



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Aplikace pro podporu výuky angličtiny
Student:	Michal Píbyl
Vedoucí:	Ing. Miroslav Balík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Vytvořte Android aplikaci pro podporu výuky angličtiny. Tato aplikace bude sloužit uživatelům, kteří chtějí zlepšit svoji znalost anglického jazyka. Aplikace bude obsahovat základní tematicky rozdělenou sadu slovníků. Uživatelé si do aplikace budou moci vkládat skupiny slov, které v daném jazyce neznají. Dále pak aplikace nabídne možnost procvičit znalost dané skupiny vložených slov. Uživatelé si budou moci procvičit výslovnost a překlady slov. U procvičování výslovnosti by měla aplikace oznámit uživateli, jestli je výslovnost správná nebo nesprávná, případně by měla oznámit, kde je ve výslovnosti chyba. Použijte vhodné API pro on-line rozpoznávání hlasu (voice recognition).

Práce bude obsahovat:

- analýzu konkurenčních aplikací dostupných pro OS Android,
- návrh aplikace,
- implementaci aplikace podle návrhu,
- testování aplikace včetně testování uživatelského rozhraní,
- seznam možných vylepšení.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 29. listopadu 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Aplikace pro podporu výuky angličtiny

Michal Příbyl

Vedoucí práce: Ing. Miroslav Balík, Ph.D.

8. ledna 2018

Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu Ing. Miroslav Balík, Ph.D. za mnoho cenných rad při návrhu a realizaci práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 8. ledna 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Michal Příbyl. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Příbyl, Michal. *Aplikace pro podporu výuky angličtiny*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Výsledkem této práce bude aplikace, která podpoří výuku anglického jazyka. Aplikace zjednoduší učení se cizího jazyka tím, že se uživatelé budou moci učit jeho různé oblasti, jako je slovní zásoba, výslovnost a poslech na jednom místě.

V práci je nejprve popsána analýza, kde se analyzovaly konkurenční aplikace. Poté se z cílů a analýzy konkurenčních aplikací vytvořily funkční a nefunkční požadavky. Další část práce je složena z návrhu, kde jsou popsány hlavní komponenty aplikace, její architektura a uživatelské rozhraní. Dále pak jsou v návrhu popsány případy užití, které jsou vytvořeny z funkčních požadavků. V dalších částech je pak popsána implementace a testování aplikace.

Klíčová slova voice recognition, android, výuka jazyka, mobilní aplikace

Abstract

The end result of this work will be an application, which supports learning in the English language. The application will make the learning of foreign language easier by allowing users to learn different areas of the language, like vocabulary, pronunciation and listening in one place.

This work first describes analysis, where competitors are analysed. Functional and Non-functional requirements were created from this analysis. Next part of application consists of design, where main components, architecture and user interface are described. The design also describes use cases, which are created from functional requirements. In remaining parts, implementation and testing are described.

Keywords voice recognition, android, language learning, mobile application

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Konkurenční aplikace	5
2.2 Analýza požadavků	9
2.3 Analýza zdrojů dat	11
2.4 Analýza technologií	12
3 Návrh	15
3.1 Kvíz	15
3.2 Případy užití	16
3.3 Doménový model	18
3.4 Databázový model	19
3.5 Agregace dat	20
3.6 Architektura	21
3.7 Uživatelské rozhraní	24
3.8 Technologie pro zjednodušení vývoje	26
4 Implementace	29
4.1 Implementace prototypů	29
4.2 Implementace aplikace	31
5 Testování	33
5.1 Testování funkčních požadavků	33
5.2 Testování uživatelského rozhraní	34
6 Možné vylepšení aplikace	35

Závěr	37
Literatura	39
A Seznam použitých zkratk	41
B Náhledy obrazovek z výsledné aplikace	43
C Instalační manuál	47
D Obsah příloženého CD	49

Seznam obrázků

3.1	doménový model	19
3.2	databázový model	20
3.3	reprezentace Clean Architecture	21
3.4	návrh architektury	22
3.5	obrazovky úkolů	25
B.1	obrazovky pro vybrání typu procvičování spolu s hlavní obrazovkou	43
B.2	obrazovky úkolů	44
B.3	obrazovky přechodů mezi úkoly	45
B.4	obrazovky statistik a přehled odehraného kvízu	45
B.5	obrazovky pro editaci a tvorbu skupin	46

Seznam tabulek

2.1	shrnutí charakteristik Elsa Speak: Reduce your accent	6
2.2	shrnutí charakteristik Speak English Pronunciation	6
2.3	shrnutí charakteristik Learn English Vocabulary	7
2.4	shrnutí charakteristik English Fun Easy Learn	7
2.5	shrnutí charakteristik English vocabulary : flashcard	8
2.6	shrnutí charakteristik Andy - English Speaking Bot	8
2.7	shrnutí charakteristik English Grammar and Vocabulary	9
4.1	Výsledky testu pro Google Voice Recognition	30
4.2	Výsledky testu pro PocketSphinx	30

Úvod

Pro úspěšné naučení se nějakého cizího jazyka je nutné být s ním co nejvíce v kontaktu - tj. trénovat slovní zásobu, znát výslovnost daných slov atd.

