jejich zobrazení nad textem. Po vyextrahování akordu se vždy na řádek s akordy vloží počet mezer odpovídající počtu znaků do dalšího akordu, aby byl i on při extrakci správně zarovnán. V poslední řadě dojde k nastýlování akordů a textu podle nastavení. Dojde k barevnému zvýraznění akordů

a nastaví se velikost písma. Je zvolen typ písma, ve kterém mají všechny znaky stejnou šířku, aby nedocházelo k nesprávnému posunu akordů oproti textu. Po této fázi je již text připraven k zobrazení uživateli.

5.4.3 Komunikace s Android wear

Velice zajímavým rozšířením aplikace je podpora hodinek s operačním systémem Android wear. Aplikace nyní umožňuje odeslat text písně do hodinek, kde je možné jej zobrazit a provádět s ním další akce. V této části textu si představíme zejména komunikační část tohoto řešení. Pokud čtenář není seznámen s funkcemi hodinek Android wear nebo s navrhovanou strukturou a funkcionalitou aplikace na hodinkách, doporučuji nejdříve přečíst sekci 5.3 popisující aplikaci pro tento typ zařízení.

Komunikace s hodinkami Android wear probíhá s využitím systémové třídy `GoogleApiClient`. Z každého místa, odkud je potřeba posílat data do dalšího zařízení je instanciován a nastaven objekt tohoto typu. Data určená k odeslání se serializují a přidají do objektu typu `PutDataMapRequest`. Je dokonce možné i odesílat malé obrázky, pokud jsou předtím převedeny do datového typu `Asset`. Celý datový balík se poté odešle pomocí metody `Wearable.DataApi.putDataItem(myGoogleApiClient, putDataRequest)`. Tím je odesílání dat vyřešeno, zbytek závisí na protistraně, která musí data zpracovat. Není však možné ověřit, zda data v pořádku dorazila. Výše popsáný systém odesílání dat je použit na straně telefonu v `SongDetailActivity` pro odeslání textu do hodinek a na straně hodinek v `SetTimingActivity` pro odeslání časových známek posunů mobilnímu zařízení k uložení.

Strana posluchače zaslaných dat je o něco komplikovanější. Posluchač musí být implementován ve službě, protože aktivita nemusí být vždy spuštěna, nebo ji systém může po chvíli běhu na pozadí ukončit. Tuto službu je také potřeba označit v souboru `AndroidManifest.xml`, aby opravdu dostávala hlášení o poslaných zprávách. Třída služby musí dědit z `WearableListenerService`. Celé zachytávání a zpracování dat se pak děje v přepsané metodě `onDataChanged`. Zde je nutné přijatá data podle vybraných identifikátorů zase deserializovat a tím je jejich samotný přenos ukončen. Tento posluchač je implementován v telefonu ve třídě `MyWearableListenerService` a probíhá tam zpracování přijatých časových známek a jejich uložení do databáze. Na straně Android wear aplikace je tento posluchač v rámci třídy `NotificationUpdateService` a přijímá všechna data písně potřebná k zobrazení.

Tato přijatá data jsou však na straně hodinek dále zpracovávána. Je potřeba ještě vyvolat upozornění, které se uživateli zobrazí při přijetí. Ve třídě posluchače je tedy i logika správy upozornění, která obsluhuje správné vytvoření a zničení upozornovací kartičky. Při vytváření upozornění se určí, jaká aktivita má být spuštěna po výběru určité zobrazené akce. Zároveň jsou opět serializována data, které budou této aktivitě později předána. Pokud přijatá data obsahují časové známky posunů, tak bude potřeba zobrazit jednu možnost navíc – zapnutí režimu automatického posunu textu. Společně s informacemi o textu písně však může dorazit dokonce i stažený obrázek obalu alba. V tomto případě je obrázek využit jako pozadí pro vyvolané upozornění. Upozornění se všemi komponenty tedy bude obsahovat (stránky zleva doprava):

- **Kartička upozornění** – Obsahuje název skladby, umělce. U kartičky je zobrazena ikona aplikace a na pozadí je obrázek obalu alba.
- **Úprava časování** – Vstup do obrazovky pro časování posunu textu.

- **Automatický posun** – Vstup do aktivity s automatickým posunem textu.
- **Otevřít** – Otevření textu běžným způsobem.

Celé schéma pohybu v upozornění na Android wear je popsáno v sekci [4.4.8](#) návrh uživatelských obrazovek – Android wear upozornění.

5.4.4 Automatické stažení obalu alba

Aplikace automaticky stahuje obal alba pro lepší dotvoření dojmu z grafické stránky prohlížení textů. Pro stažení obrázku je použito Google vyhledávání obrázků. Při zobrazení obrazovky detailu textu se okamžitě spustí vyhledávání obrázku na pozadí. Jako klíčová slova jsem po různých testovaných variantách zvolil kombinaci názvu písně, jména umělce a slova “cover”. Toto spojení vyhledávalo nejrelevantnější výsledky a fungovalo docela spolehlivě pro český i anglický název písně, nebo umělce.

