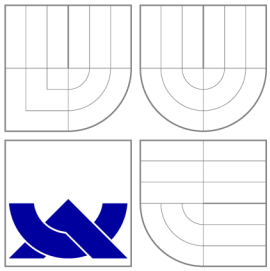
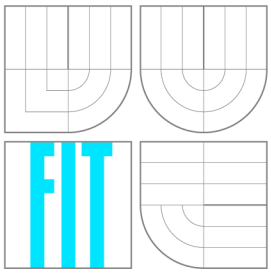


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



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

ANDROID APLIKACE - SLOVNÍK S PŘÍKLADY

ANDROID APPLICATION - DICTIONARY WITH EXAMPLES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LIBOR MAŇÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. IGOR SZŐKE, Ph.D.

BRNO 2016

Zadání diplomové práce/18823/2015/xmanak17

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Maňák Libor, Bc.**
Obor: Management a informační technologie
Téma: **Android aplikace - slovník s příklady**
Android Application - Dictionary with Examples
Kategorie: Softwarové inženýrství

Pokyny:

1. Nastudujte teorii a praxi ke slovníkovým aplikacím, najděte a porovnejte podobné aplikace.
2. Navrhněte a implementujte aplikaci slovníku s příklady. V aplikaci se zaměřte na UX (uživatelskou zkušenost) a oboustranou komunikaci se službou Dicteee.com.
3. Aplikaci otestujte na vhodném vzorku beta-testerů. Zveřejněte aplikaci na Google Play.
4. Pokračujte v implementaci a změnách GUI pro co nejlepší UX. Získejte zpětnou vazbu od uživatelů.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szőke Igor, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato diplomová práce se zabývá popisem vývoje slovníkové aplikace pro zařízení s operačním systémem Android. Aplikace poskytuje uživatelům příklady použití jednotlivých slov a frází v podobě vět, díky kterým může uživatel lépe rozlišit, která překladová varianta se mu hodí. Je vytvořena jako alternativa k webové službě benimsozluk.com, se kterou úzce spolupracuje. V práci jsou popsány teoretické znalosti relevantní k problematice Androidu a vývoji aplikace a dále je zde popsán návrh a celková implementace této aplikace. Čtenář je seznámen s testováním, které bylo provedeno jak za účelem vylepšování aplikace, tak pro její zhodnocení. V závěrečné části jsou uvedeny možnosti dalšího potenciálního vývoje aplikace.

Abstract

This master's thesis deals with development of dictionary application for devices with Android operating system. The application provides examples in form of sentences, which can help users decide what translation variant is the right for them. This application is created as alternative for web service benimsozluk.com and there is cooperation between them. This thesis describes theoretical knowledge related to Android and development of this application. Parts of this thesis are dedicated to design as well as implementation of the application. The reader is introduced to testing conclusions. Testing was done for improving the application as well as for making an assessment of the application final state. In the final part potential future development of the application is drafted.

Klíčová slova

Android, chytrý telefon, tablet, mobilní aplikace, slovník, příklady, uživatelská zkušenost, uživatelské prostředí

Keywords

Android, smartphone, tablet, mobile application, dictionary, examples, user experience, user interface

Citace

MAŇÁK, Libor. *Android aplikace - slovník s příklady.* Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Szőke Igor.

Android aplikace - slovník s příklady

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doktora Igora Szőkeho.
Další informace mi poskytl pan Jan Všianský a pan inženýr Josef Žižka.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Libor Maňák
25. května 2016

Poděkování

Tímto bych rád poděkoval vedoucímu mé diplomové práce panu doktorovi Igoru Szőkemu za jeho odbornou pomoc a cenné rady, které mi poskytl při tvorbě této práce. Dále panu Janu Všianskému a inženýrovi Josefu Žižkovi za všechny připomínky a informace, díky kterým je vytvořená aplikace na mnohem vyšší úrovni.

Obsah

Kapitola 1

Úvod

V dnešní době je znalost jazyků stále důležitější. Dnes už mnohdy nestačí umět pouze jeden cizí jazyk, ale moderní svět vyvíjí na člověka stále větší tlak, aby jazyků ovládal větší množství. Při učení a taktéž během zlepšování nového cizího jazyka lidé využívají mnoho nástrojů a pomůcek, mezi něž patří v první řadě zejména slovníky. Slovníky mohou sloužit jak začátečníkům, tak také zkušeným uživatelům jazyka pro vyhledání neznámého slovíčka. Jelikož existuje velké množství slovníků pro jakékoliv dva konkrétní jazyky, většina lidí ocení, pokud se u překladu mohou rovnou podívat i na ukázky použití daného slova či fráze. Tím se kvalita slovníku zvyšuje a má šanci uspět mezi ostatními.

Rozšířením uživatelů chytrých telefonů mají lidé možnost použít slovníkovou aplikaci nebo webovou stránku a vyhledávat slovíčka přímo pomocí telefonu. Možnosti výběru jsou v takovémto případě ještě mnohem větší a v obchodech s aplikacemi si může uživatel vybrat mnoho slovníků různé kvality, obsahu i ceny.

Cílem mé diplomové práce je navrhnout a implementovat aplikaci na zařízení s operačním systémem Android, která bude sloužit jako slovník s příklady mezi dvěma jazyky. Aplikace by měla mít přehledné a intuitivní uživatelské prostředí a měla by plnit funkce pro pohodlné vyhledávání slov a frází. Operační systém Android jsem zvolil zejména ze dvou důvodů. Jedná se o nejrozšířenější operační systém pro chytré telefony, který ve druhém kvartále roku 2015 vlastní 82,8% světového trhu [1]. Druhým důvodem je velké množství oficiální i komunitní dokumentace a návodů.

Aplikace je vytvořena jako alternativa k webové službě benimsozluk.com, kterou vyvinula brněnská společnost ReplayWell, se kterou jsem spolupracoval při vývoji mé aplikace. Služba v aktuální době nabízí překlad mezi angličtinou a turečtinou a také mezi němčinou a turečtinou. Druhý zmíněný jazykový pár byl přidán v průběhu implementace této práce. Oba jazyky budou k dispozici v mé vyvíjené aplikaci.

Text diplomové práce je rozdělen do sedmi kapitol. Druhá kapitola obsahuje informace o cíli práce, službě benimsozluk a budou zde rovněž představeny již existující slovníkové aplikace, které jsou podobné alespoň v nějaké míře mé aplikaci. Popíši API, které využívá tato služba a se kterým budu pracovat v mé aplikaci. V kapitole 3 popíši návrh aplikace, jak z hlediska funkce, tak uživatelského rozhraní. V následující kapitole bude představena platforma Android, její základní komponenty a práce s databázemi, které jsem využil nejen k ukládání historie vyhledávání, ale zejména pro offline databázi slov a překladových vět. V 5. kapitole popíši implementační část až po zveřejnění celkové aplikace na Google Play. V předposlední kapitole bude popsáno, jak byla aplikace testována a co z testování vyplynulo. Poslední kapitole patří zhodnocení výsledné práce a taktéž zde nastíním, jaký další potenciální vývoj aplikace přichází v úvahu.

Kapitola 2

Slovník v chytrém telefonu

Mobilních aplikací, které nabízejí funkce slovníku, je nepřeberné množství. Většina slovníků pracuje mezi dvěma konkrétními jazyky (například čeština/angličtina) a přeje-li si uživatel použít nový jazykový pár, musí stáhnout zcela jinou aplikaci, čímž se tento počet ještě více zvyšuje. Existující slovníky ovšem nenabízejí jednu zásadní věc, a to jsou příklady použití. Jistě se každému z nás někdy stalo, že po přeložení slova, jsme jednoduše nevěděli, který z výsledků je ten, který momentálně potřebujeme. Tato skutečnost bývá z velké části eliminována v případech, kdy slovník uživateli nabídne praktické příklady použití. Cílem této diplomové práce je tedy vytvoření takové intuitivní a přehledné slovníkové aplikace, která nabídne největší počet příkladů použití a jejíž uživatelský prožitek při jejím užívání bude co nejvyšší. Uživatelé musí mít radost slovník používat, jinak mohou přejít ke konkurenci.

Příklady použití již nabízí služba Benimsozluk, která, jak bylo zmíněno výše, je projektem společnosti ReplayWell, se kterou jsem spolupracoval při vývoji mé aplikace. Tato služba bude představena v kapitole 2.1. Tato společnost mi poskytla API[1], přes které lze získávat příklady použití slov a jejich překladových variant. API společně s offline databází je popsáno v kapitole 2.2. Některé, již existující konkurenční aplikace, jsou popsány v kapitole navazující. U některých zmíněných aplikací jsem se inspiroval a některé mi naopak ukázaly, čeho se vyvarovat.

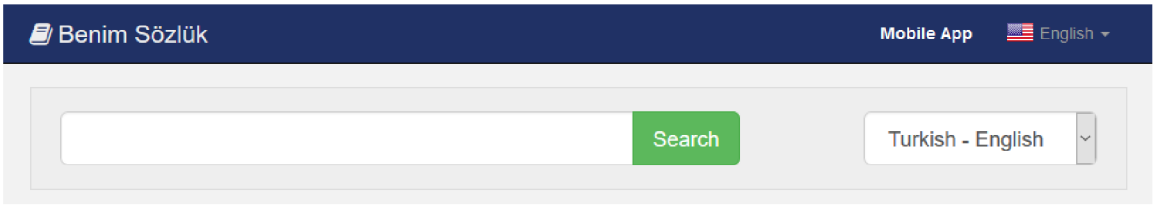
2.1 Benimsozluk.com

Nyní bude představena služba, na které bude postavena má aplikace. Služba Benimsozluk[2] poskytuje online slovník s příklady mezi turečtinou a angličtinou a dále mezi turečtinou a němčinou. Webová stránka služby je lokalizována pro angličtinu a turečtinu a při zvážení jejího jména, jazykových párů a lokalizace lze vypozorovat, že je určena zejména pro tureckého uživatele.

Hlavní stránka Benimsozluk.com (obr. 2.1) zobrazuje textový box pro vepsání hledaného slova či fráze.) a dále umožňuje vybrat směr překladu, ve kterém bude uživatel vyhledávat (tedy 4 směry v případě dvou jazykových párů).

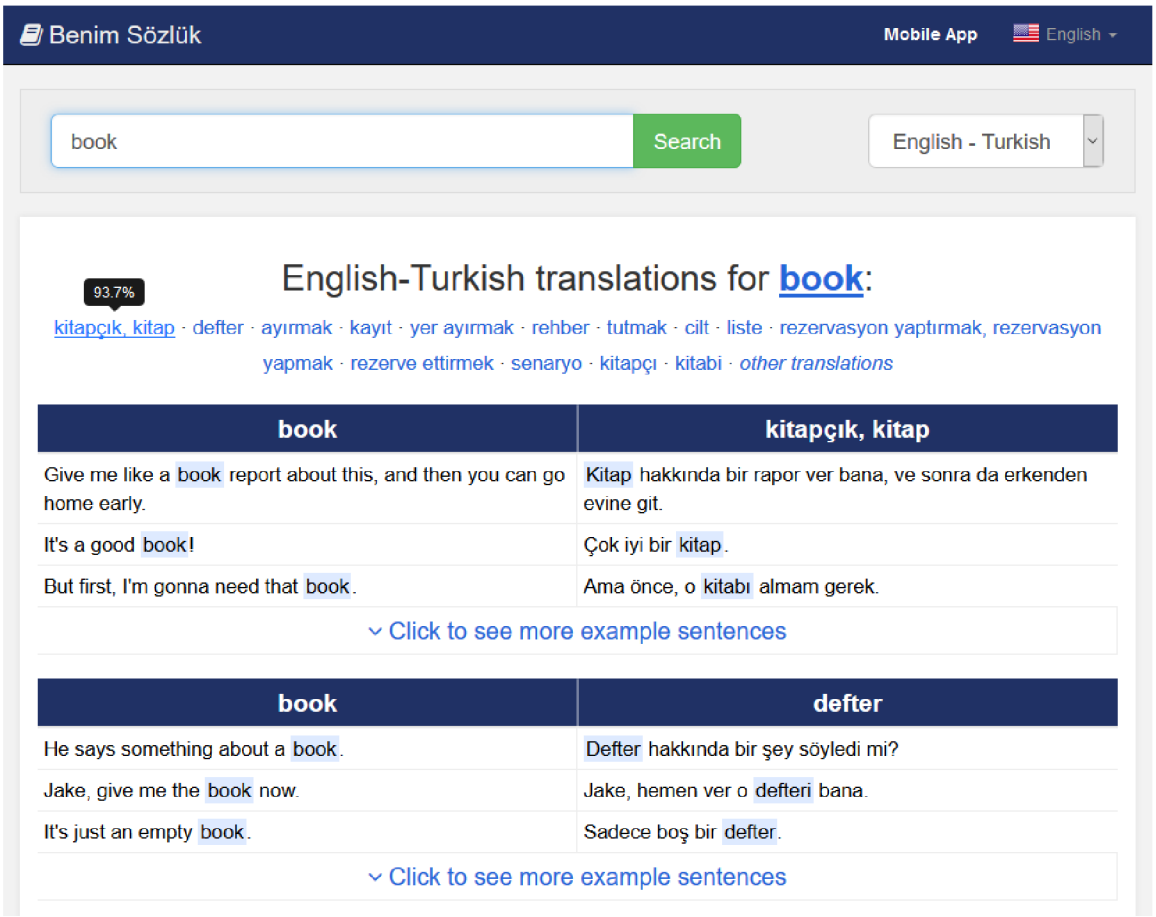
Po zadání hledaného slova či fráze jsou pod textovým polem pro vstup zobrazeny překladové varianty a po najetí ukazatele myši na jakoukoliv překladovou variantu je uživateli zobrazeno procentuální vyjádření, v kolika příkladových větách se slovo vyskytuje. Pokud pro hledané slovo/frázi nejsou nalezeny překladové varianty, je horní část prázdná a jsou

[1] Application Program Interface
[2] http://benimsozluk.com/



Obrázek 2.1: Benimsozluk.com – horní část stránky

pouze zobrazeny příklady použití hledané fráze. V případě, že nejsou nalezeny příklady vůbec, je zobrazen text „not found.“ Na následujícím obrázku je ukázka několika nalezených tureckých překladových variant pro anglické slovo „book“ (česky kniha), včetně zobrazení najetí myší na první překladovou variantu.



Obrázek 2.2: Benimsozluk.com – výsledky hledání

Pod překladovými variantami jsou zobrazeny maximálně tři příklady použití pro jednotlivé varianty překladu. Po stisknutí textu „Click to see more example sentences“ jsou zobrazeny další výsledky (maximálně 10) a po opětovném stisknutí podobného textu je uživatel přesměrován na stránku s příklady jen pro konkrétní překladovou variantu. Ukázka jednotlivých příkladů a překladových variant je ukázána taktéž na obrázku 2.2. Jak zde lze vidět, hledané slovo a jeho varianta je v příkladech zvýrazněna modrým podbarvením textu. Uživatel může na libovolné slovo kliknout, čímž se mu zobrazí výsledky pro kliknuté

slovo. Taktéž může delší frázi označit, čímž se mu zobrazí bublina, na kterou může kliknout pro hledání označené fráze. Vyhledávání probíhá ve směru jazyků na základě toho, v jaké větě byl text označen. Vezměme například obrázek 2.2. Pokud by tedy uživatel označil frázi „bir kitap“ ve druhém příkladě v turecké části, dojde k hledání z turečtiny do angličtiny.

Zobrazí-li si uživatel příklady pro jednu překladovou variantu, může na této stránce prohlížet velký počet příkladů (jsou-li k dispozici). Toto provádí pomocí výběru stránkovacího systému na spodní straně stránky.

2.2 Lingebra API a offline databáze

Pro získání příkladů překladů bude využita Lingebra API, které vrací výsledky ve formátu XML nebo JSON. Struktura dotazu pro získání příkladů použití fráze je následující:

- *phrase* - slovo/fráze, kterou hledáme v překladech,
- *sourceLanguage* - jazyk, ze kterého hledáme (například: english),
- *targetLanguage* - jazyk, do kterého překládáme (například: turkish),
- *limit* - maximální počet navrácených překladů (implicitně: 10),
- *offset* - od jakého výsledku poslat další (implicitně: 0),
- *highlight* - zvýrazni fráze pomocí a (implicitně: 1),
  - 0 - nezvýrazňuj,
  - 1 - zvýrazni,
  - 2 - zvýrazni pouze zadané fráze,
- *format* - výsledek ve formátu xml nebo json (implicitně: xml).

Pokud známe výsledný překlad (například „kitapçık“ pro slovo „book“) lze použít modifikace „phrase“ a to použitím „@english book @turkish kitapçık” pokud výsledky mohou obsahovat tyto fráze v různých tvarech (časování, skloňování atd.) nebo „@english=”book” @turkish=”kitapçık”“ chceme-li výsledky s přesnými shodami. Získání příkladů pro slovo „book“ z angličtiny do turečtiny na základě výše popsaného použití lze pomocí následující adresy: „?phrase=book&sourceLanguage=english&targetLanguage=turkish“.

V aplikaci je také nutné použít databázi pro vyhledávání překladů a pro funkci našeptávače. Tato databáze (kterou využívá také Benimsozluk) má následující strukturu.

- Tabulka 1 - Slova 1. jazyka (například anglické slova). Obsahuje následující sloupce: *id* - číselné označení/primární klíč, *word* - slovo, *all_sentences* - počet vět, kde se slovo nachází, *all_sentences_trans*
- Tabulka 2 - Překlady slov z tabulky 1 (například turecké překlady). Obsahuje následující sloupce: *id* - číselné označení/primární klíč, *id_word* - číselné označení slova z tabulky 1/cizí klíč vázaný na id z tabulky 1, *translation* - slovo (překlad nějakého slova z tabulky 1), *sentence_count* - počet příkladových vět tohoto slova
- Tabulka 3 - Příklady použití slov. Obsahuje následující sloupce: *id* - číselné označení/primární klíč, *id_trans* - číselné označení slova z tabulky 2/cizí klíč vázaný na id z tabulky 2, *source_phrase* - příkladová věta (zdrojový jazyk), *target_phrase* - příkladová věta (jazyk výsledku).

- Tabulky 4 až 6 jsou obdobné tabulkám 1 až 3, s tím rozdílem, že jazyky jsou v opačném směru, tedy tabulka 4 obsahuje turecké slova, tabulka 5 anglické překlady.

Pro výpočet procentuálního vyjádření označující v jak velkém počtu příkladových vět se jednotlivé překladové varianty vyskytují, se využívají hodnoty ze sloupce *all_sentences* v tabulce zdrojového jazyku a hodnoty ze sloupce *sentence_count* z tabulky překladů. Výsledná hodnota se vypočítá jako výsledek dělení *all_sentences* a *sentence_count*.

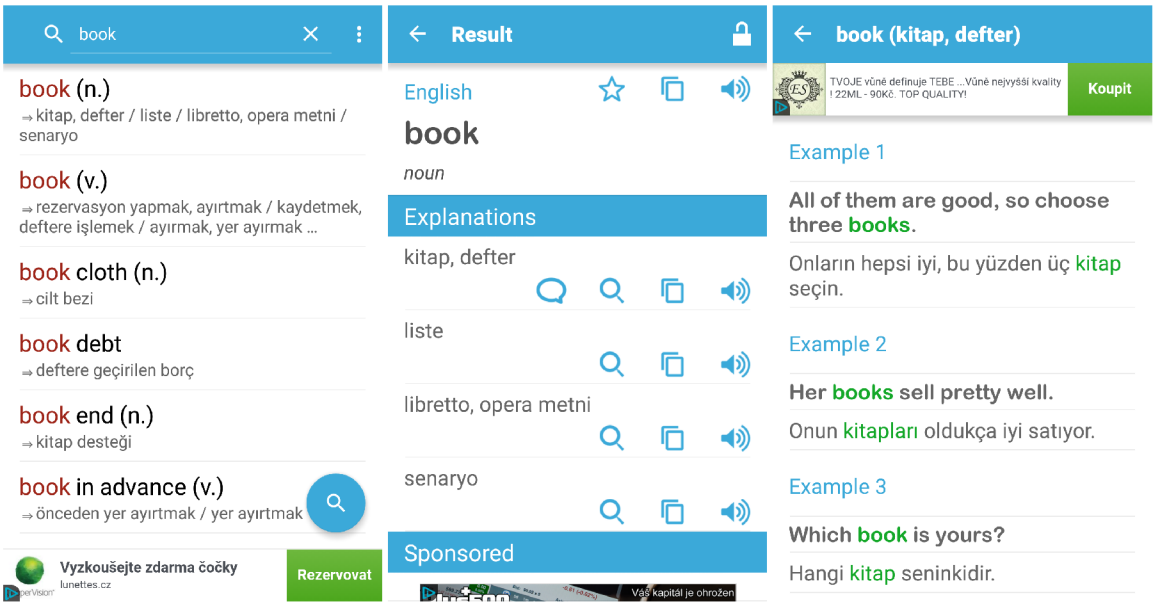
2.3 Existující konkurenční aplikace

Při analýze existujících konkurenčních aplikací jsem se zaměřil zejména na ty, které operují se stejnými jazyky jako služba benimsozluk, a které tedy pracují se stejnými jazyky jako aplikace, která je cílem této práce. Dále mým záměrem bylo najít ty aplikace, které alespoň v určité formě nabízejí příklady použití překládaných slov a frází. Je možné, že nějaká slovníková aplikace mezi jazykem X a jazykem Y příklady použití obsahuje ve větší míře, ale její nalezení z důvodu velkého množství aplikací je relativně složitý úkol. U analyzovaných aplikací bude hodnoceno množství překladových vět, a dále to, zda aplikace poskytují také vyhledávání bez internetu, grafické uživatelské prostředí a další aspekty. Na závěr této kapitoly bude v tabulce 2.1 porovnání určitých informací o zmíněných konkurentech.