Existuje mnoho aplikací zaměřujících se na výuku anglického jazyka. Většina z nich se však zaměřuje pouze na jednu oblast a uživatel pak musí používat více aplikací.

Motivací pro moji práci je zjednodušit lidem učení se anglického jazyka, a to tím, že se mohou učit jeho různé oblasti, jako je slovní zásoba, výslovnost a poslech na jednom místě. V mojí aplikaci si budou moci uživatelé zlepšovat svoji znalost slov zábavnou formou pomocí kvízů, které je vyzkouší z překladu i správné výslovnosti.

V první části této práce se budu věnovat analýze, kde nejprve detailněji prozkoumám konkurenční aplikace. Poté určím funkční a nefunkční požadavky a technologie nebo data, které budou potřeba pro jejich splnění.

V dalších částech se budu zabívat návrhem, implementací a testováním aplikace.

Cíl práce

Jednotlivé cíle jsou popsány níže:

- aplikace bude obsahovat základní tematicky rozdělené skupiny slov
- v aplikaci bude možné vytvářet vlastní skupiny slov, které pak budou vloženy mezi ty základní
- v aplikaci bude možné procvičovat skupiny slov pomocí kvízů
- aplikace bude obsahovat více druhů kvízů
- v aplikaci bude možné procvičovat překlad nebo výslovnost skupiny slov
- aplikace bude informovat o pokroku učení v každé skupině slov

Analýza

2.1 Konkurenční aplikace

V současné době existují spousty výukových aplikací cizích jazyků pro Android, většina z nich se ale zaměřuje pouze na jeden aspekt daného jazyka, např. na učení se nových slov, ale pak opomíjí zlepšování výslovností, gramatiky a jiné. Konkurenční aplikace jsem tedy rozdělil do tří kategorií: aplikace zabývající se výslovností, učením se nových slov a aplikace zabývající se jinými oblastmi jazyka. U každé kategorie jsem vybral jednoho nebo více zástupců, které jsem popsal detailněji.

U konkurenčních aplikací jsem se zaměřoval na jejich 3 charakteristiky: funkcionalitu, vzhled a intuitivnost. Tyto charakteristiky jsem pak ohodnotil. Níže se budu zabývat pouze aplikacemi, které lze stáhnout z Google Play. Aplikace z jiných zdrojů nebyly vyhledávány.

2.1.1 Aplikace zabývající se výslovností

Problémem u aplikací které se zabývají převážně výslovností daných slov je nemožnost používat je bez připojení k internetu.

Zástupci:

- Speak English Pronunciation[1]
- Elsa Speak: Reduce your accent[2]

První zástupce je aplikace Speak English Pronunciation. Uživatelé si v ní mohou trénovat výslovnost souhlásek nebo samotných slov. Kontrolu výslovnosti však musí uživatel provést sám tím, že ji nahraje a poté si nechá přehrát svoji a správnou výslovnost. Další věc která jde procvičovat v této aplikaci je znalost fonetického přepisu anglických slov.

2. ANALÝZA

Funkcionalita	<ul style="list-style-type: none">• nahrát výslovnost• přehrát výslovnost
Klady	<ul style="list-style-type: none">• lze ji používat bez internetového připojení• je zdarma
Zápory	<ul style="list-style-type: none">• je jen částečně v češtině• neintuitivnost UI

Tabulka 2.1: shrnutí charakteristik Elsa Speak: Reduce your accent

Dalším zástupcem je Elsa Speak: Reduce your accent. V této aplikaci mají tematicky rozdělená slova, fráze nebo věty do různých kategorií jako je např. cestování, jídlo, zdraví, atd. U procvičování vět nebo frází uživatelům oznámí i v jakých slovech udělali chybu. Aplikace však kontroluje většinou pouze jednu část slova, místo celé věty.