Stažení obrázku probíhá ve dvou asynchronních operacích. V první se pouze vyvolá Google vyhledávání a v HTML kódu se nalezne první výsledek vyhledávání. Odkaz na samotný obrázek se potom předá do další části, kdy se dokončí samotné uložení obrázku. Pokud je stažení úspěšné, může být nahrazen základní obrázek značící, že obal alba zatím ještě není stažen.

5.4.5 Uživatelsky časovaný posun textu

Tato funkce je dostupná v aplikaci pro mobilní zařízení i hodinky a v obou z nich je implementace téměř totožná. Po vstupu do aktivity k záznamu časování a prvním dotyku obrazovky je celá funkce spuštěna. Zobrazí se malý nápis s uplynulým časem a uloží se časová známka začátku časování. V této aktivitě jsou přepsány funkce zpracování dotykových gest, takže není možné posouvat textem běžným způsobem. Jediným zaznamenaným gestem je prostý dotyk textu bez tažení. Po takovém dotyku je obrazovka posunuta o asi polovinu své výšky a je uložena časová známka akce. Takto se proces opakuje až do ukončení časování. K tomu dojde při návratu uživatele do předchozí aktivity, kdy se časování uloží. Zde je jediný rozdíl v aplikaci pro Android wear, kdy je kromě dotyku povoleno i gesto posunutí prstem zleva doprava pro zavření aplikace a vytvoření dialogu odpočítávajícího uložení časování.

Reprodukce posouvání je poměrně jednoduché. Po spuštění této funkce se vytvoří objekt typu `ScheduledExecutorService`. Ten umožní vykonat stejnou metodu, jaká byla použita pro posunutí textu při záznamu časování a zároveň naplánuje čas další akce podle další časové známky. Uživateli však není na rozdíl od záznamu posunu zakázáno posouvat text gesty, takže si může přesnou pozici upravit podle libosti i během samotného automatického posouvání. Celý systém funguje, protože rozložení prvků je na obrazovce časování i prohlížení stejné. Jediný problém nastane při změně velikosti textu nebo jeho úpravě, kdy se vykonává jiný posun v textu, než uživatel očekává. Uživatele však při změně velikosti písma na tento fakt upozorňuji a navíc předpokládám, že si uživatel nastaví jednu velikost písma a tu bude potom používat celou dobu.

5.5 Další zajímavé funkce

Zde si v rychlosti představíme další zajímavé funkce a drobné detaily, které nebyly zatím popsány, ale můžou mít významný vliv na pohodlí užívání aplikace.

Pro časté uživatele se spoustou uložených písní je praktický rychlý posuvník s přibližným zobrazením písmene abecedy, u kterého se právě nachází. Tuto funkci je možné u seznamu položek `ListView` běžně zapnout. Ve svém projektu jsem však pro seznam položek zvolil modernější a paměťově efektivnější `RecyclerView`, které tuto možnost nemá. Implementoval jsem tedy srovnatelnou funkci s podobným chováním. Funkce je znázorněna na obrázku 4.1.

Při zobrazení seznamu písní se zejména na menších displejích často nevejde celý název písně a autora. Použil jsem tedy funkci, která bude celým textem pomalu posouvat zprava doleva, aby uživatel mohl přečíst i původně skryté části textu.

5.6 Použité knihovny

Při řešení projektu jsem využil několik knihoven, které urychlily implementaci nějaké funkcionality. V této sekci popíšu použité knihovny nebo převzatý kód a důvod, proč bylo vhodné je využít.

Color picker

Tato knihovna mi usnadnila práci s vytvářením specifické položky v nastavení. Systém Android neposkytuje žádnou nativní možnost výběru barvy a vytváření vlastního řešení by bylo příliš komplikované. Bylo tedy zcela logické využít existující knihovnu, která výběr barvy v nastavení podporovala. Knihovna je dostupná na adrese:

<https://github.com/yukuku/ambilwarna>.

Jsoup

Pro účely stahování písní z webu bylo nutné použít nějaký nástroj na usnadnění zpracování HTML kódu. Protože Java žádný takový nástroj nativně nepodporuje, bylo nutné sáhnout po knihovně. Pro dané účely se hodil nejvíce právě parser Jsoup. Zvolil jsem ho zejména z důvodu relativně snadného použití a dobré dokumentaci. Parser mi umožnil zpracovat data mnohem efektivněji, než když bych používal vlastní řešení. Knihovna je dostupná na adrese:

<http://jsoup.org/>.