Turkish English Dictionary[3]
Jedná se o zřejmě o nejkvalitnější slovníkovou aplikaci, která v určité míře nabízí příklady použití jednotlivých slov a jejich překladů, nicméně neporadí si s delšími frázemi a příklady jsou hlavně pro „důležité“ slovíčka. Funguje kompletně bez použití internetu a má přehledné a dobře ovladatelné uživatelské rozhraní. Nabízí uživatelům ukládání výsledků do záložek, ale přechod k těmto záložkám je zbytečně zdlouhavý a uživatel by tuto funkci díky tomuto faktu mohl brzy přestat využívat. Jednou z její velkých nevýhod je, že první spuštění trvalo téměř 2 minuty, což na starších telefonech může být ještě delší doba. Aplikace obsahuje relativně velké množství reklam, které jsou často zobrazeny přes celou obrazovku, které lze ale je za jednorázový poplatek vypnout. Dalším podstatným nedostatkem je to, že aplikace neumí pracovat v orientaci na šířku, což mnoho uživatelům nemusí vyhovovat. Při jejím testování bylo kladem to, že jsem nezaznamenal žádný výrazný problém či dokonce pád. Na obrázku 2.3 lze vidět našeptávání slov, dále detail vybraného slova a následně seznam příkladů použití.

Sesli Sözlük[4]
Sesli Sözlük je další povedená slovníková aplikace na Google Play s příklady použití, nicméně jejich nabídka je pouze pro známější (jednodušší) slova a příkladová věta je zobrazena pouze jedna pro překladovou variantu. Příkladové věty jsou k dispozici pouze má-li uživatel přístup k internetu, v opačném případě nabízí pouze několik překladových variant pro jednu frázi. Velkou výhodu aplikace vidím v tom, že nabízí uživatelům 20 jazykových párů, které mohou být zapnuty a vypnuty jak si uživatel přeje. Aplikaci je možné používat v pěti různých jazycích, ale při testování byla spuštěna v turečtině, i když moje zařízení bylo nastaveno na češtinu, což by mohlo uživatele odradit, i když změna jazyku je v aplikace možná. Tím, že se aplikace spustila v turečtině, jsem si nemohl přeložit informaci o tom, že po změně jazyku musím aplikaci ručně restartovat. V tomto vidím jeden z velkých implementačních

[3] https://play.google.com/store/apps/details?id=com.bravolang.dictionary.turkish
[4] https://play.google.com/store/apps/details?id=com.seslisozluk



Obrázek 2.3: Ukázka aplikace Turkish English Dictionary

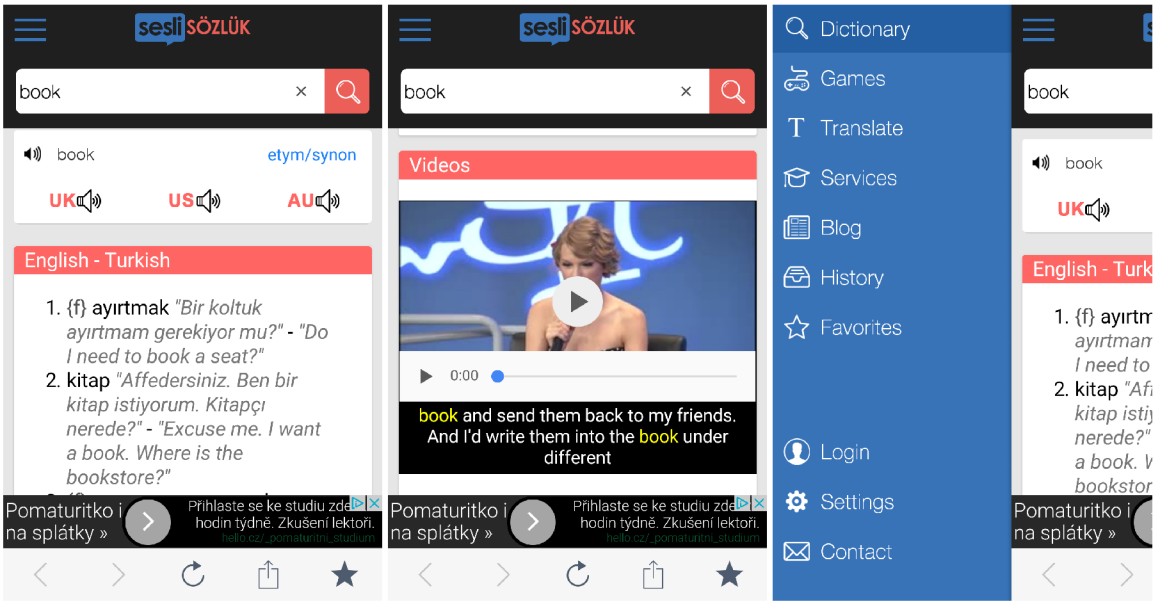
problémů aplikace. Dalším menším problémem bylo nefunkční našeptávání, když jsem aplikaci poprvé spustil. Uživatel má i zde, stejně jako v předchozí aplikaci, možnost uložit si slovíčka pro pozdější použití. Zajímavou funkcí je možnost přehrát si video se slovíčkem, které hledal. Otázkou ovšem je, jak je tato možnost využívána, když si lze výslovnost slova nechat pouze přečíst. Aplikace obsahuje velice diskrétní reklamu a v jejím menu (které je v postranním panelu) jsou odkazy na překladatelské služby nebo na další aplikace, což lze hodnotit negativně, neboť dle mého názoru si nikdo nestahuje tuto aplikaci, aby si nechal zobrazit například nic neříkající Blog.

Free Dict Turkish English[5]

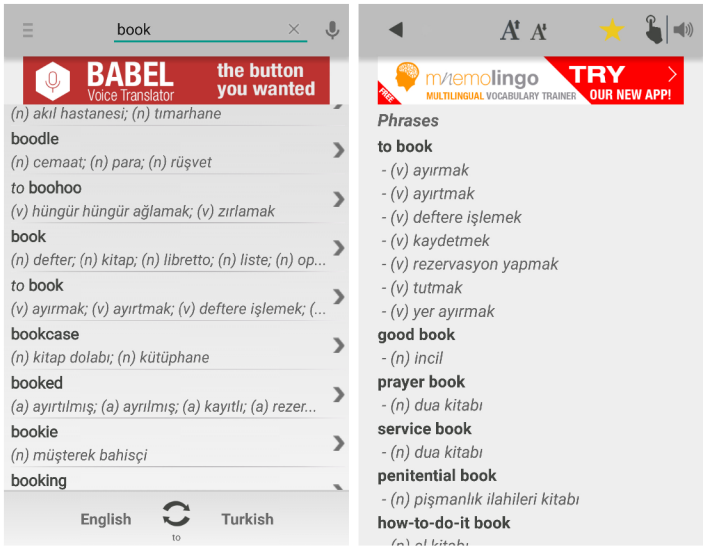
Tato aplikace je dobrým příkladem podobného typu aplikací, které nabízejí spíše než celé věty jen určité menší fráze použití. Tyto fráze jsou nicméně k dispozici pro malý počet hledaných slov. Pracuje bez přístupu na internet a navigace v tomto slovníku je relativně nepřehledná. Například oblíbená slova nebo historie vyhledávání jsou schována a vrátit se zpět k výsledkům z těchto sekcí je možné relativně nepřehledným postupem. Taktéž scrollování ve výsledcích je špatně vyřešeno, jelikož nedojde k vyfiltrování výsledků, ale je zobrazena celá databáze slov a na výsledek je seznam pouze posunut. Pokud se uživatel posune dále od výsledku, pak se již nemůže jednoduše vrátit na výslednou frázi. Aplikace je naplněna rušivou reklamou, která vyskakuje na uživatele v podobě oken na celou obrazovku. Za relativně drahý update (160 korun) dostane uživatel přístup k dalším funkcím, jako je například vyhledávání z výsledků nebo neomezený počet oblíbených položek. Toto je uživateli nabídnuto ihned po prvním spuštění aplikace, aniž by ji měl možnost vyzkoušet. Celkově působí uživatelské prostředí nepřehledně a nejednotně a aplikace se rozhodně nedrží doporučení Googlu pro jednotný vzhled aplikací.

Türkçe-Latince Sözlük[6]

[5] https://play.google.com/store/apps/details?id=com.bitknights.dict.engtur.free
[6] https://play.google.com/store/apps/details?id=com.cloud_inside.mobile.glosbedictionary.latr

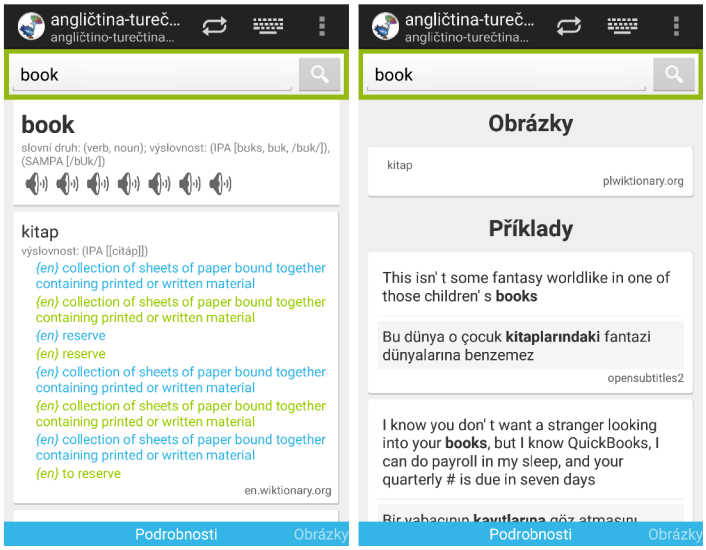


Obrázek 2.4: Ukázka aplikace Sesli Sözlük



Obrázek 2.5: Ukázka aplikace Free Dict Turkish English

Poslední aplikací, kterou zde představím, je anglicko-turecký slovník s názvem Türkçe-Latince Sözlük. Tato aplikace pracuje s online databází s tím, že je možné si ji ručně stáhnout pro offline použití. Aplikace je relativně velká (9,6MB stažení a 29MB po nainstalování), přestože v základu funguje pouze s internetem. Příjemným prvkem byla lokalizace prostředí do češtiny, nicméně čeština je kostrbatá (například „Stáhněte režimu offline databázi nebo připojení k internetu“). Aplikace má celkem nepěkné uživatelské rozhraní, kde například ve výsledcích slova „book“ je zobrazeno 7 stejných tlačítek, které uživatel bez vyzkoušení nemá možnost vědět, co znamenají. Dále ve výsledcích dochází k opakování stejné informace bez žádného zřejmého důvodu, což je vidět na následujícím obrázku. Aplikace dokáže zobrazit příkladové věty, ale zobrazuje je najednou a ne jednotlivě pro překladovou variantu. Tímto dochází k tomu, že příkladové věty jsou přeházeny a dříve mohou být zobrazeny méně důležité překlady. Aplikace nabízí možnost zobrazit si obrázky pro hledaná slova, ovšem při testování tato funkce nefungovala (mohlo jít o výpadek).



Obrázek 2.6: Ukázka aplikace Türkçe-Latince Sözlük

Tabulka 2.1: Dodatečné informace o konkurenčních aplikacích

Aplikace	Vývojář	Počet instalací	Velikost	Průměrné hodnocení
Turkish English Dictionary	Bravolol - Language Learning	500 000–1 000 000	20 MB	4,0
Sesli Sözlük	Sesli Sozluk Ltd.	500 000–1 000 000	16 MB	4,2
Free Dict Turkish English	BitKnights Ltd	100 000–500 000	3,3 MB	4,3
Türkçe-Latince Sözlük	Glosbe Parfieniuk i Stawiński s. j	5 000–10 000	9,6 MB	2,8

Kapitola 3

Návrh

V této kapitole bude popsán návrh mojí aplikace. Popíši, jaké funkce bude obsahovat, jakou strukturu budou mít databáze a jak bude vypadat grafické uživatelské prostředí. Na závěr kapitoly představím některé prvky uživatelského prostředí, které v ní budou použity.

3.1 Funkce

Při návrhu aplikace jsem se zaměřil zejména na to, aby měla minimálně všechny funkce, které obsahuje webová služba benimsozluk.com. V první řadě je tedy důležité, aby uživatel mohl vyhledávat slovíčka a fráze. Toto vyhledávání jsem se rozhodl doplnit o ukládání historie. Tato historie se bude zobrazovat, když uživatel zapne aplikaci a když klikne na textové pole pro zadávání slov. Během psaní v textovém poli uživatel uvidí našeptávač, který bude spolupracovat s offline databází a který mu bude zobrazovat předpokládané slova, která se blíží momentálně vepsané frázi.

Mnoho aplikací, které jsem vyzkoušel během průzkumu trhu, má funkci pro přidávání slov mezi oblíbené položky (někde například nazvané jako záložky). Uvážil jsem, že se jedná o zajímavý nápad a tuto funkci nabídnu také ve svojí aplikaci. Uživatel bude moci ukládat výsledné překlady jak ze seznamu výsledků, tak také z obrazovky příkladů použití.

Další důležitý aspekt výše zmíněné webové služby je možnost zobrazení počtu procent na základě četnosti výskytu slov v příkladových větách pro jednotlivé překladové varianty. Tato funkce je důležitá, aby uživatel viděl, který výsledek je zřejmě ten, který doopravdy hledá, ale stále má možnost vidět i další výsledky a zobrazit si jejich příklady použití. Jak se toto procentuální vyjádření získává je popsáno v kapitole 2.2.

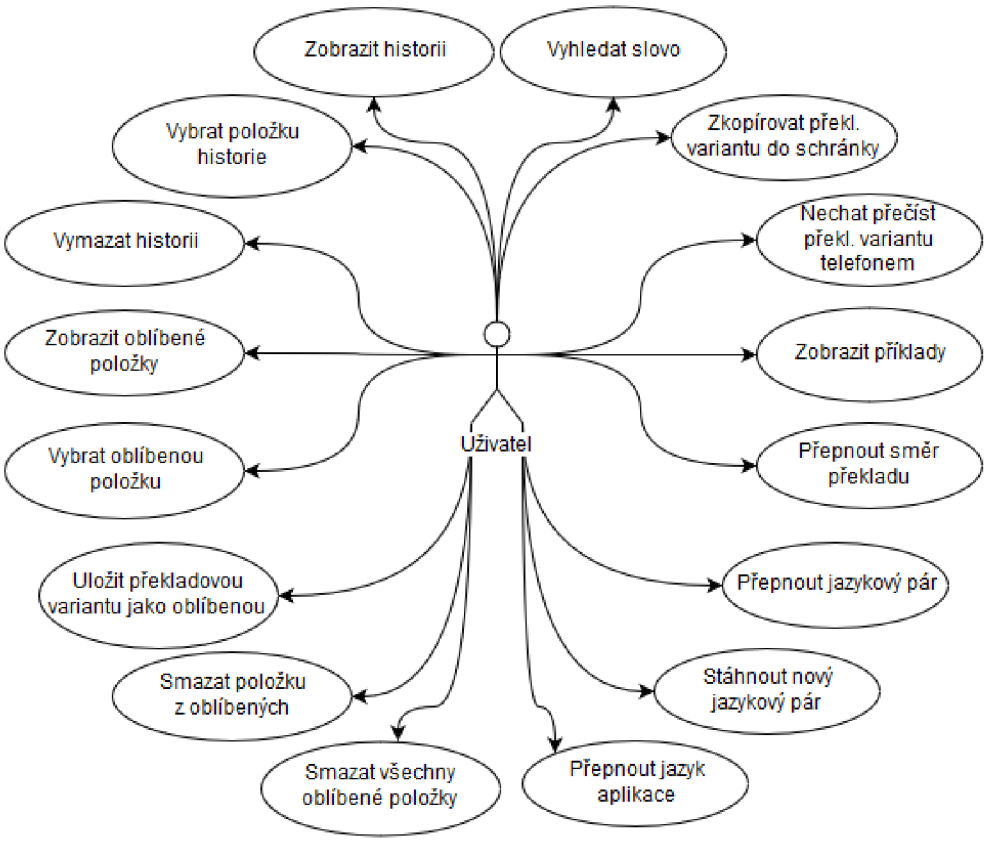
Nyní se dostávám zřejmě k nejdůležitější funkci a to jsou samotné příklady použití slov a frází. Tyto příklady, stejně jako na webové službě, budou zobrazeny jak v seznamu výsledků, tak také pro konkrétní překlad. Během prohlížení výsledků je možné, že uživatel narazí na nové slovo, které nezná a které by si chtěl vyhledat. Aby toto slovo nemusel opisovat, bude mít možnost na něj kliknout a zobrazí se mu výsledky v podobě bubliny (okna, které není na celou obrazovku). Z toho náhledu bude mít možnost přejít na výsledek, zobrazit si všechny výsledky nebo náhled jednoduše zavřít, pokud mu tato rychlá informace stačila. Pokud již existující konkurenční aplikace zobrazují nějaké příklady, pak tuto funkci většina nenabízí vůbec.

Aplikace bude doplněna o několik menších funkcí, které mají za cíl ulehčit uživateli práci s touto aplikací. Jedná se například o možnost zkopírovat si výslednou frázi do schránky a použít ji třeba v emailu. Další funkcí bude možnost nechat si telefonem přečíst překladové

varianty.

Aplikace je vytvořena se záměrem hlavního jazyka turečtiny, s tím že každý slovníkový pár bude tedy sestávat z turečtiny a nějakého dalšího jazyka. Moje aplikace bude dodávána s jedním jazykovým párem turečtina-angličtina zejména z toho důvodu, že většinou uživatelé nebudou ihned potřebovat všechny dostupné jazykové páry, kterých by později mohl být velký počet a stahovaný soubor by byl příliš velký. Toto by nebylo ideální nejen z hlediska čerpání mobilních dat (pokud by aplikaci uživatel stahoval přes mobilní připojení), ale také existencí velkého souboru v mobilním zařízení. Pokud uživatelé budou potřebovat další jazykový pár, jednoduše si ho z prostředí aplikace stáhnou v podobě sqlite databáze.

Diagram případů užití na obrázku 3.1 zobrazuje navržené interakce uživatele s aplikací.



Obrázek 3.1: Diagram případů užití

3.2 Databáze

Jak bylo řečeno v předchozí kapitole, aplikace bude ukládat historii a oblíbené položky. Kvůli těmto funkcím jsem musel navrhnout databázi, kde se budou tyto položky ukládat. Tato databáze se bude sestávat ze dvou tabulek. První tabulka určená pro historii má celkem čtyři sloupce a tabulka pro oblíbené položky sloupců pět. Struktura první resp. druhé tabulky je popsána v tabulkách 3.1 resp. 3.2.

Další použité databáze budou určeny pro dvojici jazykových párů. Tabulky těchto databází musí být uloženy ve zvláštních databázích, protože uživatel nemusí mít naráz všechny

Tabulka 3.1: Schéma a popis sloupců tabulky pro historii

Jméno sloupce	Datový typ	Popis
id	INTEGER, PRIMARY KEY AUTOINCREMENT	Číselné označení položky historie, slouží jako primární klíč.
name	TEXT	Konkrétní slovo nebo fráze, které byla vyhledáváno.
lang	TEXT	Označení jazyku z/ do kterého bylo vyhledáváno.
date	DATETIME DEFAULT CURRENT_TIMESTAMP	Aktuální datum a čas, kdy bylo vyhledávání uskutečněno; tento sloupec je použit, aby se položky zobrazovaly ve správném pořadí, jelikož použití pouze id zde není dostatečné (například uživatel mohl znovu hledat položku s id 1 a položek v tabulce je již například 30).

Tabulka 3.2: Schéma a popis sloupců tabulky pro oblíbené položky

Jméno sloupce	Datový typ	Popis
id	INTEGER, PRIMARY KEY AUTOINCREMENT	Číselné označení položky, kterou si uživatel označil jako oblíbenou, slouží jako primární klíč.
name	TEXT	Konkrétní slovo nebo fráze označené jako oblíbené.
lang	TEXT	Označení jazyku oblíbené fráze.
url	TEXT	Webová adresa pro API dotaz.
date	DATETIME DEFAULT CURRENT_TIMESTAMP	Aktuální datum a čas, kdy bylo označení uskutečněno.

jazykové páry v zařízení, jelikož při nainstalování aplikace bude mít k dispozici pouze pár turečtina-angličtina. Tyto databáze budou mít stejnou strukturu, jaká byla popsána v kapitole 2.2.

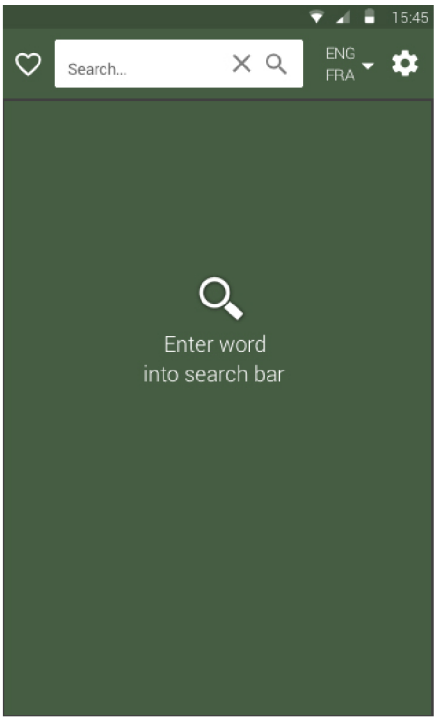
3.3 Uživatelské prostředí

Při návrhu uživatelského rozhraní mojí aplikace jsem vycházel z koncepce Material Design, který Google představil v polovině roku 2014 a na kterém je možné postavit aplikace od verze Android 5.0 (ale je zpětně kompatibilní). Tento design má za cíl vizuálně sjednotit nové aplikace, aby uživatel měl co nejjednotnější zážitek při používání svého zařízení. Součástí této grafické koncepce jsou mimo jiné nové komponenty uživatelského rozhraní, možnost vytvářet stíny, měnit tvary elementů a jednoduše vytvářet nové animace.