Funkcionalita	<ul style="list-style-type: none">• nahrát výslovnost• přehrát výslovnost• kontrola výslovnosti
Klady	<ul style="list-style-type: none">• nabízí trial verzi se základními lekcemi k procvičení• zobrazuje přehledné statistiky z procvičených lekcí
Zápory	<ul style="list-style-type: none">• není v češtině• za každou další lekci si uživatel musí zaplatit

Tabulka 2.2: shrnutí charakteristik Speak English Pronunciation

Aplikace zabývající se učením se nových slov

Mnoho aplikací zabývajících se učením se nových slov, jako jsou Babbel nebo Memrise nejsou dostupné v češtině. Nabízí jen možnost učit se ze známých jazyků (němčina, španělština, ...) do angličtiny, ne však z anglického jazyka do anglického. Pro české uživatele, kteří jsou již pokročilí v anglickém jazyce a chtějí si rozšířit svoji slovní zásobu, pak tyto aplikace použít také nemohou.

Zástupci:

- English Fun Easy Learn[3]
- Learn English Vocabulary[4]
- English vocabulary : flashcard[5]

Mezi aplikace které mohou použít čeští uživatelé jsem vybral tři zástupce. První z nich je English Fun Easy Learn, ve které si uživatelé mohou procvičit svoji slovní zásobu pomocí různých kvízů. Aplikace nabízí spousty kategorií

rozdělených podle tématu (cestování, zvířata, sport, ...) k procvičování, ve kterých je obsaženo 6000 slov v bezplatné verzi. Chybí mi v ní však kategorie, které jsou cílené na různě pokročilé uživatele nebo možnost tvorby vlastních kategorií.

Funkcionalita	<ul style="list-style-type: none"> • procvičování překladu
Klady	<ul style="list-style-type: none"> • 7 různých kvízů • vzhled • obrázky ke každému slovu • 6000 slov zdarma k procvičení • je v češtině
Zápory	<ul style="list-style-type: none"> • chybí kategorie rozdělené podle úrovně uživatele • nemožnost si vytvářet vlastní skupiny slov

Tabulka 2.3: shrnutí charakteristik Learn English Vocabulary

Jako druhého zástupce jsem vybral Learn English Vocabulary, který si od English Fun Easy Learn liší hlavně tím, že nabízí kategorie slov rozdělené nejen podle tématu, ale i podle úrovně uživatele. Dalším rozdílem je možnost vytvářet vlastní kategorie. Vytváření vlastní kategorie tu však není řešeno nejlépe, protože do kategorie lze přidat pouze 2 slova najednou, takže to zabere příliš mnoho času.

Funkcionalita	<ul style="list-style-type: none"> • procvičování překladu • tvorba vlastních skupin slov
Klady	<ul style="list-style-type: none"> • 4 různé kvízy • možnost si vytvářet vlastní skupiny slov • 10000 slov zdarma k procvičení • je v češtině
Zápory	<ul style="list-style-type: none"> • mnoho obrázků nesedí ke slově, kterým jsou přiřazena • není úspůsobena na malé displeje

Tabulka 2.4: shrnutí charakteristik English Fun Easy Learn

Poslední zástupce této kategorie je English vocabulary : flashcard. Tato aplikace je velice podobná English Fun Easy Learn, její cílovou skupinou jsou však děti nebo úplní začátečníci. Jejich kategorie slov jsou rozdělené dle tématu. Ke každému slovu v kategorii mají obrázek, avšak často velmi matoucí.

2. ANALÝZA

Funkcionalita	<ul style="list-style-type: none">• procvičování překladu
Klady	<ul style="list-style-type: none">• 6 různých kvízů• je v češtině
Zápory	<ul style="list-style-type: none">• mnoho obrázků nesedí ke slovům, kterým jsou přiřazena• nemožnost si vytvářet vlastní skupiny slov

Tabulka 2.5: shrnutí charakteristik English vocabulary : flashcard

2.1.2 Aplikace zabývající se jinými oblastmi jazyka

V této kategorii jsem shromáždil aplikace zaměřující se především na jiné oblasti jazyka kategorie zmíněné výše nebo kombinují více oblastí dohromady.

Zástupci:

- Andy - English Speaking Bot[6]
- English Grammar and Vocabulary[7]

Jako prvního zástupce jsem vybral aplikaci Andy - English Speaking Bot. Liší se velmi mnoho od ostatních již zmíněných aplikací, protože pro procvičování nepoužívá kvízy, hry ani pasivně text, ale používá bota který vám dává různé úkoly nebo s vámi komunikuje o různých věcech. Uživatelé pak mají pocit jako kdyby si povídali s roditelým mluvčím.