AutoSummaryListPreference

Na tento projekt jsem narazil při hledání řešení pro zobrazení vybrané položky vícenásobného výběru v nastavení. Android nepodporuje běžně tuto často vyžadovanou možnost a je tedy třeba vyvinout vlastní řešení. Nalezený způsob byl velmi prostý a bylo tedy zbytečné vymýšlet jinou variantu implementace. Projekt je dostupný na adrese:

<https://gist.github.com/Sloy/5891065>.

FloatingActionButton

`FloatingActionButton` je často využívaný prvek v Material designu. Jde o kruhové tlačítko, které se nachází přímo v obsahu a slouží ke zvýraznění nějaké akce. Toto řešení je součástí oficiálních demo aplikací distribuovaných Googlem v rámci SDK. Proto bylo využití tohoto způsobu implementace velmi vhodné. Další informace o `FloatingActionButton` jsou

dostupné na adrese:

<https://developer.android.com/samples/FloatingActionButtonBasic/index.html>.

LollipopContactsRecyclerViewFastScroller

Funkci rychlého posuvníku se zobrazením písmene abecedy jsem plánoval již v návrhu aplikace. Během implementace jsem však zjistil, že funkce je podporovaná pouze u seznamu položek `ListView`, který jsem však nepoužil. Při hledání řešení na internetu jsem našel tento projekt. V rámci používání řešení jsem také odhalil, nahlásil a opravil chybu zobrazení posuvníku při použití menšího množství prvků. Další informace o projektu jsou dostupné na adrese:

<https://github.com/AndroidDeveloperLB/LollipopContactsRecyclerViewFastScroller>.

Kapitola 6

Testování aplikace a zpětná vazba uživatelů

Po ukončení vývoje plánovaných funkcí je před vydáním vhodné aplikaci otestovat. V této kapitole popíšu různé přístupy k testování, které jsem využil. Aplikaci jsem testoval zejména z hlediska výkonnosti, pro nalezení chyb a pádů aplikace a také z hlediska uživatelského rozhraní.

6.1 Úspěšnost analýzy a úpravy textu

Analýza a následná úprava textů je nejdůležitější funkcí této aplikace, která ji výrazně odlišuje od konkurence. Je tedy potřeba se ujistit, že tato funkce funguje velmi dobře ve většině případů. Zkoumat budeme zejména to, zda analýza vůbec proběhne kvůli vhodnosti zdrojových dat a zda nezpůsobí nějaké naformátování textu, které by se dalo považovat za chybné.

Z každého serveru jsem náhodně vybral **10 písní**, které jsem prohlédl v původním a upraveném stavu. Posoudil jsem, jak úprava textu dopadla. Úpravu jsem zařadil do jedné z kategorií:

1. **Úspěšná úprava** – Výsledný text je upraven do předpokládaného stavu bezchybně
2. **Úspěch s drobnými nedostatky** – Výsledný formát úpravy je správný, objevují se však drobné nedostatky, např. zbytečné mezery, nestandardizované značky apod.
3. **Selhání úpravy** – Zobrazený text je úpravou poničen, nebo jsou špatně rozpoznány jeho části.
4. **Neprovedení úpravy** – Text nebyl upraven kvůli nevhodnosti vstupního formátu.

	1. úspěch	2. část. úspěch	3. selhání	4. neprovedení
velkyzpevnik.cz	5	2	1	2
pisnický-akordy.cz	3	1	2	4
ultimate-guitar.com	6	1	3	1

Tabulka 6.1: Data úspěšnosti úprav náhodného vzorku písní z jednotlivých serverů

Za nejobtížnější variantu úpravy považují, pokud aplikace musí korektně rozeznat a označit jednotlivé části textu v rámci modulu normalizace a pokud musí také provést expanzi akordů s refrény. Takto text projde maximální možnou změnou. K úspěchu v úpravě nejčastěji vedl text, který byl již správně označen, ale nebyl rozexpandován, nebo text, který neměl žádné značení.

K úpravě s drobnými chybami docházelo zejména v textech se standardním značením, které obsahovaly nějaké neběžné prvky, jako například pokyn k vícenásobnému zahrání refrénu. V budoucnu se pokusím navrhnout řešení detekce a ignorování těchto odchylek.

Důvodem pro selhání byly nejčastěji velmi dlouhé texty, které obsahovaly různé směsice textů s akordy a značek k držení akordů na krku vyjádřených pomocí znaků. Tyto typy textu se nejčastěji vyskytovaly na serveru `ultimate-guitar.com`. Chyby dále způsobovaly neznačené texty s nepravidelným řádkováním, kdy docházelo k nesprávnému rozeznání slok a refrénů.

Nejčastějším důvodem pro neprovedení úpravy byl text bez akordů, nebo zapsaný v jednom bloku. V obou těchto případech je toto vyhodnocení správné. Tímto případem končila často úprava na serveru `pisnicky-akordy.cz`, protože písně navzdory názvu zdroje často akordy neobsahovaly.