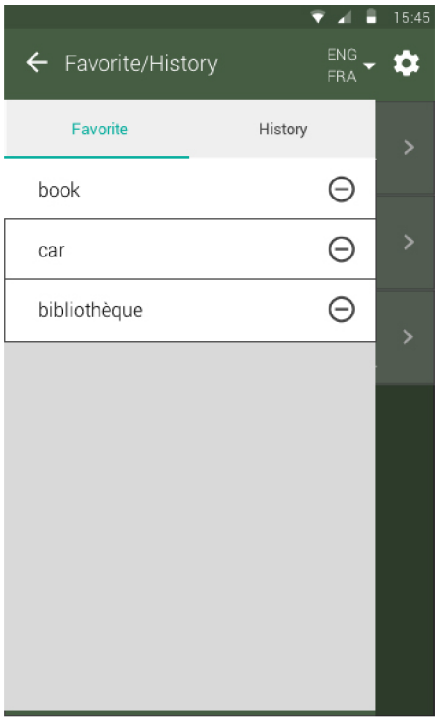
Návrh byl vytvořen jednoduchými mockup obrázky, které reprezentují důležité obra-

zovky mé aplikace. Při prvním spuštění aplikace bude uživateli zobrazen jednoduchý návod, který bude zobrazovat, co jednotlivé elementy grafického uživatelského prostředí mají na starost. Po zavření tohoto návodu jej již znovu neuvidí. Při prvním spuštění, kdy je ještě historie prázdná, bude uživateli zobrazen text informující ho, že vyhledávání slov začne vložením textu do pole pro vyhledávání, které se bude nacházet v Toolbaru (horní liště aplikace). Součástí tohoto elementu bude tlačítko pro rychlé vymazání veškerého textu a také potvrzovací tlačítko pro vyhledávání. Vyhledávání bude možné také započíst kliknutím na „Done/Next“ na softwarové klávesnici při zadávání textu. V neposlední řadě může uživatel kliknout také na jakékoliv slovo v našeptávači pro zobrazení jeho překladu. Horní lišta bude kromě vyhledávacího pole obsahovat na pravé straně také tlačítko pro změnu směru překladu a tlačítko pro zobrazení nastavení. V části této lišty úplně vlevo bude zobrazena ikona pro otevření Navigation Draweru. Navržená hlavní obrazovka je na obrázku 3.2.

Navigation Drawer bude obsahovat historii a oblíbené položky. V horní části budou dvě tlačítka, pomocí kterých bude možné mezi těmito dvěma seznamy položek přepínat. Pod jedním z tlačítek bude vizuální indikátor označující, která sekce je momentálně aktivní. Navigation Drawer bude možné otevřít jak výše zmíněnou ikonou v levé části Toolbaru, tak také pomocí gesta (přejetí prstu z levé strany obrazovky směrem doprava). Zavření uživatel provede buďto kliknutím mimo Navigation Drawer nebo opět gestem (přejetí prstu tentokrát doleva). Klikne-li uživatel na položku historie, bude provedeno vyhledávání s frází, která byla vybrána. Položka oblíbených slov povede rovnou na obrazovku s příklady použití vybraného slova. Oblíbené položky bude možné v elementu Navigation Drawer odstraňovat kliknutím na jediné tlačítko na pravé straně konkrétní položky. Navrhovaná podoba Navigation Drawer je na obrázku 3.3.



Obrázek 3.2: Návrh hlavní obrazovky

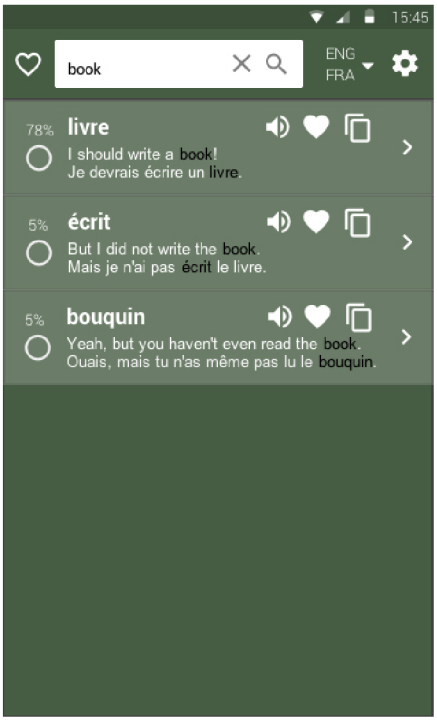


Obrázek 3.3: Návrh Navigation Drawer

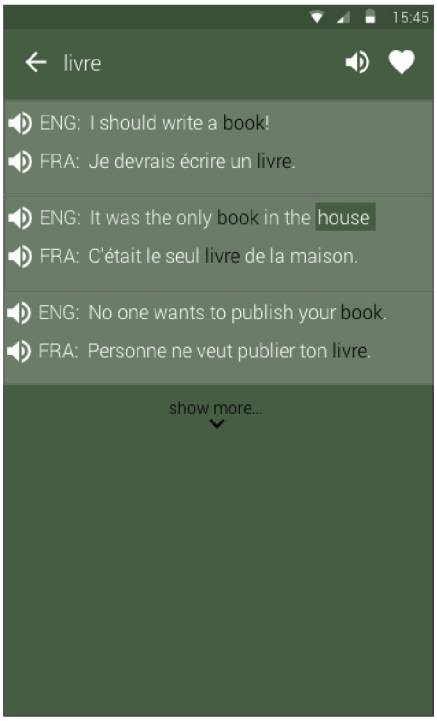
Po potvrzení vyhledávání bude uživateli zobrazen seznam všech překladů. Pokud pro

frázi neexistují konkrétní překladové varianty, bude mu zobrazena obrazovka s příklady použití. Jednotlivé položky v seznamu výsledků překladových variant budou mít na levé straně zobrazenu četnost výskytu v příkladech v přehledné grafické podobě, dále samotný překlad spolu s jedním příkladem použití. Tento jeden příklad zde bude z toho důvodu, že uživateli může často stačit jediný příklad, aby si potvrdil, že nějaký překlad je opravdu ten, který hledá. V příkladových větách bude hledaná fráze a její překlad barevně zvýrazněn. Dále bude mít uživatel možnost překlad zkopírovat do schránky, nechat si ho přečíst a také si ho označit jako oblíbenou frázi. V neposlední řadě může na výsledkovou položku kliknout, což ho zavede na obrazovku s mnoha příklady (pokud jsou k dispozici). Návrh této obrazovky je na obrázku 3.4.

Poslední důležitou obrazovkou je obrazovka pro příklady pro konkrétní výsledek. Zde bude umístěn seznam jednotlivých příkladových vět s možností nechat si celé věty přečíst telefonem. Hledané a výsledné fráze budou opět odlišeny jinou barvou. Uživateli nebudou zobrazeny všechny příklady najednou, ale postupně během scrollování se mu budou načítat další. Pokud bude bez připojení na internet, zobrazí se mu příklady pouze z offline databáze (tedy maximálně 3). Uživatel bude mít možnost podržet prst na libovolném slově a bude mu zobrazena bublina s maximálně dvěma výsledky, na které může kliknout. Návrh této obrazovky je na následujícím obrázku 3.5.



Obrázek 3.4: Návrh zobrazení výsledků

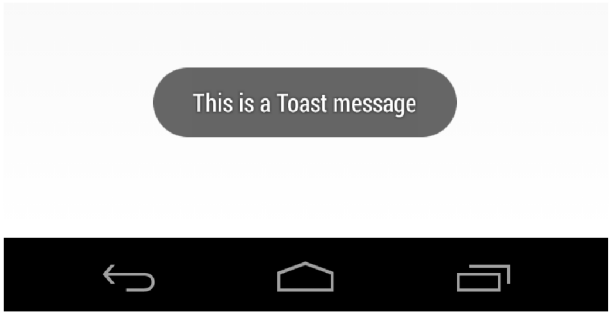


Obrázek 3.5: Návrh obrazovky s příklady

3.4 Další použité komponenty UI

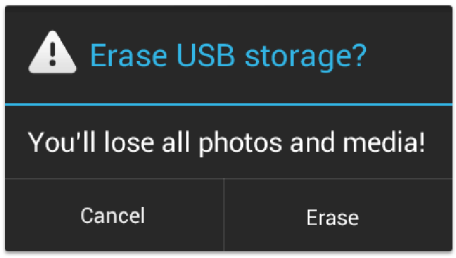
Moje aplikace bude využívat některé důležité komponenty uživatelského rozhraní. Jedním z těchto komponent jsou zprávy Toast, které slouží k informování uživatele o různých akcích nerušivým způsobem. Například tyto zprávy využiji pro informování uživatele, že slovo bylo zkopírováno do schránky nebo že slovo bylo smazáno z oblíbených položek. Toast zprávy

lze zobrazit na různá místa displeje. Na obrázku níže lze vidět Toast zprávu ve spodní části displeje blízko navigačního panelu telefonu. Lze je také zobrazit na různě dlouhou dobu.



Obrázek 3.6: Příklad Toast zprávy[7]

Dalším důležitým prvkem, který využiji, jsou dialogová okna například pro potvrzení vymazání všech položek historie. Tyto dialogy slouží k potvrzení určité akce nebo naopak k informování o akci dodatečné. Dialogové okno zpravidla nevyplňuje celou šířku displeje, ale zobrazuje se v podobě menšího okna na střed obrazovky. V systému Android lze vytvořit několik různých typů dialogových oken. Například se může jednat o dialog, jehož obsahem je výběr různých voleb pomocí přepínacích tlačítek nebo dialog, kde uživatel vybírá datum. Lze také vytvořit dialogové okno s jakýmkoliv obsahem, který lze definovat do klasického XML souboru. V horní části dialogu bývá nadpis a naopak ve spodní části jsou tlačítka pro provedení akce. Tyto tlačítka jsou maximálně tři a každé je označeno jiným typem. První typ je pozitivní tlačítko, které se používá pro potvrzení akce, druhé je označeno jako negativní pro zrušení akce a posledním typem je tlačítko neutrální například použitelné pro „Připomeň mi později.“ Příklad dialogového okna je na následujícím obrázku.



Obrázek 3.7: Příklad dialogového okna[8]

[7]Převzato z: http://www.devactionscript.com/wp-content/uploads/2013/11/toast.png
[8]Převzato z: http://i.stack.imgur.com/qA3d3.png

Kapitola 4

Android aplikace

Vzhledem k tomu, že moje aplikace bude vyvíjena na zařízení s operačním systémem Android, bude nyní tento systém obecně představen. Dále budou popsány základní komponenty, ze kterých se skládá každá aplikace. Budou zde popsány také významné elementy grafického uživatelského prostředí, které budu na základě návrhu z kapitoly 3 potřebovat. Bude popsáno jak na platformě Android ukládat perzistentní data, jelikož v aplikaci bude jak dříve zmíněná databáze slov a jejich překladových variant, tak také se bude ukládat historie a oblíbené položky. Poté bude čtenář seznámen se způsobem, jak lze v aplikaci provádět úlohy ve vláknech. Na závěr této kapitoly bude popsán systém Google Analytics a jeho návaznost na platformu Android. Teoretické informace o Androidu jsou čerpány zejména z oficiální dokumentace [2]. Dále byla použita kniha o Androidu [3] a kniha o intuitivním návrhu aplikací [4].

4.1 Android

Tato podkapitola je převzata z mojí bakalářské práce [5] a upravena, aby odrážela aktuální stav problematiky.

Operační systém Android vlastní od srpna roku 2005 společnost Google a vyvíjí ho konsorcium Open Handset Alliance[9] (skupina 84 technologických a mobilních firem) s cílem nabídnout lepší, rychlejší a levnější mobilní zážitek. Důležité je zmínit, že Android je open source a tedy nabízí všem zdrojový kód nejen k nahlédnutí, ale také k úpravám.

Android je založen na linuxovém jádře a primárně je určen pro mobilní telefony a tablety. V dnešní době je již tento systém k nalezení v mnoha dalších zařízeních, jako jsou například televize, automobily, herní konzole, ale i jinde. Jeho architektura se skládá z pěti vrstev, přičemž všechny tyto vrstvy pracují společně a každá z nich má specifický účel.

Nejnižší (a základní) vrstvou je linuxové jádro s úpravami pro mobilní zařízení (např. kvůli úspoře energie). Tato vrstva se stará o hardwarové operace a proto také obsahuje všechny nutné hardwarové řadiče. Dále je jádro využíváno pro všechny základní úkony jako správa paměti, správa procesů, bezpečností nastavení atd.

Další vrstvou jsou nativní knihovny poskytující další služby aplikacím, které operační jádro již samo neposkytuje. Dále je zde vrstva Android Runtime, která je složena z *Dalvik Virtual Machine* (dále DVM) a *Core Java Libraries* (dále CJL). DVM je optimalizován tak, aby spotřebovával co nejméně energie a neměl velké paměťové nároky. Účelem DVM je běh aplikací, zatímco CJL poskytují většinu funkcionality definované v Java SE knihovnách.

[9] http://www.openhandsetalliance.com/oha_faq.html

Nejdůležitější vrstvou pro mě, jakožto programátora, je Application Framework. V této vrstvě jsou základní bloky, se kterými pracuje moje aplikace. Poslední vrstvou architektury je Applications vrstva. Zde nalezneme všechny aplikace i ty předinstalované na telefonu. Právě do této vrstvy bude nainstalována i moje aplikace.

Nástroje pro vývoj aplikací na platformě Android jsou ke stažení zdarma z vývojové stránky Androidu. Nejjednodušší způsob je stáhnout Android Studio, což je oficiální vývojové prostředí obsahující SDK Androidu. Existují verze pro Windows (32 a 64 bitové verze), Mac OS a Linux. Toto studio také obsahuje manažera SDK, díky kterému je možné zvolit si různé verze vývojových nástrojů. V neposlední řadě Android Studio také podporuje emulaci pro jednodušší vývoj. Druhý způsob pro vývoj aplikací na platformu Android je možný pomocí ADT (Android Developer Tools) pluginu pro vývojové prostředí Eclipse.

4.2 Komponenty aplikace

Tato podkapitola je převzata z mojí bakalářské práce [5] a upravena, aby odrážela aktuální stav problematiky.

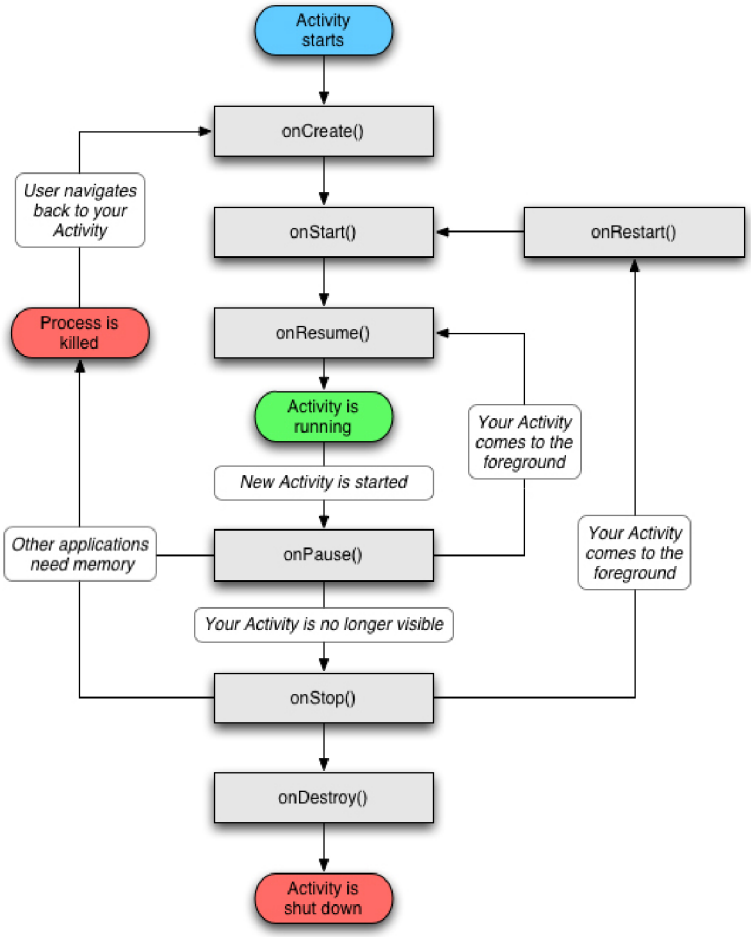
Aplikace na tuto platformu se píší v jazyce Java a Android SDK[10] je kompiluje společně s daty a zdroji do archivního souboru formátu apk. Pomocí tohoto souboru se instaluje program přímo do telefonu. Jakmile je aplikace nainstalována, její chod není ovlivňován okolím a pracuje odděleně od jiných v tzv. sandboxu.

Komponenty aplikace jsou základní stavební prvky aplikace. Každý tento prvek stojí samostatně a každý plní jinou funkci. Existují celkem čtyři typy komponent:

- *Activity*[11] (Aktivita): Aktivita reprezentuje jednu obrazovku aplikace s grafickým rozhraním. Typicky má aplikace jednu hlavní aktivitu, z které může uživatel přejít do dalších aktivit, které jsou mezi sebou volně spojené. Z každé aktivity potom uživatel může přejít do jiné, aby provedl akci, která se v aktuální nenachází. Při přechodu na další aktivitu je aktuální pozastavena a uložena na zásobník, aby se na ni mohl uživatel kdykoliv vrátit. Aktivita má několik *callback* funkcí, v kterých programátor může ovlivnit, co se s aktivitou má stát. Tyto funkce se provádí při změně stavu aktivity jako je vytvoření aktivity, zastavení aktivity, ničení aktivity. Například při zastavení aktivity by měl programátor uvolnit velké objekty, mezi které patří síťové nebo databázové spojení. Při opětovném spuštění se tyto zdroje opět naváží a činnost může pokračovat. Všechny tyto přechody jsou součástí života aktivity, která je pro názornost zobrazena na obrázku 4.1.

  Součástí aktivity mohou být Fragmenty (*Fragments*). Fragment reprezentuje chování nebo část uživatelského rozhraní uvnitř Aktivity. Aktivita může obsahovat větší počet Fragmentů, což je výhodné například u tabletů, kde lze vytvořit obrazovku skládající se z několika částí. Fragmenty mají vlastní životní cyklus a mohou být přidávány a odstraňovány během chodu aktivity. Tento element ovšem musí být vždy součástí nějaké aktivity a je z ní ovlivňován několika způsoby. Například když hostitelská aktivita je pozastavena, fragment je pozastaven taktéž. Fragment lze přidat jako součást layoutu aktivity a žije ve *ViewGroup*. V takovém případě definuje vlastní uživatelské rozhraní. Fragment lze také přidat bez uživatelského rozhraní například pro vykonání

[10] Software Development Kit
[11] http://developer.android.com/guide/components/activities.html
[12] Převzato z: http://androhub.com/wp-content/uploads/2015/04/activity_lifecycle.png



Obrázek 4.1: Životní cyklus aktivity[12]

nějaké činnosti. Toto lze využít například pro ukládání velkého množství dat, když dojde ke změně orientace zařízení.

- *Services*[13] (Služby): Služba je komponenta, která běží na pozadí, a provádí dlouho trvající operace nebo pracuje pro vzdálené procesy. Služby nemají uživatelské prostředí. Služba se může například starat o poslech hudby, zatímco je uživatel v jiné aplikaci, nebo může provádět operace s datovou sítí, aby se neblokovala aktuální aktivita.

- *Content providers* (Poskytovatelé obsahu): Content provider je rozhraní, které se stará o sdílení dat. Programátor si může vybrat, kam data z aplikace uloží, a pomocí Content provideru mohou k těmto datům jiné aplikace přistupovat nebo je měnit. Content provider je také možné využít pro čtení a zápis dat, která jsou určená pouze pro jednu aplikaci.

- *Broadcast receivers*: Broadcast receiver je komponenta, která se stará o přístup k systémovým hlášením (například příchozí SMS, stav nízké baterie, aj.). Programátor taktéž může vytvořit vlastní hlášení, jako například dát vědět ostatním aplikacím, že nějaká činnost byla dokončena.

[13] http://developer.android.com/guide/components/services.html

K aktivaci komponent se využívá asynchronní zpráva zvaná *Intent*, pomocí které lze jednotlivé komponenty navázat na sebe. Při vytváření objektu *Intent* se musí definovat zpráva pro aktivaci specifické komponenty nebo specifického typu komponenty. Unikátním aspektem Androidu je, že jakákoliv aplikace může spustit komponenty jiné aplikace (například za účelem vyfocení fotky z jiné aplikace). Objekt *Intent* se používá pro aktivaci aktivit, služeb a broadcast recieverů, zatímco content provider je aktivován, když přijde požadavek od *ContentResolveru*.

Android poskytuje rozmanitý výběr již vytvořených grafických komponent pro uživatelské rozhraní, mezi něž patří např. layouty, notifikace, dialogy, menu a další grafické komponenty. Grafické rozhraní aktivit aplikace se zapisuje do XML souboru (pokud není vytvořeno dynamicky za chodu aplikace). Já nyní uvedu některé základní grafické komponenty, které budou využity v mé práci:

- *View*[14] (Pohled): Třída *View* představuje základní stavební jednotku pro uživatelské prostředí. View je čtvercová oblast na obrazovce, které se stará o vykreslování a práci s událostmi. Je to rodičovská třída pro widgety, které vytváří další grafické komponenty jako tlačítka či textová pole. Podtřída *ViewGroup* je rodičovská třída pro layouty, což jsou neviditelné komponenty, které uchovávají další pohledy a zajišťují jejich grafické uspořádání na obrazovce.

- *Button*[15] (Tlačítko): Tlačítko je základní grafický prvek, který vyvolá definovanou akci při jeho stisknutí. Může se skládat z textu (*Button*), obrázku (*ImageButton*) nebo obojího (znázorněno na obrázku 4.2). Samozřejmě Android poskytuje další bohaté funkce jako například vytvoření tlačítka bez okraje nebo vytvoření pozadí pro tlačítko. Při stisknutí tlačítka se zavolá metoda, která byla definována v atributu `android:onClick` v elementu `<Button>` patřící stisknutému tlačítku nebo se o tento stisk stará *listener*, pokud byl definován pro toto tlačítko.