Funkcionalita	<ul style="list-style-type: none">• chatování s botem
Klady	<ul style="list-style-type: none">• velmi zábavný přístup učení se jazyka
Zápory	<ul style="list-style-type: none">• není v češtině• lekce v procvičování gramatiky si musí uživatel dokupovat

Tabulka 2.6: shrnutí charakteristik Andy - English Speaking Bot

Dalším zástupcem je English Grammar and Vocabulary, která se zabývá, jak již napovídá její název především gramatikou a vytvořením si vlastního seznamu slov, které v angličtině neznáte. Pro procvičování gramatiky aplikace obsahuje mnoho testů, které pokrývají problematické oblasti gramatiky. Slovní zásoba se však v této aplikaci procvičit nedá. Je tu tak nevyužitý potenciál vlastního seznamu slov, který si můžete tvořit v průběhu procvičování gramatiky, aplikace totiž dovoluje jednoduché přidání slov do seznamu tím, že po označení jakékoliv slova se automaticky uloží.

Funkcionalita	<ul style="list-style-type: none"> • získání překladu po kliknutí na slovo v aplikaci • přehrát výslovnost
Klady	<ul style="list-style-type: none"> • je v češtině
Zápory	<ul style="list-style-type: none"> • uživatel si může vytvářet svoji skupinu slov, kterou pak ale nemůže nijak procvičovat

Tabulka 2.7: shrnutí charakteristik English Grammar and Vocabulary

2.2 Analýza požadavků

Funkční požadavky vychází z cílů popsaných výše. Jedním z cílů této aplikace je možnost procvičit si 2 oblasti jazyka: překlad a výslovnost. Některé funkční požadavky tedy vychází z analyzovaných funkcionalit konkurenčních aplikací, které se zaměřují pouze na jednu oblast.

2.2.1 Funkční požadavky

- F1 - Vytvořit kvíz
- F2 - Procvičit skupinu slov
- F3 - Zobrazit pokrok učení
- F4 - Zvolit druh procvičování
- F5 - Zobrazit skupiny slov
- F6 - Vytvořit skupinu slov
- F7 - Editovat skupinu slov
- F8 - Zobrazit skupinu slov

2.2.1.1 F1 - Vytvořit kvíz

Aplikace bude umět vytvářet kvízy, které se budou skládat z úkolů. Za tyto úkoly pak uživatelé získají skóre. Tvorba nových kvízů bude zohledňovat uživatelské dosavadní skóre.

2.2.1.2 F2 - Procvičit skupinu slov

Uživatelé si budou moci procvičovat skupinu slov čtyřmi způsoby. Prvním z nich je spustit kvíz, který uživatele procvičí z překladu. Druhý způsob je spustit kvíz, který je zaměřený na zlepšení uživatelské výslovnosti. Třetí z nich kombinuje dva předchozí, tj. spustit kvíz, který je zaměřený na překlad i výslovnost. Posledním způsobem je zobrazit si všechny slova skupiny, kde bude mít uživatel zobrazené anglické slovo spolu s jeho všemi překlady.

2.2.1.3 F3 - Zobrazit pokrok učení

Uživatelé si budou moci v aplikaci zjistit svůj pokrok v učení. Aplikace zobrazí celkovou statistiku znázorňující pokrok v jednotlivých oblastech jazyka (výslovnost, slovní zásoba). Uživatelé si budou moci zobrazit pokrok v jednotlivých skupinách slov.

2.2.1.4 F4 - Zvolit druh procvičování

Aplikace nabídne uživateli navolit si pomocí dvou atributů druh procvičování. Prvním z nich je oblast jazyka, kde bude mít uživatel na výběr mezi procvičováním překladu, výslovnosti nebo obojího. Druhým atributem je kritérium podle kterého se určí z jakých slov se vytvoří kvíz. Uživatel si zvolí, jestli se kvíz vytvoří celý náhodně nebo jestli má zohlednit uživatelův současný pokrok ve zvolené oblasti respektive oblastech jazyka ve skupině.

2.2.1.5 F5 - Zobrazit skupinu slov

V aplikaci si budou moci uživatelé zobrazit seznam všech slov obsažený ve skupině.

2.2.1.6 F6 - Vytvořit skupinu slov

Aplikace bude umožňovat tvorbu vlastní skupiny slov, která bude tvořena pouze anglickými slovy. Do skupin se budou moci vkládat pouze anglická slova, která jsou uložena v databázi. Minimální počet slov ve skupině bude 6. Název skupiny musí být unikátní. Slova v ní se budou moci opakovat ve více skupinách.