Úspěšnost úprav vlastně částečně reflektuje schopnost jednotlivých komunit serverů vkládat písně se smysluplným značením. V tomto vnímám jako nejúspěšnější servery `velkyzpevnik.cz` a `ultimate-guitar.com`. Chyby v úpravě u textů z druhého zmiňovaného zdroje by bylo možné odstranit rozpoznáním vzoru používaného pro značení držení akordů a ignorováním úpravy takových textů. Na základě zjištěných údajů jsem postupně ladil jednotlivé algoritmy, aby se co nejvíce textů způsobujících výsledky patřící do kategorie selhání úpravy vůbec neupravovalo. Toho se mi podařilo dosáhnout zejména u neznačených písní s nepravidelným řádkováním a v dalších specifických případech. Data tohoto posouzení úspěšnosti byla naměřena až po všech modifikacích upravujících algoritmů.

6.2 Výkonnost

Reálná výkonnost a rychlost aplikace bude velmi důležitá zejména pro majitele starších zařízení. Je nutné se ujistit, že v hlavním programovém vlákne neprobíhají žádné operace, které mohou trvat delší dobu a tím zastavit překreslování uživatelského rozhraní. Toto nebezpečí hrozí zejména u databázových operací, zacházení se sítí, nebo analýze dlouhých textů. Dále je vhodné se ujistit, že aplikace je schopna zpracovávat velké množství dat v přijatelné době. Testování plynulosti rozhraní přenechám uživatelům a výsledky zjistím pomocí zpětné vazby. Rychlost zpracování velkého množství dat se pokusím prověřit sám.

6.2.1 Testovací data

Pro otestování extrémního vytížení aplikace bylo nutné získat velké množství reálných dat. K tomuto účelu jsem využil databázi aplikace Zpěvníkátor, která obsahuje 10584 písní. Vytvořil jsem jednoduchý modifikační skript v jazyce Groovy, který upravil data do formátu importovatelného do mé aplikace. Podobným skriptem jsem data modifikoval do formy akceptované aplikací Zpěvník. Tuto aplikaci jsem lépe popsal v sekci [2.1.3](#) Podobná stávající řešení – Zpěvník.

Zatímco import do mé aplikace trval na zařízení s čtyřjádrovým procesorem Qualcomm Snapdragon 801 přibližně 45 sekund, Zpěvníku tato akce trvala přes půl hodiny. Zde je vidět řádový rozdíl ve výkonosti oproti konkurenčnímu řešení. Zatímco načtení seznamu

všech těchto písní trvá v mé aplikaci zhruba 9–12 sekund, Zpěvník tuto akci provádí dlouhé minuty a často je jeho běh během vykonávání ukončen. Export takovéto databáze zabere mé aplikaci zhruba 25 sekund, Zpěvníkátor ji však provádí asi 15 minut. Vyhledávání v databázi zabere při takovémto objemu písní obvykle asi 11 sekund, Zpěvníkátor tuto funkci bohužel pro srovnání nenabízí.

S naměřenou výkonností jsem docela spokojen, protože je řádově lepší, než u konkurenční aplikace. Reálný uživatel jen stěží dosáhne počtu bytů jen tisíce písní, takže všechny operace budou pravděpodobně probíhat maximálně v řádu nízkých jednotek sekund i na starších zařízeních.

6.3 Uživatelské testování

Cílem uživatelského testování nebude získat exaktní data, nebo přesně zhodnotit použitelnost aplikace podle nějaké metodiky. Mým záměrem je zejména vypořádat nejasnosti v uživatelském rozhraní, které mohou uživatele zmást a kvůli kterým například nemusí požadovanou funkci vůbec nalézt.

6.3.1 Betaverze v Google Play Store

Tento distribuční kanál umožňuje vydání aplikace v režimu betaverze. Výhodou je možnost vyzkoušení distribuce běžným způsobem, ale jen pro omezené množství lidí. Proběhne také prověření aplikace automatickými skripty a osobami z Googlu testujícími kvalitu aplikace. Program je dostupný k otestování pouze vybrané skupině uživatelů, kteří obdrží pozvánku do testovací komunity na síti Google+. Nejdůležitějším cílem tohoto způsobu testování je ověřit, zda aplikace funguje bezproblémově na různých typech zařízení. Během této fáze je také možné po pádu aplikace odeslat systémové hlášení, které se mi zobrazí ve webové vývojářské konzoli. Dalším způsobem získání zpětné vazby je komunikace s testery v rámci komunity. Díky tomuto testování jsem odhalil jednu chybu způsobující pád aplikace a získal zpětnou vazbu vedoucí k několika drobným vylepšením, jako je například posouvající se text názvu písně a autora, pokud se nevejde na obrazovku celý.