Obrázek 4.2: Různé typy tlačítek[16]

- *EditText*[17] (Textové pole): Textové pole slouží uživateli k vepsání nějaké informace. Toto pole může být jednořádkové nebo víceřádkové. Textová pole mohou mít různé typy vstupů, například číslice, data, hesla aj. Taktéž může programátor specifikovat, jaká klávesnice se zobrazí uživateli při zadávání do takovéhoto pole.

- *TextView*[18] (Textový pohled): Jedná se o základní komponentu pro zobrazování textu. Android opět nabízí širou škálu možností tohoto elementu, k těm zajímavějším patří například možnost pro výběr zobrazeného textu.

- *ListView*[19] (Seznam položek): *ListView* je důležitý prvek pro zobrazení scrollovatelného seznamu položek. Jednotlivé položky *ListView* mohou být pouze jednoduché tex-

[14] http://developer.android.com/reference/android/view/View.html
[15] http://developer.android.com/reference/android/view/View.html
[16] Převzato z: http://developer.android.com/images/ui/button-types.png
[17] http://developer.android.com/guide/topics/ui/controls/text.html
[18] http://developer.android.com/reference/android/widget/TextView.html
[19] http://developer.android.com/guide/topics/ui/layout/listview.html

tové pohledy, ale také mnohem komplikovanější pohledy s obrázky a tlačítky. S tímto prvkem se pojí také *Adapter*, což je objekt, který se chová jako jakýsi most mezi *ListView* a daty, které se mají v seznamu položek zobrazit. *Adapter* je zodpovědný za vytvoření *View* pro každý prvek v seznamu. Třídu pro adaptér lze vytvořit pomocí bázové třídy *BaseAdapter*. Kromě konstruktoru je důležité implementovat metodu `getView()`, která vrací View pro jednotlivé prvky seznamu. Takovýto *View* může být založen na vytvořeném XML souboru obsahujícím rozmístění grafických komponent nebo jej lze dynamicky vytvořit. Vytvořený objekt *Adapter* se k objektu *ListView* připojí voláním metody `setAdapter()` s parametrem objektu *Adapter*.

Součástí práce s grafickým prostředím na platformě Android je problematika různých velikostí displeje [6]. Aby aplikace byla vzhledově stejná na všech zařízeních, musí se vytvořit unikátní XML soubory obsahující layout pro všechny velikostní typy obrazovek, které chce programátor podporovat. Tyto soubory se poté vkládají do zvláštních složek pro každou velikost displeje a při pokusu o zobrazení se podle velikosti zobrazovací plochy vybere, který soubor se použije. Druhým způsobem, jak lze zabezpečit stejné zobrazení informací na různých velikostech displeje, je používání jednotek dp a sp. Jednotka dp (*density-independent pixel*) je taková jednotka, která je nezávislá na hustotě pixelů a jednotka sp (*scale-independent pixel*) je podobná jednotka jako dp, ale využívá se pro velikosti písma. Tyto jednotky je tedy lepší využívat pro definicí velikostí jako například velikosti okrajů, šířek elementů a dalších na rozdíl od klasických pixelů (px).

4.3 Ukládání dat

Pro ukládání dat, které je nutné uložit i když dojde k vypnutí zařízení, nabízí operační systém Android hned několik možností. Důležitým faktorem při výběru je to, zda chceme, aby byla data uložena soukromě, tedy aby k ní neměli přístup jiné aplikace, nebo naopak aby byla přístupná i odjinud. Dále neméně důležitý výběr je založen na tom, jak velké množství dat budeme ukládat. Na platformě Android existují tyto možnosti:

- Interní nebo externí úložiště: Při ukládání dat do interního úložiště lze zvolit, jestli chceme, zda soubory uložit privátně nebo zda mohou být čitelné/zapisovatelné jinými aplikacemi. U externího ukládání jsou data vždy čitelné komukoliv.

- *SharedPreferences*: Třída *SharedPreferences* poskytuje rozhraní pro ukládání dvojic klíč-hodnota. Lze uložit primitivní data (čísla, celá i desetinná, řetězce a booleanovské hodnoty). Pro získání objektu této třídy lze použít metodu `getPreferences()` nebo metodu *getSharedPreferences()* pokud je třeba použít více souborů k uložení *Preferences*. Prvním parametrem druhé metody je název souboru, který budeme používat k ukládání/čtení dat. Druhým parametrem (jediným parametrem první metody) je celočíselná hodnota, která představuje, v jakém módu lze data ukládat. Módy jsou stejné, jako při ukládání do interní paměti (viz výše). Data lze ukládat zavoláním metody `edit()` jejíž návratovou hodnotou je objekt *SharedPreferences.Editor*. Poté lze použít metody `putBoolean()`, `putString()` a jiné. Tyto metody mají dva parametry, kdy první je klíč v podobě řetězce a druhým parametrem je ukládaná hodnota. Pro potvrzení změn použijeme metodu `commit()` či `apply()`. Pro čtení dat lze použít metody `getBoolean()`, `getString()` a další. Tento způsob ukládání je vhodný například pro uložení informací o nastavení aplikace.

- Databáze: Android poskytuje podporu pro *SQLite* databáze. Vytvořená databáze je přístupná pouze uvnitř aplikace. Použití databází je vhodné, je-li třeba uložit větší množství dat. Pro práci s databází je vhodné vytvořit podtřídu z *SQLiteOpenHelper*, překrýt metodu `onCreate()` a spustit *SQLite* příkaz pro vytvoření tabulek. Pro zápis, respektive čtení, z databáze je potřeba použít metodu `getWritableDatabase()`, respektive `getReadableDatabase()`, které vrací objekt *SQLiteDatabase*. Poté lze použít metodu `rawQuery()` tohoto objektu pro provedení databázového příkazu připraveného v řetězci. Každý SQL příkaz vrací *Cursor*, který obsahuje všechny nalezené řádky SQL dotazu. Tento výsledek lze procházet mnoha metodami.

4.4 AsyncTask

Android zpracovává všechny příkazy a změny v uživatelském rozhraní v jediném vlákně nazývaném hlavní vlákno. Pokud nechceme, aby se na některé úkoly čekalo a toto hlavní vlákno bylo mezitím blokováno (uživatel nemůže pracovat s aplikací, zatímco čeká na výsledek), musíme použít různých způsobů paralelizace. To je výhodné, chceme-li například získat data z internetu asynchronně a poté uživateli pouze ukázat výsledek. Platforma Android nabízí mnoho technik pro práci s vlákny.

První technikou je využití třídy *Thread* pro vytvoření nového vlákna. Tato třída lze použít dvěma způsoby. Buďto lze vytvořit podtřídu této třídy a překrýt a implementovat metodu `run()`, nebo vytvořit nové vlákno rovnou a poslat *Runnable* konstruktoru. Nové vlákno se vykoná zavoláním metody `start()`.

Druhou technikou je použití asynchronních úloh pomocí *AsyncTask*. Tato třída provede blokující úlohy v pracovním vlákně a výsledky zobrazí ve vlákně, na kterém běží uživatelské rozhraní. Pro použití je třeba vytvořit podtřídu třídy *AsyncTask* a implementovat minimálně metodu `doInBackground()`, ve které proběhnou paralelní úkoly. Další metody, které je vhodné implementovat, jsou metody `onPreExecute()`, `onPostExecute()` a `onProgressUpdate()`. V metodě `onPreExecute()` je vhodné provést přípravné úkoly jako například zobrazení načítací animace. Metoda `onPostExecute()` slouží pro zobrazení výsledků vlákna. `OnProgressUpdate()` metoda může být volána kdykoliv z `doInBackground()` pomocí `publishResult()` a slouží pro zaktuálnění informací ohledně prováděné úlohy (například změna počtu procent dokončení prováděné úlohy). Tyto 3 metody běží ve vlákně uživatelského prostředí. *AsyncTask* lze vyvolat metodou `execute()` s libovolnými parametry, které příjme metoda `doInBackground()`. Běžící *AsyncTask* je možné kdykoliv ukončit z jakéhokoliv vlákna.

4.5 Google Analytics

Google Analytics[20] nabízí vývojářům nástroj pro měření a sledování jak uživatelé používají jejich aplikace a webové stránky. Vývojář může sledovat mnoho aspektů jeho aplikace, které mu mohou pomoct zlepšit aplikaci. Například lze sledovat pády aplikace, jak uživatelé navigují v aplikaci, jaké obrazovky jsou zobrazovány nejčastěji a mnoho dalšího. Pokud aplikace podporuje různé druhy nákupů, analytics může pomoct i v tomto případě. V neposlední řadě si programátor může nastavit speciální události, které se mají zaznamenávat. To může být například kliknutí na tlačítko, mazání položek nebo cokoliv jiného, co si myslí, že stojí

[20] https://www.google.com/analytics/

za to měřit. Díky tomuto měření lze zjistit, jak uživatelé používají aplikaci, které části navštěvují nejčastěji nebo třeba o kterou funkci není velký zájem. Na základě těchto údajů lze aplikaci měnit, vylepšovat a dále zdokonalovat.

Google Analytics lze také používat v aplikaci na systému Android. Nejdříve je třeba přidat povolení aplikaci, aby mohla přistupovat k internetu a číst stav připojení k síti. Dále je třeba přidat pomocnou knihovnu a aplikovat plugin. Konfigurační soubor pro Analytics se získá na stránkách Googlu, kde je nutné zadat několik technických informací o aplikaci, aby měření fungovalo. Tento soubor je ve formátu JSON, který se vloží do složky */app*. Nejlepší způsob jak aplikovat Google Analytics v aplikaci je vytvořit podtřídu třídy *Application* a implementovat pomocnou metodu, která vrací objekt *Tracker*, který je použit pro získávání a ukládání dat a jejich následné zaslání na servery Googlu. Poté lze v aktivitě nebo fragmentu získat tento objekt pomocí metody, která byla implementována ve výše zmíněné podtřídě. Zaslání události lze například pomocí následujícího kódu.

Zdrojový kód 4.1: Google Analytics - zaslání události

```
mTracker.send(new HitBuilder.EventBuilder()
        .setCategory("Action")
        .setAction("Share")
        .build());
```

Kapitola 5

Implementace

Tato kapitola obsahuje zřejmě nejdůležitější část mé diplomové práce. V kapitole 5.1 budou představeny obecné věci o aplikaci, jakým způsobem byla překládána, jakou minimální verzi Androidu podporuje a mnohé další zásadní věci. Dále je v kapitole 5.2 popsána souborová organizace projektu aplikace a v následující kapitole je představeno, z jakých aktivit a fragmentů se aplikace skládá, aby mohla být správně spuštěna jak na chytrém telefonu tak na tabletu. V kapitole 5.4 je popsána implementace databáze, jaké třídy ji tvoří a jaké metody jsem vytvořil. V dalších kapitolách je popsána implementace elementu Navigation Drawer a budou představeny všechny obrazovky, které tvoří moji aplikaci. Na závěr této kapitoly popíši, jak byla aplikace zvěřejněna v oficiálním obchodě Google Play.

5.1 Obecné informace

Aplikace je implementována v programovacím jazyce Java v prostředí Android Studio 1.4.1 a je sestavena pomocí *Gradle*. *Gradle* je systém pro sestavování aplikací, který je určen pro sestavení, testování a vytváření konečných balíčků aplikací. Soubor pro sestavení aplikace má název *build.gradle* a nachází se ve složce *app*. V tomto souboru lze upravovat a konfigurovat celý proces sestavování. Důležité informace pro sestavení jsou v následujícím zdrojovém kódu.

Zdrojový kód 5.1: build.gradle soubor

```
defaultConfig {
    applicationId "com.replaywell.benimsozluk"
    minSdkVersion 14
    targetSdkVersion 23
    versionCode 2
    versionName "1.1"
}
buildTypes {
    release {
        minifyEnabled true
        proguardFiles getDefaultProguardFile
       ('proguard-android.txt'), 'proguard-rules.pro'
    }
}
```

Tento kód v elementu *defaultConfig* obsahuje hlavní informace pro všechny varianty sestavení aplikace. První atribut tohoto elementu je jméno aplikačních balíčků pro unikátní identifikaci. Pod stejným jménem bude potom aplikace vedena v Google Play Storu a po prvním vložení do tohoto obchodu již nemůže být dále měněno. Dalším elementem je minimální verze Androidu, kterou aplikace podporuje, a která je v tomto případě 4.0 (API 14). Aplikace byla cílena na verzi Android 6.0 (API 23), což říká třetí element. `VersionCode` a `versionName` jsou elementy pro označení verze, která bude vytvářena. Chce-li programátor vydat novou verzi v obchodě, musí zvýšit `versionCode` na další celé číslo a typicky také `versionName`.

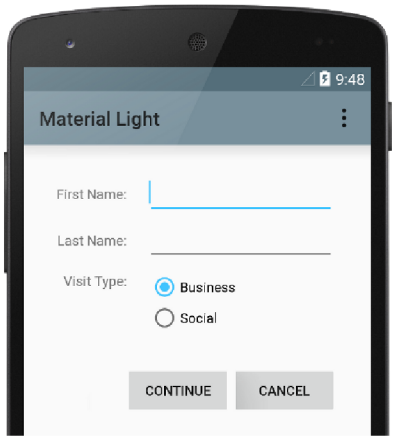
V následujícím elementu *buildTypes* lze specifikovat různé nastavení verzí pro debugging a pro zveřejnění. Element *minifyEnabled* je velice užitečným nastavením, díky kterému lze zmenšit velikost finálních aplikačních souborů odstraněním nepoužitých zdrojů a kódu. Kromě tohoto atributu, atribut *proguardFiles* definuje pravidla pro ProGuard, který pomáhá ochránit aplikaci pomocí obfuskace zdrojového kódu, aby bylo co nejobtížnější „přečíst“, jak aplikace funguje.

Posledním elementem, který jsem využil při tvorbě aplikace, je element *dependencies*, který specifikuje externí knihovny. Kromě dvou základních support knihoven Androidu (*v7 appcompat library* a *Design Support Library*), které Google doporučuje použít kvůli podpoře pro velké množství Android verzí a mimo jiné poskytuje novější API pro starší zařízení, jsem použil tři externí knihovny. První je *SwipeMenuListView*, jejíž použití bude popsáno dále. Další knihovnou je *Android SQLiteAssetHelper*, kterou jsem využil kvůli tomu, že offline databázi slovíček a vět nebudu sestavovat při prvním spuštění aplikace, ale dodám ji jako již existující databázi. Tuto činnost Android ve svém základu neumí, proto jsem využil služeb této knihovny. Poslední závislost je *Google Services plugin* pro funkci Google Analytics.

Dalším důležitým souborem je *AndroidManifest.xml*, ve kterém jsou uvedeny důležité informace pro celkové fungování aplikace. Tyto informace využívá systém Androidu, aby byl schopen aplikaci korektně spustit. Například je zde uvedeno, k jakým zdrojům má aplikace přístup. Moje aplikace momentálně musí mít přístup k internetu, za účelem získávání příkladů použití vět (přes API). Dále musí mít přístup k čtení informace o stavu připojení. Toho je využito, abych mohl v aplikaci zjistit, zda má uživatel přístup k internetu nebo nikoliv. Třetí a momentálně poslední zdroj je přístup k zapisování do externí paměti (SD karta), kde bude mít uživatel možnost přesunout databázi, jelikož se jedná o značně velký soubor. V tomto souboru jsou dále v elementu `<application>` uvedeno atributy umístění ikony aplikace a jméno aplikace. Aplikace je pojmenována stejně jako webová služba, tedy Benim Sözlük (turecky Můj Slovník). Ikona aplikace je na obrázku 5.1.

Obrázek 5.1: Ikona aplikace

Dále je zde atribut pro grafické téma, které aplikace využívá. Grafické téma jsem založil na *Theme.AppCompat.Light.DarkActionBar*, což je předpřipravené téma pro uživatelské rozhraní (zobrazeno na obrázku 5.2), kterému jsem upravil barvy do odstínu modré. V době návrhu webová služba byla barevně velice neutrální, s žádnou výraznou barvou, proto jsem založil návrh na barvě zelené. Barva uživatelského grafického prostředí webové stránky benimsozluk byla nicméně v průběhu implementace mírně změněna do modra, proto jsem nakonec od této zelené barvy upustil, což bylo důležité z hlediska uživatelské přívětivosti. Pokud již uživatel používal slovník na internetu, je důležité, aby měl pocit známého prostředí. Úplně jiná barva by mohla uživatele mást. Stejný problém by mohl nastat, pokud by uživatel nejprve používal aplikaci v chytrém telefonu a poté až na internetu.



Obrázek 5.2: Téma aplikace[21]

Posledním důležitým atributem je atribut name, který specifikuje podtřídu bázové třídy *Application*. Tato podtřída slouží pouze pro poskytování sdíleného objektu *Tracker*, který je využit pro Google Analytics (viz 4.5).

V celém elementu jsou poté vnořeny elementy `<activity>`, které reprezentují jednotlivé aktivity aplikace. Všechny aktivity, které aplikace využívá, zde musí být popsány. Tento element může znovu obsahovat velké množství atributů popisující mnoho funkčních i grafických aspektů. Použil jsem následující elementy:

- *name*, který specifikuje třídu, která implementuje danou aktivitu,
- *theme* popisující grafické téma aktivity,
- *label*, který určuje, co bude zobrazeno v Toolbaru (například: „Settings“),
- *parentActivityName*, který specifikuje nadřazenou aktivitu, která má být spuštěna, když uživatel stiskne tlačítko pro návrat; tento element je podporován od API 16 výše, pro podporu nižších verzí se musí použít element `<meta-data>`, který specifikuje hodnotu pro `"android.support.PARENT_ACTIVITY"`.

Aplikace využívá tři aktivity, které budou popsány v dalších kapitolách. Část manifestu je na následujícím zdrojovém kódu.

[21] Převzato z: https://blog.xamarin.com/wp-content/uploads/2015/01/MaterialDesign.png

Zdrojový kód 5.2: Část elementu application v AndroidManifest.xml

```
<application
   android:allowBackup="true"
   android:icon="@mipmap/ic_launcher"
   android:label="@string/app_name"
   android:supportsRtl="true"
   android:theme="@style/AppTheme"
   android:name="com.replaywell.benimsozluk.MyApplication">

   <activity
      android:name="com.replaywell.benimsozluk.ActivityMain"
      android:theme="@style/AppTheme.NoActionBar" >
       <intent-filter>
          <action android:name="android.intent.action.MAIN" />
          <category android:name="android.intent.category.LAUNCHER" />
       </intent-filter>
   </activity>
       ...
</application>
```

Aplikace je momentálně zaměřena na překlad mezi turečtinou a angličtinou a dále si uživatel může stáhnout databázi pro překlad mezi turečtinou a němčinou. Z tohoto vyplývá, že aplikace je určena zejména pro turecké uživatele, jelikož hlavním jazykem je turečtina a další jazyky jsou určeny pro překlad právě s turečtinou. Přidání podpory pro další jazyky je relativně jednoduchá činnost, ale samozřejmě musí být vytvořena offline databáze a přidána podpora API pro získávání příkladů použití překladových variant.

5.2 Souborová organizace aplikace

Aplikace ve svém kořenovém adresáři obsahuje kromě automaticky vygenerovaných souborů adresář *app*, kde jsou uloženy zdrojové kódy a soubory pro aplikaci, adresář *build* pro výstup sestavení aplikace a adresář *gradle* obsahující *gradle-wrapper* soubory.

V adresáři *app*, je dříve zmíněný soubor *build.gradle*, dále konfigurační JSON soubor pro Google Analytics a další vygenerované soubory. Dále je zde adresář *libs*, který slouží pro knihovny, které aplikace využívá (v případě, že je nelze importovat pomocí *gradle*), a adresář *src/main* obsahující zdrojové soubory. V posledním zmíněném adresáři je soubor *AndroidManifest.xml* popsaný v kapitole 5.1 a dále tyto adresáře:

- *assets/databases*: zde je uložena offline databáze, jedná se o sqlite soubor
- *res/drawable*: zde jsou XML soubory, které obsahují grafický vzhled různých elementů uživatelského prostředí (například okraje tlačítek, pozadí tlačítek, tvary grafických objektů a další)
- *res/drawable-xxx*: v těchto složkách, kde „xxx“ je označení velikosti obrazovky (např. hdpi, mdpi aj.), jsou uloženy bitmapové soubory (PNG, JPG), které aplikace využívá (viz následující obrázek)

Obrázek 5.3: Příklad drawable souborů

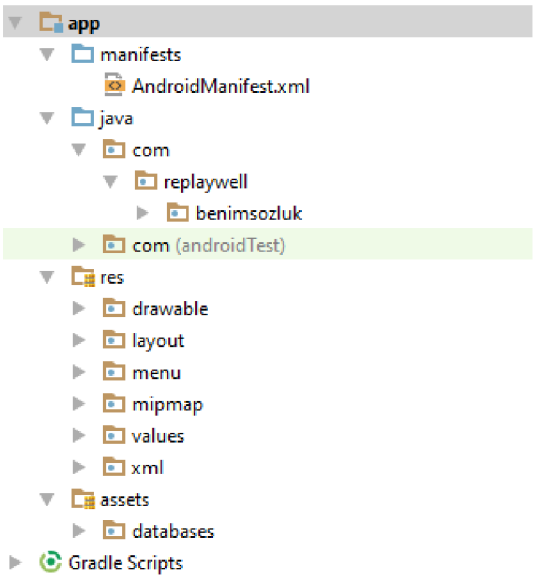
- *res/layout*: tyto XML soubory popisují vzhled všech významných obrazovek, položek seznamů, bublin a dalších elementů, které jsem použil při tvorbě aplikace
- *res/menu*: XML soubory v této složce popisují pořadí tlačítek v toolbaru v jednotlivých aktivitách
- *res/mipmap-xxx*: zde je uložena ikona aplikace, opět „xxx“ je označení pro různé velikosti obrazovek
- *res/values*: v následujících XML souborech jsou uloženy různé data, které je vhodné mít na jednom místě:
  - *array.xml* – definuje typované pole
  - *colors.xml* – definuje barvy specifikované RGB hodnotami a alfa kanálem
  - *dimens.xml* – definuje různé rozměry definované jménem a vzdálenostní jednotkou (například 16dp)
  - *strings.xml* – definuje řetězce uživatelského rozhraní, aby je programátor nemusel definovat ve zdrojových kódech, ale pouze referencoval řetězce z tohoto souboru (buď v Javě pomocí `R.string.jméno` nebo v XML `@string/jméno`)
  - *styles.xml* – definuje formát a vzhled uživatelského rozhraní (například základní téma aplikace, vzhled dialogů aj.)
- *res/values-tr*: stejné jako res/values, ale řetězce jsou v turečtině pro tureckou lokalizaci
- *res/xml*: obsahuje soubor analytics.xml, kde jsou definovány některé globální nastavení pro Google Analytics
- *java/com/replaywell/benimsozluk*: zde jsou uloženy v souborech java všechny třídy aplikace

Stručný popis tříd následuje:

- *ActivityMain*: Třída obsahuje aktivitu hlavní obrazovky
- *ActivityWord*: Třída obsahuje aktivitu pro zobrazení příkladů použití slov a frází
- *ActivitySettings*: Třída obsahuje aktivitu pro nastavení aplikace
- *FragmentXXX*: Tyto třídy obsahují jednotlivé fragmenty
- *AdapterXXX*: Tyto třídy obsahují adaptéry, které se starají o plnění dat různým elementům *ListView* (více viz podkapitola 4.2)
- *DbLangWorkerEN/DbLangWorkerDE*: Třída obsahující funkce pro práci s offline databází slov a překladových variant
- *DbWorker*: Třída obsahující funkce pro práci s druhou databází, kde je uložena historie a oblíbené položky

- *MyApplication*: Třída dědící z bázové třídy *Application*; je zde uložen globální objekt pro Google Analytics
- *ItemXXX*: Třídy entit
- *XmlParser*: V této třídě dochází ke zpracování XML souboru, který aplikace obdrží od Lingebra API
- *CustomEditText*: Třída dědící z klasického *TextView*, překrývající metodu `onKeyPreIme`, kvůli skrývání softwarové klávesnice

Celková organizační struktura aplikace je na následujícím obrázku.