6.3.2 Testování navigace v aplikaci

Zde si představíme průchod aplikací při několika různých cílech uživatele. Zamysleme se nad efektivitou a smysluplností rozhraní a necháme uživatele provést uvedené postupy. Aplikaci budu testovat na uživatelích s různou zkušeností zacházení se systémem Android, na lidech různých věkových skupin, nebo na testerech, kteří se již s ovládním aplikace setkali.

Cíl: Stáhnout a zahrát si načasovanou píseň s akordy

Toto by měl být nejběžnější scénář používání aplikace domácím hráčem na kytaru. Uživateli se po spuštění zobrazí hlavní obrazovka **moje písně**. I nový uživatel aplikace by měl pochopit význam akčního tlačítka s ikonou plus a dostat se tak k možnosti zadat výraz vyhledávání. Po vyhledání a zobrazení detailu písně bude nutné najít ikonu pro uložení. Poté by se měl uživatel vrátit do seznamu stažených písní a tam ji nalézt. Po otevření textu by měl provést časování posunů. Po dokončení by měl identifikovat ikonu spuštění časování. Tím je celý test ukončen.

Cíl: Uživatel bude chtít změnit velikost písma

Uživatel bude mít za úkol otevřít libovolnou píseň ze seznamu. V aplikaci bude předem nastavena velikost textu na nejmenší, takže upozorníme uživatele, že by si mohl velikost písma změnit. Poté jej pozorujeme v jeho snaze najít tuto možnost. Cílem je ověřit, že je uživatel schopen dohledat nějakou možnost v obrazovce nastavení, pokud mu současný stav nevyhovuje.

Cíl: Vyhledat již stažený text mezi velkým množstvím položek

Uživateli spustíme aplikaci s testovacími daty zahrnující více než 10 000 písní a ukážeme mu systém označení písně ikonou zdroje. Poté před uživatelem stáhneme libovolnou píseň a požádáme jej, ať ji najde mezi ostatními. Cílem je ověřit, po jaké metodě usnadňující nalezení písně uživatel nejčastěji sáhne a zda píseň nalezne dostatečně rychle.

Vyhodnocení

Testování se celkem zúčastnilo 7 osob. Mezi nejčastější chyby uživatelů při navigaci patřily pokusy o vyhledávání písně ke stažení mezi již staženými položkami, lehké zmatení po otevření téměř prázdné stránky pro vyhledání písně ke stažení, neschopnost na první pohled najít nastavení velikosti písma mezi ostatními preferencemi, nebo hledání možnosti nastavení v úpravě textu písně. Uživatelé naopak neměli problém s nalezením samotného menu nastavení, se stažením písně, nebo s jejím načasováním. Bez problémů bylo také nalezení a otevření dané položky mezi větším množstvím písní, kdy lidé většinou použili funkci vyhledávání. Většina uživatelů rychle pochopila význam jednotlivých obrazovek i smysl celé aplikace. Uživatelé byli po předvedení funkce pro automatické vylepšování textů často nadšeni a zajímalo je, jak taková věc funguje.

Na základě zpětné vazby jsem v obrazovce pro vyhledávání písní ke stažení vytvořil ikonu s nápisem **Začít vyhledávat**. Po dotyku nápisu nebo ikony se objeví kurzor v řádku vyhledávání a vysune se klávesnice. Upravil jsem také popisek položky změny písma v nastavení, aby se tato možnost dala jednodušeji najít. S uživateli jsem dále diskutoval o jejich dojmech z používání, které byly nejčastěji velmi pozitivní. Takto jsem si alespoň velmi neformálně ověřil smysluplnost některých funkcí a získal několik námětů na vylepšení.

Kapitola 7

Vydání aplikace a propagace

Pečlivé naplánování vydání aplikace je podobně důležité, jako mít samotnou aplikaci pěknou a zajímavou. Bylo tedy nutné postupně navrhovat a vylepšovat grafické prvky, které mají za úkol přesvědčit návštěvníka stránky aplikace k instalaci. V této kapitole si projdeme jednotlivé kroky a rozhodnutí, které budou zásadní pro vytvoření prvního dojmu při návštěvě stránky aplikace.

7.1 Uvedení do Google Play Store

Skutečné uvedení do obchodu mělo několik fází. V předchozím textu jsem popisoval zejména fázi beta testování, která se nejvíce zaměřovala na vylepšování samotné aplikace. Před produkčním vydáním však byla potřeba se rozhodnout pro finální název aplikace a vytvořit propagační materiály.

Nejzásadnější volbou je jednoznačně jméno a ikona aplikace. Asi polovina instalací všech aplikací je vyvolána uživatelem, který potřebuje nějakou funkci a rozhodne se zadat [8] klíčová slova do vyhledávání v aplikačním obchodě. Se správně zvoleným názvem, ikonou a klíčovými slovy v popisu je možné dosáhnout docela zajímavých výsledků i bez propagace.

Pro aplikaci jsem zvolil český název Moderní Zpěvník. V anglickém jazyce je to pak Material Songbook. Aplikace má již svým názvem sdělit, že je vytvořena pro nejnovější systém Android 5 a je v ní použit materiální design. Ikona aplikace (obrázek 7.1) potom jen dokresluje toto sdělení jednoduchými sytými barvami na různobarevném pozadí s realistickými stíny.



Obrázek 7.1: Ikona aplikace

Propagační grafika má na pozadí zobrazen rozmazaný přírodní motiv, který vyjadřuje jedno z možných míst použití aplikace. Na jednotlivých obrázcích jsou zobrazeny různá zařízení s důležitými částmi aplikace a relevantním popisem. V grafice je také znázorněna funkce automatické úpravy textů jako hlavní výhoda zpěvníku.

Aplikace cílí zejména na české uživatele, protože dva ze tří podporovaných serverů jsou české. V aplikaci je však obsažen i anglický překlad, který se zobrazí všem uživatelům mimo Českou republiku. Podobně je nastaven i záznam v obchodě Google Play – pokud je návštěvník z cizí země, bude mu zobrazen anglicky název aplikace, propagační grafika i popis. Aplikace je dostupná na adrese: <https://play.google.com/store/apps/details?id=com.komoi.materialsongbook>

7.2 Propagace

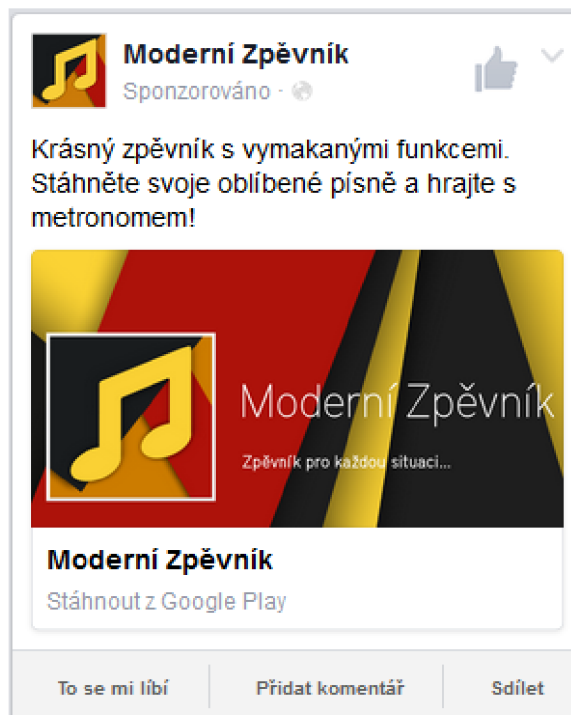
Společně s vydáním aplikace jsem plánoval její propagaci. Ta je podobně důležitá jako kvalita samotné aplikace. Pro propagaci jsem zvolil následující přístupy:

- **Rozšíření mezi okruh přátel** – Při vydání jsem na různých sociálních sítích sdílel příspěvek popisující aplikaci. Požádal jsem některé přátele o další sdílení příspěvků.
- **Propagační komunity na Google+** – Na sociální síti Google+ existuje mnoho komunit, které se zaměřují na podporu začínajících vývojářů. V těchto komunitách jsou nejčastěji ostatní vývojáři a lidé, kteří chtějí objevovat nové aplikace. Vybral jsem nejrelevantnější a největší z nich a umístil do nich krátký příspěvek vystihující aplikaci s výzvou k jejímu stažení.
- **Vydání recenze komunitním bloggerem** – Po domluvě s bloggerem píšícím na stránce [androidwearapps.org](http://www.androidwearapps.org) mi byl vydán krátký článek hodnotící aplikaci. Blogger také článek dále propagoval svými účty na sociálních sítích. Různé kanály tohoto webu sleduje dohromady asi 6000 lidí. Recenze je dostupná na adrese: <http://www.androidwearapps.org/2015/05/material-songbook-now-with-android-wear.html>.
- **Reklama na facebooku** – Rozhodl jsem se investovat částku 400Kč do propagace aplikace prostřednictvím reklamního systému facebooku. Cílem bylo zejména vyzkoušet výkonnost tohoto typu propagace a odhadnout cenu za jednu instalaci aplikace. Výhody tohoto systému spočívají zejména ve velmi přesném cílení podle zájmů, dobře pochopitelných zobrazovaných statistikách a dobré výkonnosti reklamy. V dalším textu blíže rozeberu tuto formu propagace.