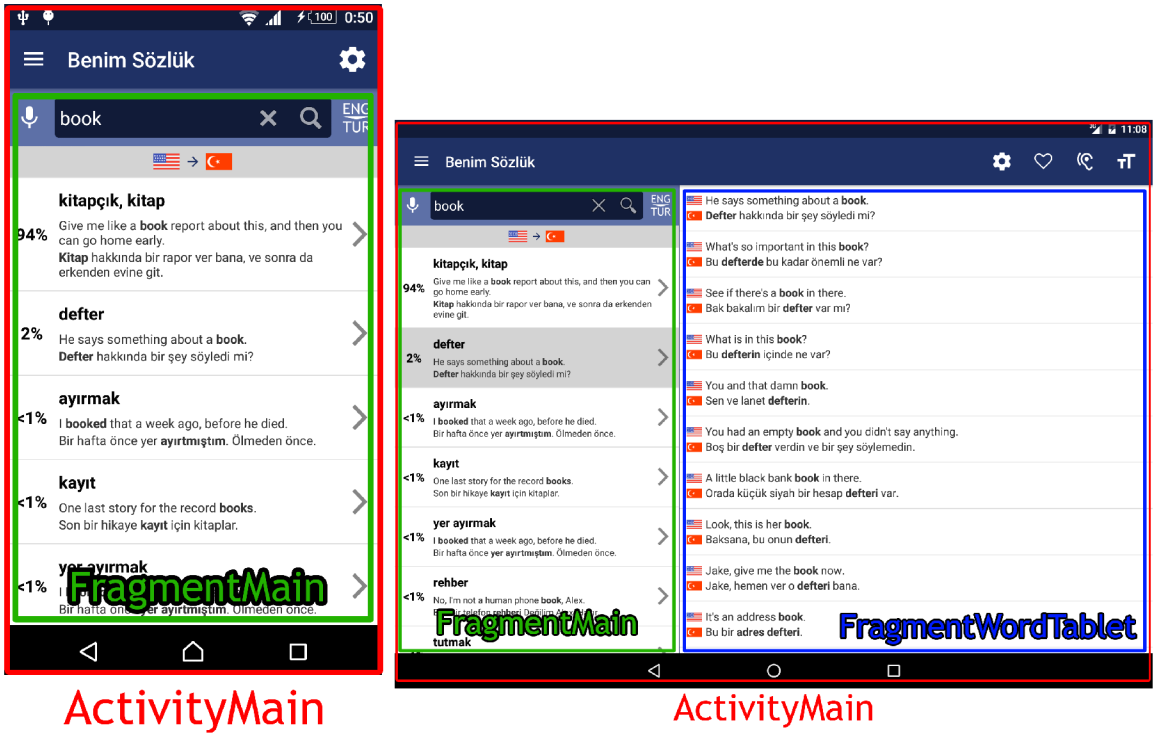
Obrázek 5.4: Organizační struktura aplikace

5.3 Aktivity a fragmenty aplikace

Aplikace byla implementovaná na tři aktivity a tři fragmenty. Jak bylo popsáno v 4.2, fragmenty byly použity zejména kvůli podpoře tabletů. Aktivity a fragmenty aplikace budou popsány detailněji dále, nicméně pro popsání struktury aplikace je nyní stručně popíši.

ActivityMain je hlavní aktivita, která je spuštěna, když uživatel zapne aplikaci. Tato aktivita zároveň ihned vytvoří *FragmentMain*, který obsahuje veškerou grafiku a logiku hlavní obrazovky aplikace. Hlavní aktivita mimo jiné zajišťuje funkci Navigation Draweru, který je určen pro prohlížení a správu oblíbených položek a historie. Druhou aktivitou je *ActivityWord*, která je spuštěna pouze na mobilních telefonech, když si uživatel vybere překladovou variantu, aby si pro ni zobrazil příklady použití. Tato aktivita při své inicializaci vytvoří fragment *FragmentWord*, kde se uživateli budou zobrazovat samotné příklady. Pokud uživatel vybere překladovou variantu na tabletu, nová aktivita se nevytvoří, ale naopak se uživateli zobrazí druhý fragment s názvem *FragmentWordTablet*, který je velice podobný fragmentu *FragmentWord*. Bylo by možné použít tento fragment i u tabletových zařízení, jelikož by ovšem fragmenty neměli komunikovat mezi sebou, ale pouze s aktivitou, ke které patří, bylo nutné vytvořit fragment nový. Poslední aktivitou je aktivita pro

obrazovku s nastavením nazvaná *ActivitySettings*. Tato aktivita je spuštěna, když uživatel klikne na hlavní obrazovce na tlačítko se symbolem ozubeného kola, tedy univerzálního symbolu pro nastavení. Aktivita neobsahuje fragment, jelikož bude jak na mobilních zařízeních, tak na tabletech zobrazena vždy na celou obrazovku a nebude tedy nutné měnit rozměry zobrazovaných elementů. Strukturu aktivit a fragmentů aplikace na mobilních telefonech a tabletech shrnuje obrázek 5.5, kde v levé části je schéma pro mobilní telefony a v pravé pro tablety.



Obrázek 5.5: Schéma aktivit a fragmentů

5.4 Morfologie a databáze

Offline databáze slovíček a jejich překladových variant je ve formě, která byla popsána v kapitole 2.2, jedná se tedy databáze mezi dvěma jazyky obsahující celkem čtyři tabulky plus jednu, která je určena pro tureckou morfologii. Jelikož je turečtina aglutinační jazyk [7], kde se skládají delší slova pomocí přípon, předpon a záložek, je nutné, aby slovník uměl přeložit nejen slovo „kitap“ (česky kniha), ale například i další tvary jako „kitaplar“ (knihy), kitaba (dativ od slova kniha) a další. Morfologii jsem obdržel od společnosti ReplayWell jako textový soubor, kde na každém řádku je morfologická konverze ve tvaru: „kitaplar > kitap.“ Pro tento textový soubor jsem vytvořil jednoduchý program v jazyce Java a převedl jsem ho do tvaru dat pro naplnění sqlite tabulky. Tato pátá tabulka byla vytvořena pomocí třech sloupců a funguje na principu lookup tabulky. Prvním sloupcem je *id* pro číselné označení/primární klíč, dále sloupec *source*, kde jsou uloženy zdrojové tvary slov (z příkladu výše tedy „kitaplar“), a posledním sloupcem je sloupec *target*, kde jsou základní tvary slov, které se dále budou používat ve vyhledávání překladových variant (zde „kitap“). Jak popis napovídá, oba sloupce *source* i *target* jsou datového typu *tinytext*.

Třídy, které jsou určeny pro operace s těmito databázemi, jsou *DbLangWorkerEN* a *DbLangWorkerDE*. První jmenovaná třída je pro slovníkový pár turečtina-angličtina a druhá pro turečtinu-němčinu. Musely být použity třídy dvě, jelikož uživatel druhý jazykový pár nemusí vůbec používat a tedy nikdy se objekt této třídy nemusí vytvořit. Vyhledávání překladu probíhá následovně. Po započetí vyhledávání se zavolá metoda `selectWords` s parametry `String word` a `int language`. První parametr obsahuje konkrétní slovo či frázi, která se bude překládat, a druhý parametr je jazyk, ze kterého resp. do kterého se bude překládat. Aplikace pracuje s následujícím označením jazyků, které jsou využity na mnoha dalších místech aplikace:

- 1 – Překlad z angličtiny do turečtiny,
- 2 – Překlad z turečtiny do angličtiny,
- 3 – Překlad z němčiny do turečtiny,
- 4 – Překlad z turečtiny do němčiny.

Nejprve se v této metodě provede kontrola, zda slovo neobsahuje apostrof. Pokud ano, tento apostrof je nahrazen apostrofy dvěmi, protože apostrofy obklopují celý jeden sqlite dotaz a došlo by k výjimce aplikace. Pokud se vyhledává z angličtiny do turečtiny, aplikace vybere řádek tabulky *entr*, kde sloupci *word* odpovídá hledané slovo. Pokud se překládá naopak z turečtiny do angličtiny, nejprve dojde k pokusu nalézt morfologicky základní slovo. Pokud tedy uživatel chce vyhledat například „kitaplar,“ nalezne se v tabulce *tr_morphology* základní slovo „kitap,“ se kterým se bude dále pokračovat ve vyhledávání. Vyhledání v tomto směru je dále stejné jako ve směru opačném s výjimkou samozřejmě konkrétní tabulky. Je-li slovo nalezeno, získá se jeho *id* a dále se vytvoří a provede další příkaz. V tomto příkazu dojde k vyhledání všech překladových variant, které odpovídají hledanému slovo. Toto se provede v tabulce *entr_trans* (resp. *tren_trans*), kde se vyberou všechny řádky, které odpovídají sloupci *id_word* a identifikátoru, které jsme získali v předcházejícím kroku. Tyto řádky tedy reprezentují překladové varianty. Nyní se projdou všechny řádky a z každého se uloží text překladové varianty a jeho identifikátor do pomocných seznamů. Jelikož u každé překladové varianty je také získán počet příkladových vět, ve kterých se slovo vyskytuje, dochází zároveň k počítání celkového počtu vět napříč všemi překladovými variantami. Jakmile se dokončí tato činnost, posledním krokem je spočítání, v kolika větách z celkového počtu se každá překladová varianta vyskytuje. To provedu tak, že projdu opět celý seznam překladových variant a pro každou variantu vypočítám výsledek jako počet vět varianty děleno celkový počet vět. Na závěr si všechny překladové varianty setřídím od nejvyššího počtu příkladových vět po nejnižší.

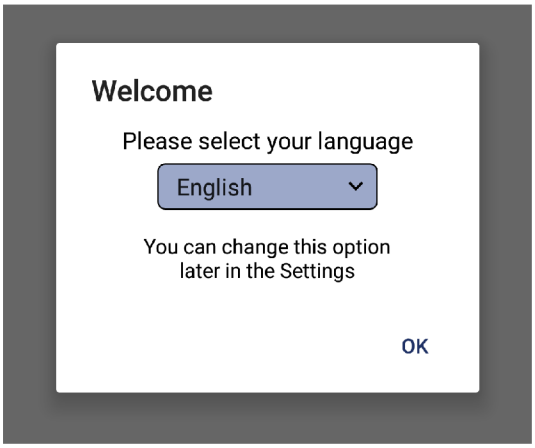
Druhou důležitou metodou těchto tříd je metoda `selectAutocomplete`, která má stejné parametry jako předchozí a je určena pro získání všech slov, která začínají stejně jako první parametr této metody, ve kterém je začátek napsaného slova. Pokud například začne psát do vstupního textového pole „kit,“ dojde k nalezené všech slov, které tak začínají („kitabe“, „kitabevi“ a mnoho dalších).

Poslední databází, kterou si moje aplikace sama vytváří a pracuje s ní, je databáze pro ukládání historie a oblíbených položek. Struktura této databáze byla popsána v kapitole 3.2. Třída pro tuto databázi má název *DbWorker*. Součástí této třídy jsou metody pro vytváření nových řádků, aktualizaci či smazání existujících řádků a metoda pro smazání celého obsahu tabulek.

5.5 První spuštění a jazyky rozhraní

Před prvním výběrem Android systém nastaví jazyk aplikace podle jazyku systému, takže pokud uživatel používá své zařízení v turečtině, má aplikace bude také v tomto jazyku spuštěna. Při používání jakéhokoliv jiného jazyku je aplikace od základu v angličtině. Při prvním spuštění aplikace je uživatel vyzván pomocí dialogu (obrázek 5.6), aby si zvolil jazyk rozhraní, který chce používat. Aplikace byla implementována tak, aby podporovala více jazyků. Prvním mezinárodním jazykem je angličtina, a protože je aplikace z velké části určena pro turecké uživatele, je možné ji používat také v turečtině. Toho je dosaženo tak, že při výběru (nebo změně v Nastavení) jsou změněny zdroje aplikace pomocí třídy *Resources* a do *SharedPreferences* je výběr jazyka uložen. Při každé změně aktivity je ovšem nutné, aby tato hodnota byla vždy přečtena a zdroje opět změněny. Tato činnost se provádí ze všeho nejdříve, aby byly aktivity a fragmenty načteny ve správném jazyku.

Spustí-li uživatel aplikaci s novou verzí (například při přechodu z verze 1.0 na 1.1) je uživateli zobrazen dialog s poznámkami, co je v aplikaci nového, aby byl upozorněn na novinky a změny, které mu pomohou a potenciálně zlepší prožitek v užívání mé aplikace.



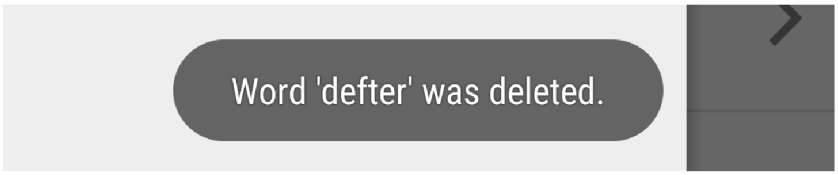
Obrázek 5.6: Dialog při první spuštění aplikace

5.6 Navigation Drawer

Navigation Drawer v mé aplikaci slouží k zobrazení historie vyhledávání a oblíbených položek. Tento prvek uživatelského rozhraní je přístupný z hlavní obrazovky. Uživatel jej otevře buďto gestem, nebo ikonou, nacházející se v levé horní části obrazovky (v Toolbaru). Zobrazuje se posledních 20 položek z historie a všechny oblíbené položky, které si uživatel uložil. Historie i oblíbené položky jsou uloženy v databázi (viz kapitola 5.4) a při každém otevření Navigation Draweru se znovu načtou do *ArrayListu*, jelikož mohlo dojít ke změně od doby prvního načtení, které se provádí při načtení hlavní aktivity. Položky jsou uloženy ve dvou seznamech, mezi kterými může uživatel přecházet kliknutím na korespondující tlačítka v horní části Navigation Drawer. Tyto tlačítka indikují aktuální zobrazovaný seznam. Tento indikátor je implementován jako speciální okraj tlačítka, který se přidá při stisknutí tlačítka a je naopak odebrán z tlačítka druhého. To je nutné, aby uživatel vždy věděl bez přemýšlení nebo studování položek, jaký seznam má zobrazen. Mohlo se třeba stát, že uživatel si zobrazil Navigation Drawer a v otevřené podobě aplikaci opustil, aby se věnoval

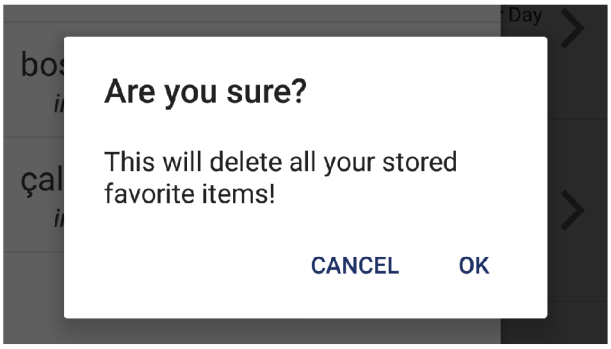
něčemu jinému. Po návratu do aplikace díky indikátoru ihned zjistí, zda měl zobrazenou historii nebo oblíbené položky.

U jednotlivých oblíbených položek bylo implementováno jejich smazání. Smazat položku lze pomocí tlačítka s ikonou křížku. Po stisknutí tohoto tlačítka dojde ke smazání dané položky a uživatel je informován o této skutečnosti pomocí krátké Toast zprávy (obrázek 5.7), jelikož je důležité z pohledu UX, aby uživatel věděl, zda akce, kterou provedl, byla úspěšná či nikoliv. Další negativní okolnost by mohla nastat, pokud by uživatel měl v seznamu velké množství položek, při jejichž odstraňování by se mohlo stát, že si nevšimne, zda došlo k úspěšnému provedení této funkce.



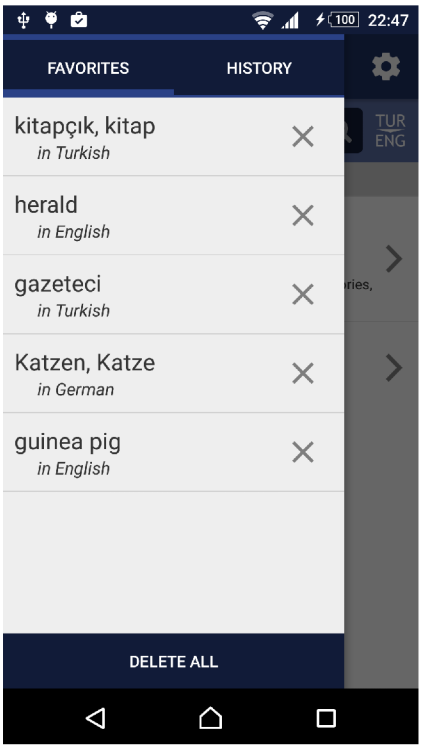
Obrázek 5.7: Informační zpráva po smazání oblíbené položky

Mazání položek historie, ve které, i po krátké době používání, může být mnoho desítek hledaných slov a frází by bylo velice zdlouhavé, kdyby je uživatel musel mazat po jednom. Pro jednoduchost jsem tedy implementoval tlačítko, které po jeho stisknutí vymaže všechny položky historie či oblíbených slov. Předtím ovšem, než se tak stane, ukážu uživateli potvrzující dialog, zda si opravdu přeje všechny položky vymazat. Z hlediska uživatelského prožitku je takovýto potvrzující dialog důležitý, aby nedošlo k vymazání obsahu omylem. Mazání lze tedy ještě včas zrušit kliknutím na tlačítko „Cancel“ nebo jednoduše kliknutím mimo zobrazený dialog, který je zobrazen na následujícím obrázku.



Obrázek 5.8: Potvrzovací dialog pro mazání všech položek

Seznamy jsou implementovány pomocí dvou *ListView* a o jejich zobrazování se starají adaptéry *AdapterDrawerFavorites* a *AdapterDrawerHistory*. Vybere-li uživatel položku historie, aplikace začne vyhledávat toto slovo, což bude podrobněji popsáno v kapitole 5.7.1. V případě, že uživatel zvolí oblíbenou položku, je uživateli zobrazena obrazovka s příklady, která bude taktéž popsána v detailu v podkapitole 5.8. Podoba Navigation Drawer je na následujícím obrázku 5.9, kde v horní části jsou zmiňované tlačítka pro přepínání mezi historií a oblíbenými položkami společně s indikátorem aktuálně zobrazeného seznamu. Dole je potom tlačítko „Delete all“ pro smazání všech položek.



Obrázek 5.9: Navigation Drawer

5.7 Hlavní obrazovka

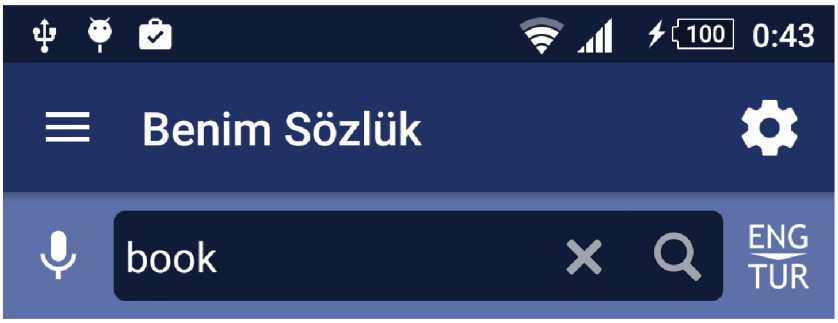
Hlavní obrazovka aplikace je implementována dvěma způsoby na základě toho, zda je spuštěna na chytrém telefonu (šířka displeje pod 600dp), nebo na tabletu (displeje větší než 600dp). Nejprve bude popsána implementace na menších displejích chytrých telefonů a poté na tabletech.