7.2.1 Reklama na sociální síti Facebook

Využití reklamy této platformy je velmi vhodné zejména pro propagaci mobilní aplikace, protože nastavení reklamy umožňuje cílit pouze na uživatele používající mobilní telefon¹. Pomocí dalších nastavení a omezení je možné vytvořit takovou reklamu, která by měla být co nejrelevantnější pro lidi, kterým se zobrazí. Pro zacílení jsem omezil zobrazení jen lidem z České republiky mezi 15–35 lety používající mobilní telefon nebo tablet se systémem android 4.4 a vyšším. Dále bylo zobrazení limitováno na osoby se zájmem o hudbu, hru na kytaru a moderní technologie. Na obrázku 7.2 je podoba reklamy, která se zobrazovala uživatelům. Text sdělení byl omezen na pouhých 90 znaků, takže bylo nutné velmi stručně popsat zajímavé funkce aplikace.

¹Podrobnější informace o cílení mobilní reklamy na facebooku: <https://developers.facebook.com/docs/ads-for-apps/mobile-app-ads>



Obrázek 7.2: Podoba reklamy zobrazované na mobilních zařízeních

Vyhodnocení

Během kampaně bylo reklamní sdělení zobrazeno celkem 4817krát. Celkový počet akcí, které zahrnovaly návštěvu obchodu, nebo jinou interakci s reklamou, se vyšplhal na 103. Kampaň měla tedy míru prokliku (CTR) 1,706%, což je velmi dobrý výsledek v porovnání s obvyklými hodnotami². Ty většinou bývají nižší než 1%. Velký zájem o reklamu přisuzuji dobrému cílení poměrně úzké skupině uživatelů. Průměrná cena za jednu akci nebo 1000 zobrazení byla 2,61 Kč. V aplikaci jsem nenasadil kód potřebný pro vysledování konkrétních instalací díky reklamě, takže tato statistika není dostupná. Předpokládám však, že reklama přivedla přibližně polovinu ze stávajících 46 uživatelů. Tyto hodnoty jsou platné ke dni 17. 5. 2015.

I když považuji tuto formu propagace za relativně efektivní, pokusím se v budoucnu použít spíše způsoby, které jsou zdarma. Plánuji oslovit další bloggery, kteří by mohli mít zájem o vydání recenze aplikace. Dále také budu vkládat propagační příspěvky do různých komunit a diskuzních fór, které se věnují tematice hry na kytaru.

²What is the average CTR, CPC, CPM of Facebook ads? <http://www.ecommercemilo.com/2015/01/facebook-ads-average-ctr-cpc-cpm.html>

Kapitola 8

Závěr

Cílem práce bylo především vyvinout nástroje, které by umožnily úpravu textu písní do podoby vhodné pro mobilní zařízení. Na základě průzkumu současných řešení a návrhu funkcí byla vypracována aplikace, která tento cíl naplňuje. Aplikace má moderní design, je přehledná a má snadné ovládání. Jednotlivé funkce byly vylepšovány na základě zpětné vazby prvních uživatelů, ale i hráčů na hudební nástroje.

8.1 Zhodnocení práce

Největším přínosem aplikace je umožnění pohodlného prohlížení textů písní na mobilním zařízení, protože dosud neexistovalo řešení, které by automaticky přizpůsobovalo texty malým mobilním obrazovkám. Aplikace přináší řadu inovativních funkcí, které značně usnadňují a zrychlují celý proces získání a použití textu písně s akordy. Na základě dosud obdržené zpětné vazby soudím, že je aplikace docela povedená a její funkce řeší skutečné problémy. Budou to však především reální uživatelé, kteří časem ukáží, zda má aplikace nějakou výraznou přidanou hodnotu oproti konkurenčním řešením. Během práce na projektu jsem se zdokonalil zejména v jazyce Java, návrhu uživatelského rozhraní a objevil jsem nejnovější trendy v oblasti programování aplikací pro operační systém Android.

8.2 Další rozvoj aplikace

Další rozvoj aplikace by se měl spíše zaměřit na vylepšování stávajících funkcí, než na masivní přidávání dalších. Hodnota aplikace nebo uživatelského rozhraní je velmi často právě v jednoduchosti, kterou se pokusím zachovat i nadále. Do budoucna plánuji další vylepšování úspěšnosti analýzy a úpravy textů, která se jeví nejdůležitější funkcí aplikace, odlišující ji výrazně od zbytku konkurence. Další prostor pro zlepšení je v možnosti přidání funkce vyhledávání textů hlasem přímo na hodinkách, nebo alespoň umožnění synchronizace oblíbených písní, aby je bylo možné zobrazit i bez použití telefonu. Další vylepšování pro platformu Android wear bude zejména záležet na inovacích a nových možnostech v této oblasti.

Aplikace v současnosti podporuje pouze uživatelsky časovaný automatický posun textů, což nemusí být ve všech situacích praktické. Pokusím se tedy tuto funkci dále vylepšit. Nejspíše by mělo smysl umožnit uživateli i úpravu již nastaveného časování, nyní jej lze pouze přepsat zcela novým. Dále uvažuji o funkci, která by odhadla požadovanou rychlost

posunu pro text na základě skutečné délky hraní písně získané třeba z oficiálního videa na YouTube.

Velký důraz budu klást na zpětnou vazbu uživatelů. Budu sledovat hodnocení aplikace na Google Play Store a přemýšlet nad implementací navrhovaných funkcí. Na základě reakcí se rozhodnu, zda má smysl vytvářet funkce jako je transpozice akordů, nebo zobrazení držení akordů na krku kytary. Uvažuji také o podpoře specifického rozděleného zobrazení pro tablety v režimu na šířku nebo možnosti zobrazit si ve webovém prohlížeči kompletní stránku, ze které byla píseň stažena. Aplikaci bude vhodné také dále propagovat, aby získala určitou základnu uživatelů a objevovala se ve vyhledávání na lepších pozicích. K tomu poslouží například zřízení mikrostránky projektu, nebo další propagace na sociálních sítích.

Pro udržení růstu uživatelské základny bude nutné nadále upravovat design, aby vzhled aplikace odpovídal standardům rozhraní nejnovější verze systému. Do budoucna jsem otevřen novým možnostem a technologiím, a proto nevyklučuji rozšíření aplikace do brýlí Google Glass, nebo jiného zařízení rozšířené, nebo virtuální reality.

Literatura

- [1] Harold, E. R.: *Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX*. Addison-Wesley Professional, 2002, iISBN-13: 978-0201771862.
- [2] Jánský, P.: *Já, písnička 1*. Music Cheb, 1994, iISBN 80-85925-01-X.
- [3] Komatineni, S.; MacLean, D.: *Pro Android 4*. Apress, 2012, iISBN13: 978-1-4302-3930-7.
- [4] WWW stránky: Android market share.
<http://www.idc.com/getdoc.jsp?containerId=prUS25450615>.
- [5] WWW stránky: Android Studio. <https://developer.android.com/sdk/index.html>.
- [6] WWW stránky: AppCompat v21 - Material Design for Pre-Lollipop Devices.
<http://android-developers.blogspot.cz/2014/10/appcompat-v21-material-design-for-pre.html>.
- [7] WWW stránky: Guidline to defining layouts on Android Wear.
<https://developer.android.com/training/wearables/ui/layouts.html>.
- [8] WWW stránky: How users find apps.
http://www.mobiledevhq.com/resources/whitepaper_how_users_find_apps.pdf.
- [9] WWW stránky: Material design guidelines.
<http://www.google.com/design/spec/material-design/introduction.html#>.

Příloha A

Obsah CD

- Instalační soubor aplikace typu APK v adresáři `android_app`
- Zdrojové kódy aplikace ve formě Android Studio projektu v adresáři `android_project`
- Ukázky z aplikace pro telefon i hodinky zahrnující snímky obrazovky a video v adresáři `app_screenshots_videos`
- Zdrojové soubory technické zprávy pro \LaTeX v adresáři `latex`
- Technická zpráva ve formátu PDF v adresáři `thesis`
- Propagační materiály použité v aplikačním obchodě Google Play Store v adresáři `promotional_pictures`
- Stručný návod na přeložení aplikace **na kořenovém adresáři CD**

Příloha B

Plakát

Plakátem prezentující aplikaci je samotný online záznam v obchodu Google Play. Obsahuje samotnou aplikaci, její popis a propagační materiály. Je dostupný na adrese: <https://play.google.com/store/apps/details?id=com.komoi.materialsongbook>.

Moderní Zpěvník
Ondřej Komárek - 13. května 2015
Hudba a zvuk

Instalováno

Tato aplikace je kompatibilní s některými z vašich zařízení.

★★★★☆ (16)

8+1 +18, například Ondřej

Funguje i na Android Wear!

Popis

Vyzkoušejte tento úžasný zpěvník a už nikdy nebudete chtít použít ten papírový! Stahujte vaše oblíbené písně ze třech různých zdrojů, nebo vložte vlastní. Užijte si prohlížení textů se staženým obalem alba v krásném Material designu. Aplikace je zcela zdarma a bez reklam.

Hlavní funkce:

- Material design
- Získejte písničky z internetu, importujte je, nebo sami vložte
- Různá rychlost posunu textu pro každou část písně
- Automatické vylepšení formátování a zobrazení akordů pro mnohem lepší čitelnost
- Obsahuje metronom
- Vyhledejte otevřenou píseň na YouTube jediným dotykem
- Podporuje Android Wear!

Obrázek B.1: Prezentace aplikace v Google Play Store