5.7.1 Chytré telefony

Na chytrých telefonech je hlavní obrazovka implementována pomocí hlavní aktivity a jednoho fragmentu. Tento fragment je inicializován společně při startu aktivity. Aktivita má za úkol zejména správu důležitých proměnných, jako jsou například aktuální směr překladu či jazyk výsledků. Tyto proměnné je nutné ukládat pomocí tzv. *Bundle*, změní-li uživatel orientaci svého zařízení. Jelikož systém Androidu pracuje tak, že aktuální aktivita je restartována, došlo by ke ztrátě hodnot těchto proměnných a aplikace by nefungovala korektně. Proto jsou tyto hodnoty před restartem uloženy a v zápětí při novém vytvoření znovu z *Bundle* načteny. Po takovémto restartu se také nevytváří nový fragment, jelikož jeden již existuje a ten je pouze obnoven. Hlavní aktivita má také za úkol veškerou správu elementu Navigation Drawer, který byl popsán v předchozí kapitole. Veškerá další logika je součástí onoho jediného fragmentu.

Hlavní obrazovka obsahuje Toolbar v horní části obrazovky, kde v levé části je ikona pro zobrazení Navigation Drawer, v podobě tří krátkých horizontálních čar. Tato ikona je standardizována právě pro tuto činnost a někdy se jí přezdívá hamburger ikona, jelikož připomíná právě hamburger. Vlevo od ní je zobrazeno jméno aplikace. V dřívějších stádiích implementace zde bylo textové vstupní pole pro zadávání slov a frází, nicméně z marke-

tingových důvodů, aby uživatel měl stále na očích jméno aplikace, bylo toto pole posunuto pod Toolbar. V úplně pravé části Toolbaru je ikona ozubeného kola pro zobrazení nastavení aplikace. Tuto horní část hlavní obrazovky zobrazuje obrázek 5.10. Dále je na tomto obrázku zobrazena lišta pro vyhledávání, které je popsána v zápětí.

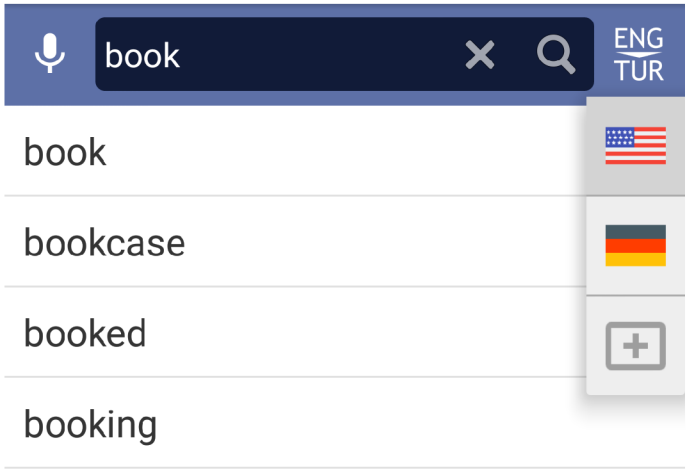


Obrázek 5.10: Horní část hlavní obrazovky

V této liště uživatel nalezne výše zmiňovaný *EditText* pro zadávání slov a frází. Na levé straně od pole pro vstup je tlačítko se symbolem mikrofonu, když si uživatel přeje zadat slovo nebo větu hlasovou formou. Po stisknutí toho tlačítka je vytvořen nový *Intent* pro vytvoření akce, kde se vestavěná funkce Androidu pro rozpoznání hlasu postará o zpracování řeči. Tento *Intent* po spuštění metodou `startActivityForResult()` vrátí seznam, kde prvním prvek tohoto seznamu je řečený text v podobě řetězce. Na pravé straně od vstupního pole jsou dvě tlačítka, která jsem vytvořil tak, aby vzhledově zapadala ke vstupnímu poli. První tlačítko je určeno pro rychlé vymazání veškerého textu, který je ve vstupním poli momentálně obsažen pro zlepšení UX. Dále je zde tlačítko pro potvrzení vyhledávání. Vyhledávání může uživatel spustit ještě dvěma dalšími způsoby. První je kliknutím na potvrzovací tlačítko na softwarové klávesnici a druhý způsob je vybrání nějakého slova z našeptávání. Našeptávání je zobrazeno uživateli pod vstupním polem a zobrazují se ty slova, které začínají stejně jako zatím napsaný vstup. Uživateli se zobrazují slova v jazyce, který má momentálně nastaven. Pokud je momentální překlad z angličtiny do turečtiny a uživatel začne vepisovat písmena do textového pole, našeptávání bude zobrazovat anglická slova. Tento našeptávač je implementován jako jednoduchý *ListView* a postupně se z databáze při jakékoliv změně ve vstupním poli získají nová slova. Změna může být vyvolána vepsáním/smazáním písmena nebo zkopírováním textu do vstupního pole. Našeptávání může výrazně pomoct s používáním aplikace, jelikož uživatel nemusí vypisovat celé slovo. Také mu může pomoct, když neví, nebo si není jistý s přesným pravopisem některého slova. Klikne-li uživatel do vstupního pole a předtím než něco začne psát, je mu zobrazena historie vyhledávání ve stejné podobě jako našeptávání. Opět může jakoukoliv položku historie vybrat a vyhledávání započne.

V úplně pravé části lišty je poslední tlačítko, které slouží ke změně směru překladu. Jakmile jej uživatel stiskne, změní se podoba tlačítka, aby odpovídalo novému směru, a zároveň se do *SharedPreferences* uloží aktuální směr překladu. Uložení je výhodné z hlediska UX, kdy v případě, že uživatel aplikaci ukončí a poté ji znovu zapne, nastavím směr překladu na ten, který uživatel naposledy používal. Tímto nebude uživatel zmaten směrem překladu, pokud má stále v paměti, v jakém vyhledával. Navíc toto tlačítko má implementován *listener* pro dlouhý stisk, kdy jej uživatel může na krátký čas podržet a zobrazí se mu nabídka nainstalovaných jazyků v podobě menšího elementu *Spinner*. Nemá-li uživatel stažen jazykový pár pro turečtinu-němčinu, zobrazí se mu pouze americká vlajka. V případě,

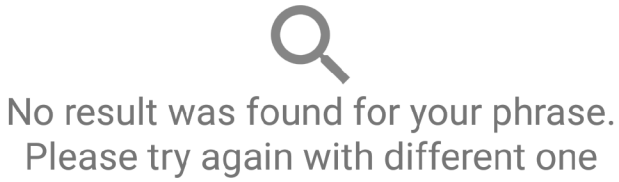
že tento pár stažený má, je navíc zobrazena německá vlajka. Kliknutím na jakoukoliv vlajku dojde ke změně hlavního překladového jazyku. Například měl-li uživatel zvolenu angličtinu a klikne na německou vlajku, bude nyní vyhledávat mezi němčinou a turečtinou. Poslední položkou ve *Spinneru* je ikona se symbolem plusu, kdy po její vybrání je uživateli zobrazena obrazovka pro nastavení, kde může stáhnout nový jazykový pár (tato obrazovka bude popsána dále v textu). Aktuální jazyková pár je zvýrazněn tmavší šedou barvou, aby uživatel jasně viděl, který pár je aktivní. Používání našeptávání a výběr překladového jazyku je zobrazeno na následujícím obrázku.



Obrázek 5.11: Našeptávání vyhledávání a výběr jazyku

Jakmile uživatel potvrdí vyhledávání, hledané slovo či fráze je uložena do databáze historie vyhledávání a také začne vykonávání asynchronní úlohy, která má mnoho menších úkolů. Nejprve dojde k zobrazení načítací animace, aby uživatel věděl, že „se něco děje.“ Aplikace si dále uloží v jakém směru se vyhledává, což je důležité, abych věděl, v jakém jazyce jsou výsledky pro další činnosti. Poté dojde k pokusu o vyhledání slova v databázi. Je-li slovo nalezeno, dojde také k pokusu o vyhledání ve směru opačném. K tomuto dojde i v případě, že slovo v původním směru nalezeno není. Třída, která má na starost vyhledávání v databázi a která byla popsána výše v textu, vrátí seznam překladových variant. Má-li uživatel přístup k internetu, dojde pro tyto varianty postupně pomocí API dotazů k získání jednoho příkladu použití. Pokud vyhledávání v databázi selže, znamená to, že slovíčko není v databázi, a dojde k získání příkladů použití. Toto nastane téměř vždy, hledá-li uživatel frázi nebo delší slovní spojení, jelikož těchto v databázi není mnoho. Nyní se uživateli zobrazí výsledek vyhledávání. Z hlediska User Experience je důležité, aby vždy uživatel ihned viděl výsledek vyhledávání. Není-li například vůbec nic nalezeno, je zobrazen text, který jej informuje, že pro hledanou frázi nebylo nic nalezeno a ať se tedy pokusí opakovat hledání s frází jinou (obrázek 5.12), nebo pokud uživatel hledal nějako slovo kupříkladu z angličtiny do turečtiny, ale překlad byl nalezen v opačném směru (může se stát, zapomene-li uživatel před vyhledáváním přepnout směr překladu), je nutné jej o této skutečnosti informovat.

Nejčastějším výsledkem vyhledávání je ovšem zobrazení výsledků. Obrázek 5.13 zobrazuje několik výsledků překladu anglického slova „book“ do turečtiny. Zobrazení výsledků je implementováno pomocí speciálního *ListView* z knihovny *SwipeMenuListView*, kdy jednotlivé řádky tohoto elementu obsahují jednu překladovou variantu. V levé části položky je zobrazení procent, které ukazují v jak velkém počtu vět je daná varianta v poměru



Obrázek 5.12: Nenalezený překlad slova/fráze

k ostatním. Dále je tučným písmem uvedena samotná překladová varianta společně s jedním příkladem, kde je klíčové slovo zvýrazněno. Dále je zde ikona indukující, že uživatel na jakýkoliv řádek může kliknout. Před celým seznamem výsledků je úzký řádek, který ukazuje směr překladu v přehledné grafické formě v podobě dvou vlajek a šipky mezi nimi. Toto je zvláště výhodné, pokud jsou nalezeny výsledky v obou směrech, a tím tento řádek rozděluje výsledky.

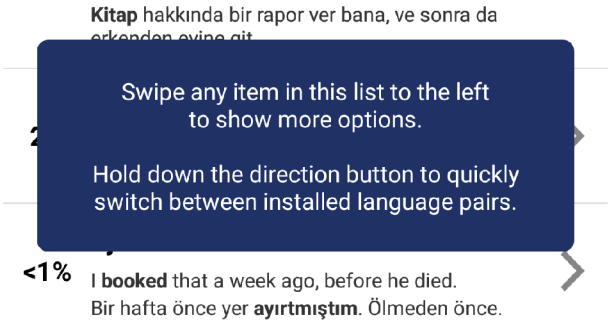
Obrázek 5.13: Výsledek úspěšného překladu

Jak je z výše uvedeného vidět, došlo k mírnému odklonu od návrhu, kdy jsem u každé varianty chtěl zobrazovat tlačítka pro oblíbení položky, přečtení překladu a zkopírování do schránky. Tyto funkce jsem víceméně schoval, jelikož se po implementaci ukázalo, že tlačítka, kterých byl velký počet, znatelně ruší uživatele od pohodlného prohlížení výsledků. Použil jsem *SwipeMenuListView*, jelikož Android ve svém základu nepodporuje odhalení dalších možností pomocí tažení libovolného řádku seznamu, jak to například umí iOS od Applu. Jeden řádek seznamu po odhalení dalších akcí je na obrázku 5.14. Po stisknutí

prvního tlačítka dojde k převedení slova na řeč pomocí třídy *TextToSpeech*, která syntetizuje řeč z textu pro okamžité přehrání. Objekt této třídy je inicializován ihned po startu fragmentu. Druhé tlačítko označí slovo jako oblíbené a uživatel jej může nyní rychle nalézt v Navigation Drawer. Tato funkce pomáhá zlepšit UX, jelikož uživatel nemusí znovu nic vyhledávat a rovnou je mu slovo s příklady zobrazeno. Vybrané slovo je uloženo spolu s jazykem a dalšími parametry do databáze. Posledním „skrytým“ tlačítkem je tlačítko pro zkopírování slova do schránky za účelem použití například v emailu. Toho je dosaženo pomocí rozhraní *ClipboardManager*, díky kterému lze vložit nový objekt *ClipData* do globální schránky.

Obrázek 5.14: Zobrazení dodatečných akcí tažením řádku doleva

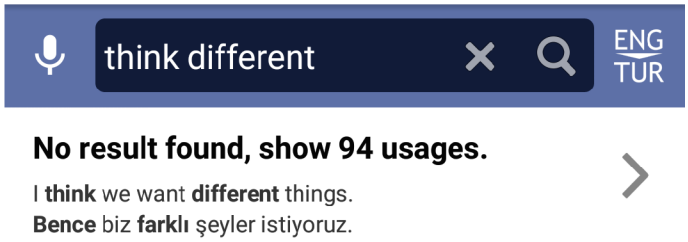
Tento prvek UI ovšem nemusí každý uživatel hned objevit a bylo by nešťastné, kdyby o funkce přišel. Proto jsem implementoval informativní bublinu, která uživatele informuje při úplně prvním vyhledání. Tato bublina je naprogramována pomocí třídy *PopupWindow*, která umí zobrazit libovolný pohled. Toto okno je plovoucí kontejner, který se zobrazí v popředí aktivity. Tímto uživatele informuji, že si může gestem zobrazit více možností a zároveň, že může podržet tlačítko pro přepínání směru, o čemž jsem se zmiňoval výše. Uživatel toto okno zavře jakýmkoliv kliknutím na obrazovku telefonu. Tuto informativní bublinu lze vidět na obrázku 5.15.



Obrázek 5.15: Informativní bublina

Druhý způsob zobrazení výsledků nastane v případě, že hledané slovo či fráze není přímo v databázi, ale pomocí API dotazu dojde k nalezení příkladů použití. V takovém případě je zobrazen uživateli výsledek v podobě jednoho řádku (obrázek 5.16), kde je opět zobrazen jeden příklad použití a také celkový počet příkladů, které si může zobrazit. Zároveň je uživateli ihned zobrazena obrazovka s příklady a to z toho důvodu, že není-li fráze nalezena v databázi, jedná se s největší pravděpodobností o komplikovanější vyhledávání a uživateli jeden příklad zřejmě nebude stačit.

Uživatel může vybrat libovolný řádek seznamu pro zobrazení většího počtu příkladů (pokud existují a je-li online) a bude mu zobrazena nová obrazovka (popsána v kapitole 5.8). Taktéž je důležité z uživatelova pohledu, aby aplikace správně zobrazovala elementy aplikace na základě jeho aktivity. Klikne-li uživatel do textového pole pro zadávání slov, je mu zobrazena historie, když začne psát, tato historie je nahrazena našeptáváním. Když



Obrázek 5.16: Výsledek pouze v podobě příkladů

poté skryje klávesnici, aniž by potvrdil vyhledávání, jsou mu zobrazeny předešlé výsledky vyhledávání. Toto je také implementováno v kombinaci se změnami orientace displeje, což přineslo dodatečné implementační složitosti, ale je to nutné, aby uživatel měl jednotný prožitek napříč aplikací.

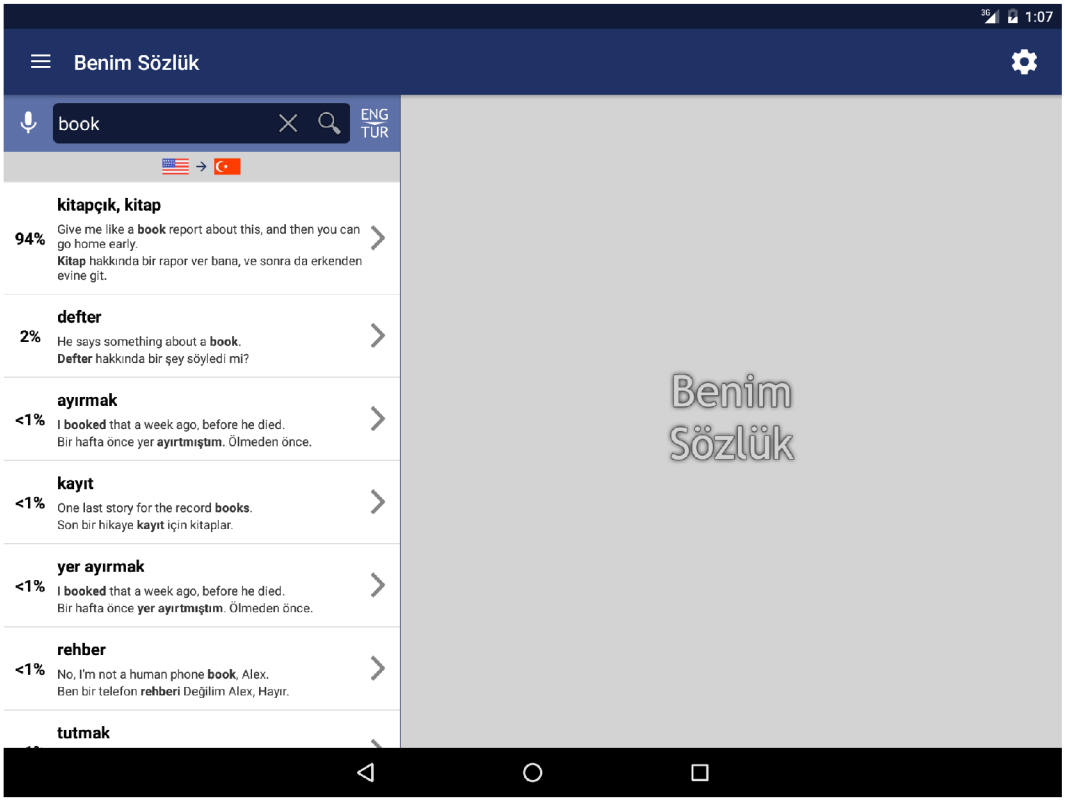
5.7.2 Tablety

Jelikož by byla škoda, aby aplikace nevyužila potenciálu větších displejů, implementoval jsem rozdílný vzhled právě pro tato zařízení. Rozdílná implementace zvedne uživatelskou spokojenost, jelikož nabízí rychlejší a příjemnější zážitek, než kdyby aplikace fungovala stejně jako na telefonech. Hlavní obrazovku tabletové verze aplikace jsem vertikálně rozdělil na dvě části v poměru 35/65. V levé části aplikace je fragment pro zobrazování vyhledávání a výsledků vyhledávání stejně jako to bylo u mobilní verze, ovšem s tím že v pravé části je dále zobrazen fragment pro příklady. S touto změnou bylo také nutné mírně změnit logiku aplikace. Hlavní aktivita se na tabletech také musí starat o zpětnou navigaci mezi fragmenty. Klikne-li uživatel na tlačítko zpět, dojde k odstranění fragmentu příkladů z *back stacku*, což je zásobník spuštěných fragmentů.

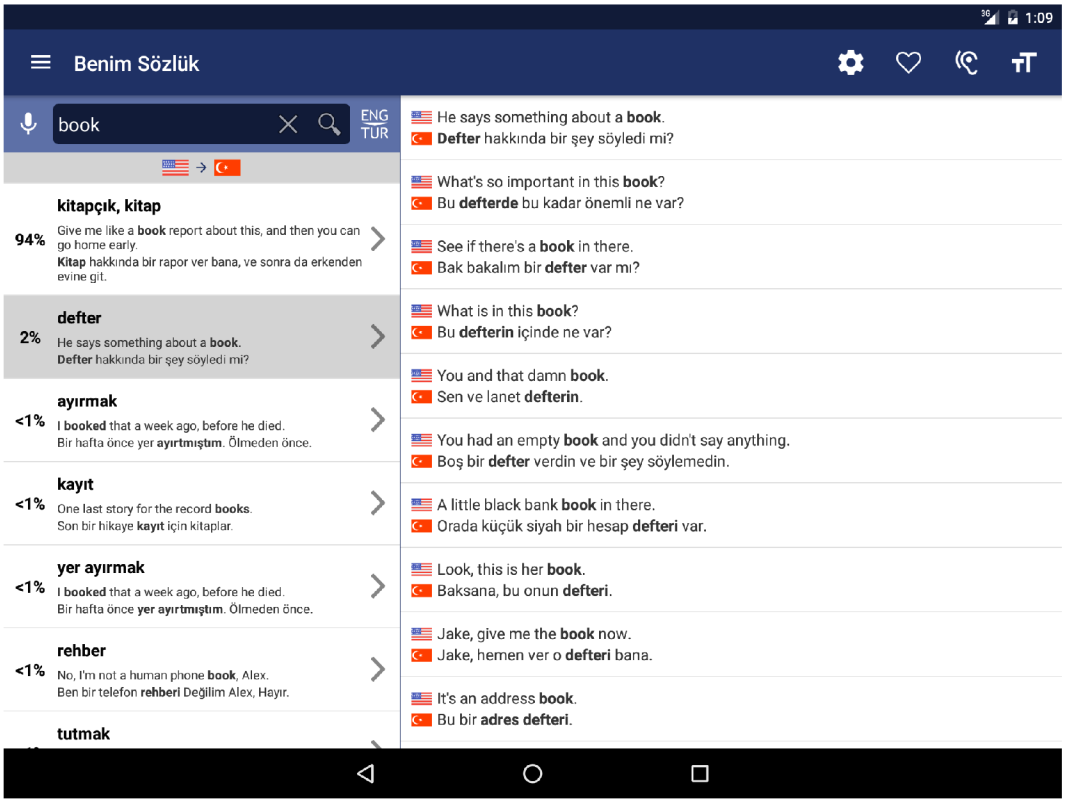
Pravý fragment je graficky zcela stejný jako na telefonech, proto bude popsán podrobněji v kapitole 5.8. Bude zobrazen, když uživatel zvolí překladovou variantu nebo naleznou-li se pouze příklady použití fráze. Protože uživatel stále v levé části vidí seznam překladových variant, jako indikátor zvoleného překladu slouží změna pozadí. Při vybrání libovolné překladové varianty je její pozadí změněno na tmavší šedou barvu, aby uživatel kdykoliv jednoduše poznal, pro jakou variantu má příklady zobrazeny. Pokud zavře fragment s příklady nebo vybere variantu jinou, dojde ke změně indikátoru na jinou resp. žádnou položku. Tento fragment může uživatel, kromě stisknutí tlačítka zpět, zavřít pomocí pohybu dvou prstů směrem k sobě. Z vlastní zkušenosti může být tento způsob zavírání fragmentu pohodlnější než klikání na tlačítko zpět.

Protože je levý fragment stejný jak na telefonech, tak na tabletech, muselo být implementováno rozhodování, zda zobrazit novou aktivitu nebo pouze nový fragment s příklady. Toto jsem učinil tak, že do souboru *dimens.xml* jsem uložil externí zdrojovou boolenovskou proměnnou s názvem *isTablet*, která je na telefonech nastavena na hodnotu *false* a na tabletech naopak na hodnotu *true*. Poté v aplikaci přečtu hodnotu této proměnné, kterou systém Androidu načte z takového souboru, jehož typ odpovídá dané velikosti displeje.

Na obrázku 5.17 je hlavní obrazovka na tabletových zařízeních s uskutečněným vyhledáváním. Dále na obrázku 5.18 je vybrána jedna překladová varianta, pro kterou jsou zobrazeny výsledky (všimněte si také změny tlačítek v Toolbaru, jejichž funkce bude popsána v podkapitole 5.8).



Obrázek 5.17: Hlavní obrazovka na tabletech

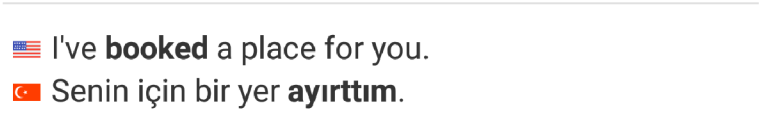


Obrázek 5.18: Hlavní obrazovka na tabletech – vybrána jedna překladová varianta

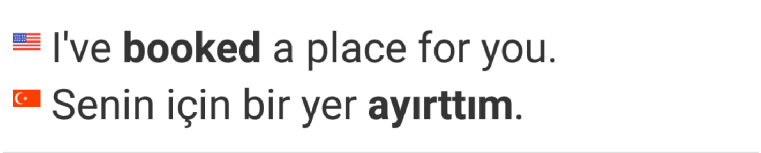
5.8 Obrazovka pro zobrazení příkladů

Na mobilních telefonech je tato obrazovka implementována pomocí aktivity *ActivityWord* a fragmentu *FragmentWord*, kdežto na tabletech je součástí hlavní aktivity jako jediný fragment, který je sice implementován třídou *FragmentWordTablet*, nicméně jeho logika je v podstatě stejná s tím rozdílem, že využívá hlavní aktivitu, jelikož je z ní vytvořen.

Tato obrazovka slouží k zobrazování příkladů použití slov a frází a je zobrazena, když uživatel vybere libovolný řádek v seznamu výsledků, který byl popsán v předchozí podkapitole. V Toolbaru této obrazovky je zobrazeno aktuální slovo, pro které jsou momentálně vypsané příklady. Po levé straně tohoto slova je tlačítko pro návrat na hlavní obrazovku (pouze na mobilních telefonech). Dále má uživatel k dispozici čtyři tlačítka, které každé plní rozdílnou funkci. Prvním tlačítko se symbolem ucha slouží k přečtení aktuálního překladu stejně, jako to uživatel mohl učinit na hlavní obrazovce po odkrytí více funkcí. Další tlačítko se symbolem srdce slouží k oblíbení aktuálního slova. Při kliknutí na toto tlačítko je uživatel informován o úspěšnosti provedení akce pomocí krátké Toast zprávy. Pokud již uživatel má slovo v oblíbených položkách, je ikona tohoto tlačítka změněna na plné srdce. Třetím tlačítkem je tlačítko pro změnu velikosti písma v příkladech, které má ikonu velkého a malého písmene „T“, která představuje právě možnost změnu velikosti. Z hlediska User Experience tato funkce nabídne uživatelům možnost si zvětšit font příkladů, což některým lidem může pomoct, když jim základní menší font nebude vyhovovat. Při výběru jiné velikosti, je tento výběr uložen do *SharedPreferences* pro další použití, aby uživatel nemusel pokaždé měnit velikost písma, protože se dá předpokládat, že poslední použitá velikost mu vyhovovala a hodlá ji použít i u dalších příkladů. Uživatel má na výběr ze tří velikostí. Na obrázku 5.19 je zobrazen jeden příklad použití se základní nejmenší velikostí a naopak na obrázku 5.20 je tentýž příklad s největší velikostí písma. Poslední tlačítko se symbolem lupy má funkci zobrazení stejného vyhledávacího pole, jako je na hlavní obrazovce. Díky tomu může uživatel vyhledávat rovnou z příkladů a nemusí se vracet o obrazovku zpět. Tuto vyhledávací lištu může uživatel kdykoliv opět zavřít tlačítkem se symbolem kříže, které se objeví, když je vyhledávací pole zobrazeno. Celá možnost zobrazit si toto pole je implementována pouze na mobilních telefonech, jelikož na tabletu uvidí uživatel vyhledávací pole stále v levé části obrazovky. Na tabletu tedy toto tlačítko chybí, protože by další zobrazovací pole nedávalo smysl a uživatel by mohl být zmaten.



Obrázek 5.19: Menší velikost písma příkladua



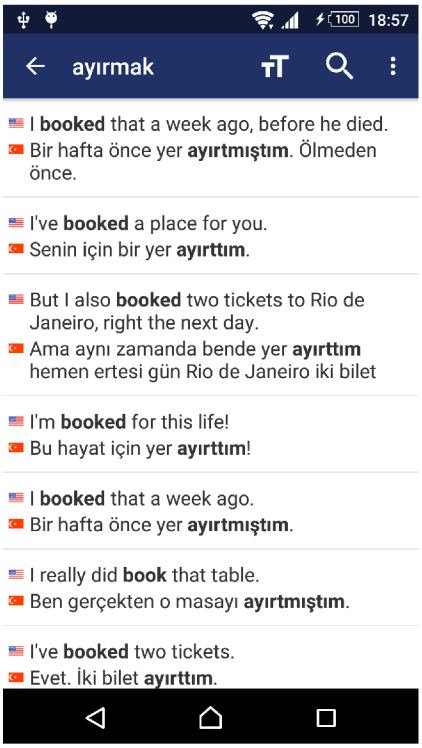
Obrázek 5.20: Největší velikost písma příkladu

Na základě velikosti obrazovky se uživateli nemusí všechna tato tlačítka zobrazit pomocí

ikony. Na menších displejích se čtyři tlačítka vedle sebe bohužel nevejdou, a proto systém Androidu může některé skrýt za tlačítko se symbolem třech vertikálních puntíků. Když uživatel toto tlačítko stiskne, jsou mu další funkce zobrazeny v textové podobě. Celý Toolbar je zobrazen na dalším obrázku, kde se právě dvě tlačítka nevešla na obrazovku, a tak byla skryta pod zmíněné tlačítko „více možností."

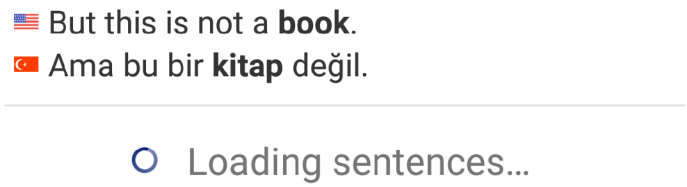
Obrázek 5.21: Toolbar na obrazovce pro zobrazení příkladů

Seznam vět v obou jazycích je implementován pomocí *ListView* a o zobrazování dat se stará adaptér *AdapterSentences*. Samotné stahování příkladů má za úkol implementovaná asynchronní úloha *SearchSentences*. V této úloze dojde k sestavení dotazu pro Lingebra API (viz kapitola 2.2) v podobě URL adresy. Tato adresa je potom odeslána pomocí *HttpURLConnection*. Jako výsledek aplikace obdrží XML soubor, který zpracuji ve třídě *XmlParser*, kde využívám rozhraní *XmlPullParser*, které definuje funkcionalitu parsování XML souborů. Kromě příkladových vět získám z výsledku také celkový počet příkladů, abych věděl, kdy další věty nenačítat. U každé věty je po levé straně zobrazena vlajka země indikující o jaký jazyk se jedná. Bylo-li přeloženo slovo z angličtiny do turečtiny, tak příklady budou zobrazeny pro vybranou tureckou překladovou variantu. První věta bude anglicky a druhá turecky. Hlavní slovo je opět zvýrazněno tučně. Celková podoba této obrazovky je na obrázku 5.22.



Obrázek 5.22: Obrazovka pro zobrazení příkladů

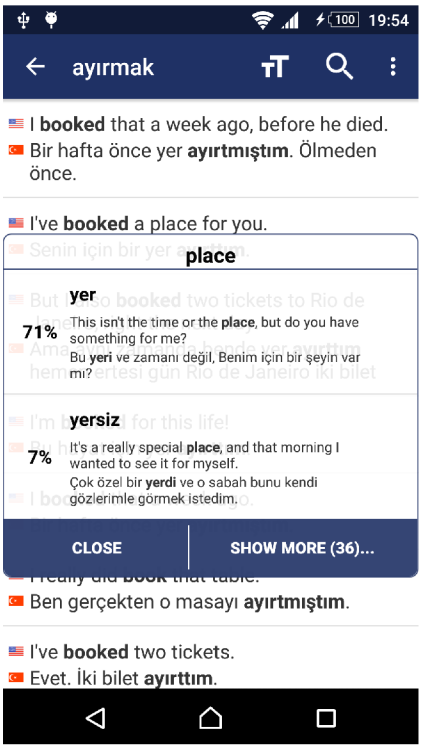
Věty jsou načítány postupně. Nejprve je načteno 20 příkladů a blíží-li se uživatel k poslednímu výsledku, je načteno dalších deset. Toto jsem implementoval pomocí rozhraní `onScrollListener`, které invokuje *callback* metodu, byl-li seznam scrollován. Při prvním načítání a při načítání nových deseti vět je uživateli zobrazen *footer* seznamu, který ho informuje, že jsou nové příklady načítány. Z hlediska UX je důležité, aby věděl, že aplikace pracuje a že zrovna čeká na výsledky. Tato načítací lišta je zobrazena na obrázku 5.23. Původně bylo v návrhu, že v databázi v zařízení budou maximálně tři příklady použití, nicméně společnost ReplayWell si zatím nepřála mít příklady offline. Je-li tedy uživatel bez přístupu na internet nebo internet ztratí v průběhu prohlížení příkladů, zobrazí se mu zpráva informující ho, že musí být online, aby viděl další.



Obrázek 5.23: Načítací footer seznamu s výsledky

V případě, že uživatel uvidí v seznamu příkladů nějaké slovo, které ho dále zajímá, může jej ihned vyhledávat, aniž by ho musel opisovat. To může učinit podržením prstu na slově po dobu jedné sekundy. Pro dosažení této funkčnosti je každé slovo v jednotlivých textových pohledech, které obsahují věty, stisknutelné. Toto se například využívá pro zobrazení odkazů v textu. Jelikož ovšem Android nenabízí vyvolání akce při delším podržení textu, bylo tohoto dosaženo vytvořením abstraktní třídy *LongClickableSpan* dědící ze třídy *ClickableSpan* a následně třídy *CustomLinkMovementMethod* dědící z třídy *LinkMovementMethod*. Původní třídy se využívají k umožnění vyvolání akce po kliknutí na určitý textový řetězec v *TextView*. Moje třídy jsou upraveny, aby podporovaly dlouhý klik. Po provedení takovéto akce, je uživateli zobrazena bublina s překlady stisknutého slova. Toto okno je vytvořeno na stejném principu jako informační bublina, která byla popsána v kapitole 5.7.1. Okno je zobrazeno ihned pod slovem, které uživatel zvolil. Na menších displejích zabírá celou šířku displeje, kdežto na tabletových displejích jen tolik, kolik je nutné. Pozadí okna je mírně průhledné, aby uživatele neomezovalo v prohlížení vět pod ním.

V horní části okna je stisknuté slovo a dále je zde implementován *ListView*, který zobrazuje samotné překlady. Jelikož tato bublina zabírá pouze malou část obrazovky, jsou uživateli zobrazeny maximálně dvě překladové varianty. Opět lze na jakýkoliv výsledek kliknout a uživateli je zobrazena stejná obrazovka jako dosud, ale s příklady pro překladovou variantu, kterou zvolil. Na spodní straně bubliny jsou dvě tlačítka. První je tlačítko „Close,“ které slouží pro zavření oné bubliny. To lze také učinit kliknutím mimo bublinu. Druhé tlačítkem slouží k zobrazení všech překladových variant a nese název „Show more. . . “ Po zvolení tohoto tlačítka je aktuální obrazovka zavřena a proběhne nové vyhledávání se slovíčkem, které bylo v bublině. Jak bublina s výsledky vypadá, je zobrazeno na následujícím obrázku. Jelikož tato funkce, kdy uživatel může podržet prst na jakémkoliv slově, není úplně patrná na první pohled, při prvním zobrazení této obrazovky je uživatel informován o této funkci pomocí stejného okna jako v podkapitole 5.7.1.



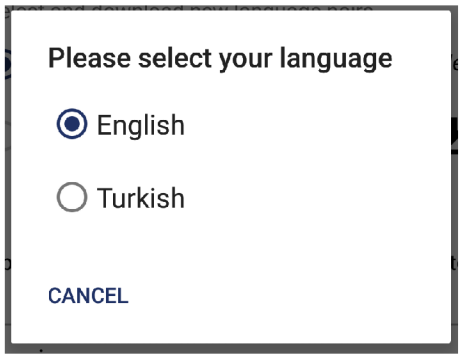
Obrázek 5.24: Překlad slova z příkladových vět

5.9 Obrazovka pro nastavení

Poslední obrazovkou, která byla implementována v mé aplikaci, je obrazovka pro nastavení. Tato obrazovka nebyla původně součástí návrhu, nicméně přidáním podpory pro druhý jazyk rozhraní a kvůli stahování nových jazykových párů, se ukázalo, že je nutné ji implementovat. Do této obrazovky se dostane uživatel po stisknutí tlačítka s ikonou ozubeného kola na hlavní obrazovce. Tato obrazovka je implementována pouze pomocí aktivity v třídě *ActivitySettings*, a tudíž je stejná jak na mobilních telefonech, tak na tabletech. Uživatel zde může změnit jazyk aplikace, stáhnout si nový jazykový pár pro překlady a také si zobrazit novinky nejnovější verze aplikace.

Změna jazyku aplikace je možná kliknutím na tlačítko „Interface“. V zápětí se uživateli zobrazí dialog, ve kterém pomocí dvou přepínacích tlačítek („English“/„Turkish“) lze změnit jazyk rozhraní, jestliže uživatel při prvním startu aplikace zvolil jazyk špatný nebo se prostě rozhodl jazyk změnit. Jakmile uživatel v tomto okně vybere volbu, dojde ke změně *Resources* aplikace a aktivita je restartována již v novém jazyce. V případě, že by změna jazyku nebyla v aplikaci možná, mohl by uživatel aplikaci odinstalovat, než aby provedl reinstalaci. Dialogové okno je zobrazeno na obrázku 5.25.

Stažení nového jazykového páru je jedna z nejdůležitějších funkcí v tomto nastavení (zobrazeno na obrázku 5.26). Aplikace při stažení z Google Play obsahuje jazykový pár turečtina-angličtina, proto tento pár uživatel stahovat nemůže a je u něj pouze poznámka, že je nainstalován. Druhý jazykový pár (turečtina-němčina) uživatel již stahovat může. To učiní kliknutím na tlačítko stahování umístěné vpravo od nápisu „German.“ Po kliknutí na tlačítko a je-li uživatel online, je provedena asynchronní úloha, která stahuje databázi tohoto jazykového páru. Databáze je stažena z internetu pomocí třídy *HttpURLConnection* a ulo-



Obrázek 5.25: Dialogové okno pro změnu jazyku aplikace

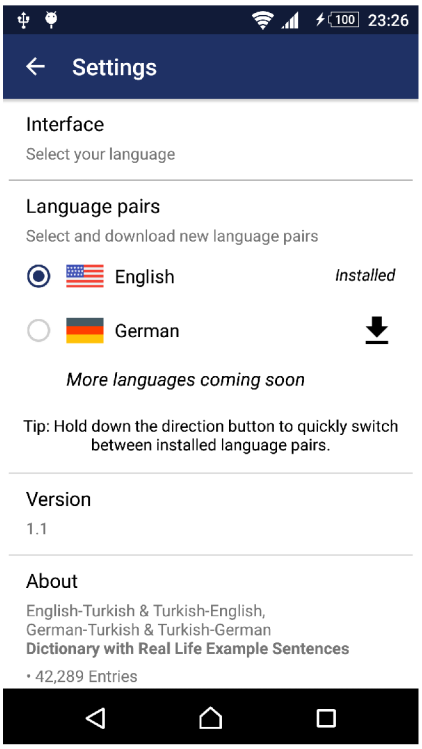
žena do adresáře, který je určen pro moji aplikaci (*data/data/com.replaywell.benimsozluk/databases/*). Během stahování uživatel vidí aktuální stav v podobě počtu procent, který reprezentuje, jaká část souboru byla již stažena. Po úspěšném stažení je do *SharedPreferences* uloženo, že jazyk byl nainstalován a je připraven k použití. Spolu s tím je zobrazeno uživateli, že také turečtina-němčina byla nainstalována. Po levé straně jazykových párů si uživatel může přepnout aktuální jazykový pár stejně, jako kdyby tak učinil na hlavní obrazovce (viz kapitola 5.7.1). Také je zde informativní poznámka, aby uživatel věděl, že lze jazyk přepnout také právě na hlavní obrazovce.

Obrázek 5.26: Stahování nového jazykového páru

Na obrazovce s nastavením dále uživatel vidí aktuální verzi aplikace a po kliknutí na tuto část obrazovky, jsou mu zobrazeny poznámky k vydání aktuální verze v podobě dialogu. Poslední část této obrazovky patří informaci „About," která uživateli zobrazí několik informací o aplikaci, což je v moderních aplikacích dnes standard, že takovýto element je v nich obsažen. Obrazovka pro nastavení je zobrazena na obrázku 5.27.

5.10 Zveřejnění aplikace

Při vývoji této aplikace (v rámci semestrálního projektu) byla vytvořena její první verze. Tato verze byla nahrána na oficiální obchod s aplikacemi Google Play. Aplikace byla nahrána pod účtem společnosti ReplayWell, která plánuje pokračovat v jejím vývoji. V první fázi vývoje uměla vyhledávat a zobrazovat příklady. Taktéž bylo již možné ukládat slovíčka jako oblíbené a zobrazovat historii. Aplikace v tomto stádiu neuměla zatím našeptávat a taktéž nebyl k dispozici druhý jazykový pár (turečtina-němčina). Po dokončení práce na



Obrázek 5.27: Obrazovka pro nastavení

diplomové práci druhá verze nahradila verzi první. Tato verze vypadá a pracuje tak, jak bylo popsáno v celé kapitole 5.

Pro nahrání musí být aplikace podepsána kryptografickým klíčem, jehož ztracení by vyústilo v to, že nepůjde nahrát jakákoliv další verze. Z finální verze by také měli být odebrány všechny ladící výpisy, které programátor použil při vývoji aplikace. Dále by měly být vykonány další úkony jako kontrola, zda aplikace neobsahuje nějaké soubory, které nejsou využity, nebo zda nevyžaduje některé oprávnění, které nejsou nutné pro její chod.

Dodatečné materiály byly vytvořeny pro nahrání aplikace na Google Play. Prvním materiálem jsou snímky obrazovky aplikace, aby potenciální uživatelé viděli, jak aplikace vypadá ještě před stažením. Dále byl vytvořen tzv. feature graphic, což je obrázek, který uživatel uvidí v horní části obrazovky, když si zobrazí detail aplikace na telefonu či tabletu. Posledním materiálem je ukázkové video aplikace, které jednoduchou formou ukazuje, co aplikace umí a jaké jsou její přednosti. Toto video je ke zhlédnutí na přiloženém CD.

Kapitola 6

Testování

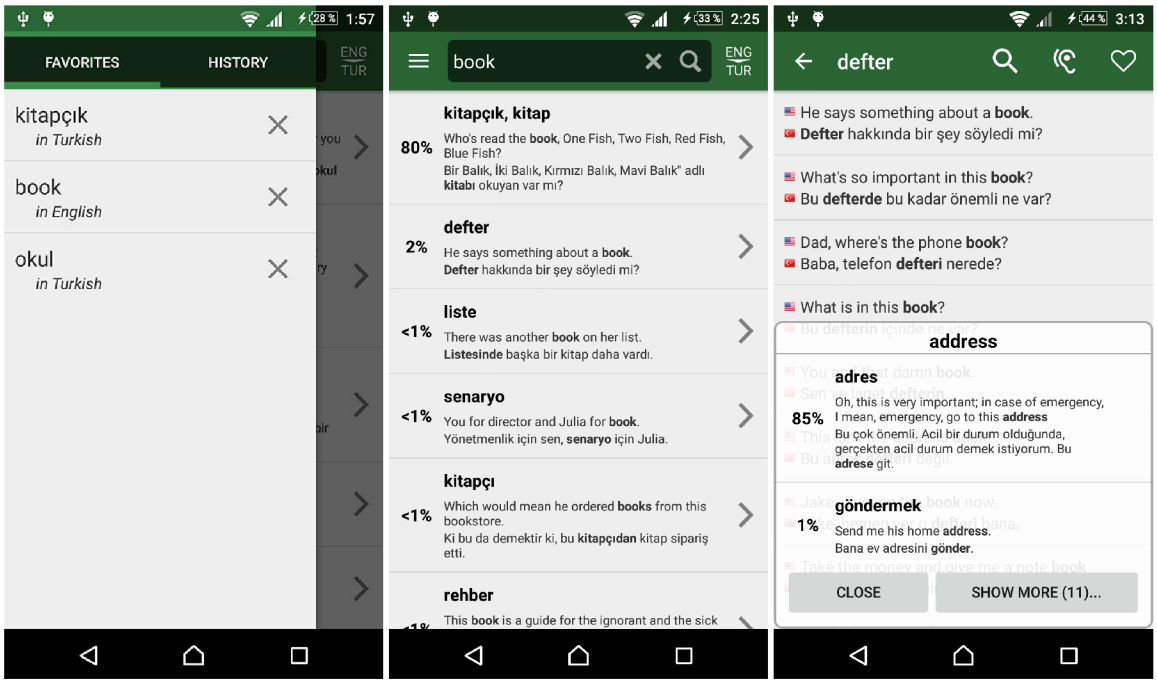
Tato kapitola je věnována testování aplikace, které proběhlo celkem ve dvou fázích. Nejprve bude představeno v 6.1 testování, které bylo uskutečněno během finalizace první verze aplikace před nahráním na Google Play. Druhé testování popsané v kapitole 6.2 bylo provedeno po dokončení druhé, finální verze aplikace.

6.1 Test aplikace pro verzi 1.0

První testování reálnými uživateli proběhlo během dokončování první verze aplikace. V této fázi vývoje aplikace pracovala pouze s jazykovým párem turečtina-angličtina a uměla zobrazovat překlady a příklady pro jednotlivé překladové varianty. Aplikace v době testování neuměla mimo jiné našeptávat a ani neobsahovala obrazovku s nastavením. Taktéž v této verzi nebyla vytvořena podpora pro tablety, která byla implementována až ve druhé verzi. Aplikace podporovala verzi Android 4.2 jako nejnižší verzi, na kterou lze aplikaci nainstalovat. Jak aplikace vypadala v tomto stádiu vývoje lze vidět na obrázku 6.1.

Aplikaci v této fázi testovalo celkem osm uživatelů, kteří ji testovali na chytrých telefonech. Snažil jsem se požádat o testování uživatele s různými verzemi Androidu s co nejpestřejšími velikostmi displeje, abych zjistil, zda se všechny elementy grafického uživatelského prostředí správně zobrazují, zda například nedochází k nechtěnému překrývání elementů nebo závažnějším problémům UI. Povedlo se mi získat uživatele se zařízením s verzí 4.2.2 až po zařízení s nejnovější verzí 6.0. V této fázi vývoje bylo také nezbytné aplikaci otestovat na technické problémy, tedy zda aplikace funguje korektně napříč zařízeními. Testoval jsem, zda aplikace správně pracuje s databází, zda dochází k úspěšnému stahování příkladů z internetu nebo zda nedochází k nechtěným pádům celé aplikace.

Před započetím testování jsem nejdříve požádal testující o vyplnění krátkého dotazníku, který měl za cíl zjistit určité informace ještě dříve, než si aplikaci nainstalují na svá zařízení. První dvě otázky byly na věk a pohlaví uživatelů. Dále měli za úkol vyplnit své zařízení a verzi Androidu, se kterou aplikaci budou testovat. V další otázce jsem se testerů ptal, zda mají nějakou oblíbenou slovníkovou aplikaci, kterou pravidelně používají. Čtyři testeři na tuto otázku odpověděli pozitivně. U těchto lidí jsem zjišťoval, zda jejich slovník nabízí příklady použití ve formě vět, načež všichni odpověděli, že nikoliv. Další otázkou bylo, jakou aplikaci by volili, pokud by si měli zvolit mezi jednou, která příklady nabízí, a druhou, která příklady neobsahuje. Zde testující odpověděli, že by spíše zvolili aplikaci s příklady, ale záleželo by na dalších faktorech jako velikost aplikace nebo schopnost fungování v offline režimu. Na velikost aplikace byla zaměřena následující otázka, kde lidé měli



Obrázek 6.1: Verze aplikace během prvního testování

zvolit, jakou velikost slovníkové aplikace by jim přišla optimální. Na výběr byly možnosti: 1–10MB, 10–20MB, 20–30MB, 30 a více MB. 5 lidí zvolilo první možnost a zbývající zvolili 10-20MB. Následující otázka byla podobná předchozí. Zde měli uživatelé zvolit ze stejných možností, jaká velikost slovníkové aplikace by jim přisla optimální, kdyby aplikace navíc obsahovala maximálně 3 příkladové věty pro každé slovo. Jeden člověk zvolil první možnost 1–10MB, dva lidé zvolili odpověď 10–20MB, tři lidé 20–30MB a poslední dva odpověděli poslední nabízenou možností. Aplikace v době testování obsahovala nejvýše 3 příklady pro každé slovo a její velikost byla 23 MB. Na základě odpovědí se ukázalo, že pět lidí z osmi by byly s velikostí spokojení, u dalších třech by mohl nastat problém. Vzhledem k tomu, že se společnost ReplayWell nakonec rozhodla, že příkladové věty v offline režimu mít nechce (budou se vždy pouze stahovat), byla velikost snížena na 5,7MB (finální aplikace má 7,8MB). Tato velikost by testujícím již bez problémů vyhovovala.

Dále jsem se ptal na určité aspekty mé aplikace, které bylo důležité otestovat dříve, než uživatelé aplikaci viděli. Ptal jsem se uživatele na význam některých ikon aplikace, abych zjistil, zda uživatelé uhodnou její funkci. Ikony a popis testujících je v tabulce 6.1. Z odpovědí mi vyplynulo, že ikony byly zvoleny správně a uživatel při výběru tlačítek s těmito ikonami dokáže bez větších problémů identifikovat jejich funkci, aniž by musel tlačítko zmáčknout a čekat, co se stane, což by bylo nežádané.

Poté byl uživatelům představen skutečný význam ikon, protože u prvních třech ikon mě také zajímalo, která barva by se uživatelům s těmito ikonami asociovala jako první. Toto mě zajímalo z toho důvodu, že barvu pozadí u ikon na obrázku 5.14 lze měnit, což by uživatelům mohlo pomoci v rozlišování mezi tlačítky a zlepšit tak UX. Jak lze vidět v tabulce 6.2 odpovědi jsou u jednotlivých ikon velice rozdílné (kromě ikony pro oblíbení slovíčka), proto jsem se rozhodl pozadí tlačítek nechat světle šedé jako dosud. Rozdílné odpovědi mohli být například způsobeny tím, že si lidé vybavovali použití ikony s rozdílnými barvami nebo by se jim jednoduše barva líbila.

Tabulka 6.1: Ikony a jejich popis od testujících

Ikona	Popis testujících
	Poslouchat; slyšet; nechat si přečíst; naslouchat;
	Nové okno; přesunout; kopírovat; odeslat;
	Oblíbit; uložit; líbí se mi to;
	Zavřít; odejít; vymazat;
	Přejít na další; ukazatel pro zobrazení více možností; další; jít dál; potvrdit; otevřít nabídku;

Tabulka 6.2: Ikony a barva, kterou by testující volili

Ikona	Barva pozadí, kterou si s ní uživatelé asociují
	Modrá (3 odpovědi); Červená (2 odpovědi); Zelená (2 odpovědi); Žlutá (1 odpověď)
	Žlutá (2 odpovědi); Modrá (2 odpovědi); Černá (2 odpovědi); Zelená (1 odpověď); Červená (1 odpověď)
	Červená (6 odpovědí); Růžová (1 odpověď); Žlutá (1 odpověď)

Po tomto dotazníku jsem testující požádal o nainstalování aplikace. Po prvním spuštění aplikace byli dotázání, zda se dialog při prvním spuštění zobrazuje korektně. S uvítacím dialogem neměl nikdo problém. Poté byli uživatelé požádáni, aby se pokusili o vyhledání několika libovolných slov. Uživatelům se překladové varianty pro hledaná slova zobrazovala úspěšně, nebo jim byla ukázána rovnou obrazovka s příklady. U jednoho uživatele byl zaznamenán pád během vyhledávání, který byl ihned opraven, aby k němu již nedocházelo. Během testování se ukázalo, že uživatelé potřebují informaci o tom, že jednotlivé překladové varianty skrývají dodatečné funkce (viz kapitole 5.7.1). Tato nápověda byla zamýšlena a tímto se ověřilo, že je opravdu nutná. Dále si měli uživatelé zobrazovat příklady. Na této obrazovce si jeden tester všiml, že aplikace musí znovu načíst příklady při změně orientace zařízení. Toto bylo vyřešeno na obrazovce s výsledky, ale opomenul jsem řešení implementovat také pro příklady. Tento fakt, byl ovšem po testování napraven. Testování dále ukázalo, že někteří uživatelé mají problém se prstem trefit na slovo, které chtějí dále překládat (na obrazovce s příklady). Toto bylo vyřešeno mírným zvětšením písma a také jsem byl přiveden na myšlenku implementovat funkci, kdy si uživatel může měnit velikost písma sám. Tato funkce byla implementována do druhé verze aplikace. Testující se také vyjádřili, že jim chybí našeptávání slov při psaní. Tato funkce, jak bylo zmíněno, byla ve fázi implementace, proto ještě nebyla ve verzi, která byla součástí testování. Nicméně byla dokončena ještě před nahráním první verze do obchodu.

Toto testování ukázalo, že aplikace neobsahuje závažné technické problémy, spíše několik menších z hlediska jednoduchosti a intuitivnosti ovládání. Ty byly buďto ihned opraveny

nebo jsem na nich pracoval pro druhou verzi.

6.2 Test finální verze 1.1

Druhé testování proběhlo během dokončování práce na aplikaci v rámci této diplomové práce. Cílem bylo opět zjistit závažné nedostatky, které se mohly vyskytnout změnami v implementaci. V této verzi již přibyla obrazovka s nastavením, kde uživatel může stahovat nový jazykový pár (turečtina-němčina), přibylo našeptávání slov a proběhlo několik změn a vylepšení pro co nejlepší User Experience.

Tohoto testu se zúčastnilo 7 z 8 lidí, kteří byli součástí testu prvního, a dále dodatečných pět, kteří aplikaci viděli poprvé. Jelikož v této verzi bylo implementováno vylepšené rozhraní pro tablety, tři lidi testovali aplikaci na tabletech. Test měl opět jednoduchý scénář, kdy se lidé měli s aplikací seznámit (hlavně tedy ti, kteří ji viděli poprvé) a poté provést několik popsaných úkolů, aby určitě viděli vše, co aplikace umí a jak vypadá. Hodnocení aplikace proběhlo ve třech stádiích. První dvě části obsahovaly rozsáhlejší dotazník, kdy byly testeři nejprve zpovídání, co si myslí o uživatelském rozhraní, o intuitivnosti ovládání aplikace a o celkovém pocitu z aplikace. Testující hodnotili otázky na Likertově škále od jedné do pěti (zcela nesouhlasím, nesouhlasím, nevím, souhlasím, zcela souhlasím). Všechny otázky a jejich průměrné hodnocení jsou zaznamenány v tabulce 6.3. Testujících jsem se například ptal, jaký mají celkový dojem z aplikace (otázka 17), zda shledávají navigaci v aplikaci intuitivní (otázka 2) nebo zda aplikace obsahuje dostatečné množství nápověd (otázka 5). Z výsledků, které jsou velice pozitivní, jsem vyvodil závěr, že z hlediska UI a funkcí byla aplikace navržena a implementována správně. Uživatelé například velice kladně hodnotili otázku číslo 15, která se vztahovala k použití elementu Navigation Drawer, z čehož lze usuzovat, že umístění oblíbených položek a historie do tohoto místa byl správný krok.

Další část dotazníku byla určena technickým věcem, jako rychlost zobrazování výsledků, stabilitě aplikace a dalším. Otázky této části jsou v tabulce 6.4. Otázka číslo 3 v této části byla hodnocena relativně negativně, což přikládám tomu, že příklady použití se musí stahovat pro každou překladovou variantu, a což zabere trochu času. Toto by šlo v budoucnu vyřešit přidáním alespoň jednoho příkladu pro každé slovo, znamenající sice větší velikost aplikace, ale výrazně by to pomohlo v rychlosti zobrazování výsledků. Kladně uživatelé hodnotili rychlost zobrazování našeptávání nebo bezproblémovost stahování nového jazykového páru. Uživatelé neutrálně hodnotili otázku 9, která se věnovala čtení překladů, což přikládám tomu, že turečtina, pokud ji uživatelé zatím na zařízení nepoužívali, se musí inicializovat, a tedy její první použití může být pomalejší.

V poslední části dotazníku měli testeři volný prostor k vyjádření, co si o aplikaci myslí. Také zde byl prostor pro cokoliv, co nebylo v dotazníku pokryto a co bych měl vědět. Jelikož byla tato část dotazníku dobrovolná, nezískal jsem mnoho odpovědí, ale například toto vyjádření jednoho testujícího stojí za zmínění: „Aplikace se mi velice líbí je rychlá a s radostí bych jí používal, kdyby byla pro češtinu s angličtinou. Uvítal bych, kdyby mě aplikace nějak uměla vyzkoušet ze slovíček.“

6.3 Výsledky z Google Analytics

V aplikaci bylo implementováno sledování, jak uživatelé používají aplikaci a jaké funkce využívají. Bohužel aplikaci zatím nepoužívá takové množství lidí (cca 50 lidí), ze kterého by šlo dělat nějaké důležité závěry, jako například kterou funkci uživatelé moc nevyužívají

Tabulka 6.3: 1. část dotazníku – Grafické uživatelské prostředí

Otázka	Průměrné hodnocení
1. Tato aplikace se snadno používá.	4,08
2. Navigace v aplikaci je intuitivní a logická.	3,92
3. Aplikaci shledávám moderní a dobře vypadající.	4,25
4. Popsané úkoly jsem dokončil bez problémů.	3,58
5. Aplikace obsahuje dostatek nápověd.	4,58
6. Nápovědy v aplikaci mají zajímavý a decentní vzhled.	3,83
7. Horní lišty v aplikaci mají důležitý obsah.	3,75
8. Vždy vím, co se v aplikaci děje a jak akce dopadly.	4
9. Výsledky vyhledávání jsou přehledně a logicky zobrazené.	4,08
10. Přepínání směru překladu a jazykových párů je intuitivní.	3,5
11. Umístění dodatečných funkcí (po přejetí prstem) je dobře vyřešeno.	3,92
12. Zobrazení příkladů je přehledné.	4,17
13. Bublina pro zobrazení překladů přímo z příkladů je zajímavá a použitelná.	4
14. Stahování nového jazykového páru je jednoduché a intuitivní.	4,75
15. Postranní lišta (Navigation Drawer) aplikace je zajímavě využita.	4,75
16. Správa historie a oblíbených položek je dostatečná.	3,67
17. Baví mě používání této aplikace a aplikaci bych využíval.	4,17

Tabulka 6.4: 2. část dotazníku – Technické aspekty

Otázka	Průměrné hodnocení
1. Aplikace je dostatečně responzivní.	4,5
2. První spuštění aplikace netrvá příliš dlouho.	2,75
3. Vyhledávání překladů je rychlé.	3,25
4. Aplikace správně reaguje na přítomnost internetu nebo naopak na jeho absenci.	3,75
5. Stahování nového jazykového páru je bezproblémové.	4,75
6. Aplikace funguje korektně i při změnách orientace zařízení.	4,92
7. Vyhledávat hlasem je bezproblémové.	3,67
8. Při psaní slov našeptávání funguje dostatečně rychle.	4,92
9. Čtení překladových variant telefonem vždy funguje.	3,33

nebo kde tráví nejvíce času. Tyto statistiky se projeví až časem, jak budou noví uživatelé zkoušet a používat moji aplikaci. Aplikace momentálně zaznamenává navštívené obrazovky, spuštěné funkce, jaký jazyk uživatelé používají a další statistiky.

Kapitola 7

Závěr

Cílem diplomové práce bylo vytvořit slovníkovou aplikaci s příklady na chytrá zařízení s operačním systémem Android. V rámci diplomové práce bylo úkolem nastudovat si informace k slovníkovým aplikacím a provést průzkum již existujících aplikací s tímto zaměřením. Nejen na základě průzkumu a porovnání aplikací byl vytvořen návrh mé slovníkové aplikace. Dále byla tato aplikace implementována se zaměřením na uživatelskou přívětivost. Aplikace v prvních stádiích vývoje fungovala tak, že stahovala celé stránky a ty zpracovávala. Později byla předělána, aby pracovala s API (pro rychlejší komunikaci) a offline databází. Aplikace obsahuje mnoho funkcí, které mají za cíl uživatelům pomoci v jejím používání a nabídnout jim při tom co nejpříjemnější zážitek. Aplikace je plně funkční a ve stavu, v jakém byla popsána v kapitole 5 o implementaci. Uživatel aplikaci nalezne na Google Play[22] a také ji lze nalézt přímo přes web benimsozluk.com[23], kde je vystaven banner na tuto aplikaci.

Práce na této aplikaci mi nabídla příležitost se dále zdokonalovat v programování aplikací na Android. Poznal jsem a naučil se novým principům a technikám, které se změnily od mé bakalářské práce. Aplikace není užitečná pouze pro mě z hlediska jejího programování, ale také pro všechny, kteří ji budou aktivně využívat k překladům a ke zdokonalování svých jazykových znalostí. Věřím, že díky ní již uživatelé vždy budou mít jasno, kterou překladovou variantu použít.

Na závěr si dovolím uvést zhodnocení aplikace od Jana Všianského a Josefa Žižky, jakožto zástupců ze společnosti ReplayWell. „Slovníková aplikace BenimSözlük nevyniká jen po obsahové stránce, kde nabízí k jednotlivým překladům unikátní zobrazení procentuální četnosti užití daného překladu a zejména možnost vidět hledané výrazy a jejich překlady v ukázkových větách tak, jak se v běžném životě skutečně používají; ale i po stránce funkčnosti a designu. Oceňuji zejména celkovou rychlost a přehledné zobrazení včetně citlivého přizpůsobení designu jak pro mobilní zařízení s menší obrazovkou, tak i pro větší tablety. Velmi praktická je dostupnost slovníků a prvních ukázkových vět v offline režimu a možnost downloadu dalších slovníkových dvojic. Celkově vysokou užitnou hodnotu podtrhuje důsledná lokalizace do více jazyků, možnost přehrávání výslovnosti jednotlivých slov a celková stabilita aplikace.“ (Všianský) „Tato aplikace vznikla jako alternativa k webové aplikaci http://www.benimsozluk.com, v mnohých ohledech ji však předčí. Jedná se zejména o povedené a optimalizované uživatelské rozhraní, které se přizpůsobuje velikosti a orientaci displaye. Dále aplikace nabízí možnost uložení hledaných výrazů do oblíbených, pamatuje si jejich historii, umí našeptávat slovníková hesla, zadávat je hlasem či přehrávat výslovnost

[22] https://play.google.com/store/apps/details?id=com.replaywell.benimsozluk&hl=cs
[23] http://benimsozluk.com/

překladových variant. Aplikace nyní nabízí dva slovníky - anglicko-turecký, který je součástí aplikace, a německo-turecký, který si uživatel v případě potřeby může stáhnout do svého zařízení. Aplikace je připravena pro rozšíření o další slovníky, částečně funguje i v off-line režimu a jistě může být mnoha uživatelům dobrým pomocníkem.“ (Žižka)

Další vývoj aplikace má velký potenciál. Aplikace by šlo dále zdokonalovat jak z hlediska funkčnosti, tak také z hlediska User Experience. Jak jeden testující zmínil, aplikace by také mohla nabídnout nějaký způsob učení či procvičování slovíček. Nicméně největší potenciál by byl v rozšiřování nabídky jazykových párů a taktéž lze na základě mé aplikace vytvářet aplikace další, které by pracovaly s jiným hlavním jazykem nežli turečtinou, která je klíčovým jazykem v této aplikaci, jež je výsledkem mojí práce. Další důležité poznatky by jistě přineslo sledování uživatelů, které v aplikaci bylo implementováno, ale které prozatím nepřineslo dostatečné množství dat pro učinění jakýchkoli závěrů. Posledním zdrojem nápadů a oprav by mohli být recenze na Google Play.

Literatura

[1] IDC. *Smartphone OS Market Share, 2015 Q2* [online]. 2015 [cit. 2016-01-02]. Dostupné na: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.

[2] Android Developers. *Introduction to Android* [online]. 2016 [cit. 2016-01-07]. Dostupné na: <http://developer.android.com/guide/index.html>.

[3] HARDY, Brian a Bill PHILLIPS. *Android Programming: The Big Nerd Ranch Guide*. Indianapolis: Pearson Technology Group, 2013. ISBN 978-0321804334.

[4] LEHTIMAKI, Juhani. *Smashing Android UI: Responsive User Interfaces and Design Patterns for Android Phones and Tablets*. Chichester: John Wiley & Sons, Inc., 2013. ISBN 978-1-118-38728-3.

[5] MAŇÁK, Libor. *Mobilní ovládač PC (Mobil jako vzdálené ovládání)*. Brno: FIT VUT v Brně, 2014. Bakalářská práce. Dostupné na: <http://www.fit.vutbr.cz/study/DP/BP.php?id=16627&file=t>.

[6] Android Developers. *Supporting Different Screens* [online]. 2016 [cit. 2016-01-07]. Dostupné na: <http://developer.android.com/training/basics/supporting-devices/screens.html>.

[7] BALPINAR, Zülal. *Turkish Phonology, Morphology and Syntax*. Eskişehir: Anadolu Universitesi, 2006. ISBN 975-06-0261-7.

Přílohy

Seznam příloh

Příloha A

Obsah CD

- /zprava/ - tato technická zpráva ve formátu PDF
  - /latex/ - zdrojové soubory pro LaTeX
- /kod/ - zdrojový kód (kompletní projekt možný importovat do Android Studia)
- /aplikace/ - instalační soubor aplikace ve formátu apk
- /dok/ - instalační návod a uživatelský manuál
- /media/ - propagační multimediální soubory
  - /video/ - propagační video
  - /plakat/ - plakát
  - /market/ - snímky aplikace umístěné na Google Play