2.2.1.7 F7 - Editovat skupiny slov

Uživatelé si budou moci mazat skupiny slov, které jsou předvytvořené nebo i vlastní. Po smazání skupiny se automaticky smažou i všechny její statistiky.

V aplikaci bude možné upravovat skupinu slov. Uživatelé budou moci upravovat skupinu slov tím, že do ní přidají nová slova nebo z ní naopak slova odeberou.

2.2.1.8 F8 - Zobrazit skupiny slov

Uživatelé si budou moci zobrazit seznam všech skupin slov. Seznam bude obsahovat předvytvořené i vlastní skupiny.

2.2.2 Nefunkční požadavky

- N1 - Funkčnost aplikace bez internetového připojení
- N2 - Funkčnost aplikace na platformě Android

- N3 - Rychlý přístup ke všem částem systému

2.2.2.1 N1 - Funkčnost aplikace bez internetového připojení

Aplikace bude bez připojení k internetu splňovat všechny funkční požadavky.

2.2.2.2 N2 - Funkčnost aplikace na platformě Android

Aplikace půjde nainstalovat a použít na Android zařízeních, které mají verzi 4.4 a vyšší.

2.2.2.3 N3 - Rychlý přístup ke všem částem systému

Uživatelé budou mít rychlý přístup ke všem částem systému, tj. UI bude intuitivní a přehledné.

2.3 Analýza zdrojů dat

Aby se naplnil nefunkční požadavek N3, tak je potřeba získat všechna potřebná data, jako jsou anglická slova, jejich české překlady, výslovnosti, popřípadě další informace a to bez použití vzdálených API, která budou vložena do aplikace. Seznamů slov obsahující anglická slova existuje velké množství. Problémem však bylo nalézt větší množství překladů nebo výslovností k daným anglickým slovům.

Dalšími získanými daty byly seznamy anglických slov rozdělených podle téma, popřípadě podle obtížnosti(A1, A2, BNC, ...).

2.3.1 Překlady

Cílem bylo nalézt dostatečně rozsáhlý, volně dostupný slovník, obsahující alespoň 10000 slov. Slovník takové velikosti jsem našel pouze jeden[8], který však vysoce přesahoval můj požadavek na množství překladů, protože obsahoval přes 80000 anglických slov, ke kterým bylo i více překladů. V tomto slovníku jsou u některých slov obsaženy slovní druhy a různé další informace.

2.3.2 Výslovnosti

U vyhledávání volně dostupných databází s anglickými slovy a jejich výslovnostmi jsem měl stejný požadavek jako u slovníků výše - tj. aby databáze obsahovala aspoň 10000 výslovností. Vybral jsem databázi od CMUDict[9], která obsahuje více než 170000 výslovností ve tvaru Arpabet.

2.3.3 Skupiny slov

Musel jsem vyhledat databáze obsahující anglická slova rozdělená do různých skupin a to buď podle téma (cestování, zvířata, ...) nebo podle použitelnosti/obtížnosti (CEFR, BNC). Jako zástupce skupiny slov rozdělené podle použitelnosti slov jsem našel BNC, který je volně dostupný na webové stránce Wiktionary. Tato stránka se ukázala jako bohatý zdroj dat, protože se zde nachází různé slovníky ve kterých jsou obsažené překlady a výslovnosti. Na získání těchto dat by se však muselo použít nástrojů, které tyto data získají přímo z webových stránek nebo jejich archivů.

2.3.4 Obrázky

Jako další zdroj dat se hledala databáze obsahující vhodný počet slov (aspoň 1000), která by obsahovala anglická slova a k nim přiřazený vystihující obrázek. Ve stránce Wiktionary, která je zmíněná výše je obsažena databáze s obrázky, obsahuje však pouze desítky obrázků. Databázi obsahující více než 1000 slov se mi nalézt nepodařilo. Další možností jak získat obrázky je pomocí vzdáleného API, které by je získávalo dynamicky za běhu aplikace nebo získat obrázky pouze ke sloům obsažených ve skupinách, které neobsahují mnoho slov.

2.4 Analýza technologií

Níže uvádím výčet technologií, které budou použity pro tvorbu aplikace. Tyto technologie budou použity pro splnění funkčních požadavků.

2.4.1 Technologie pro splnění funkčních/nefunkčních požadavků

1. rozpoznávání hlasu
2. převod textu do mluvené řeči
3. zobrazení grafu

2.4.1.1 Rozpoznávání hlasu

Důležitou součástí aplikace je Voice Recognition, který bude vhodně rozpoznávat uživatelův hlas v angličtině a bude umět pracovat i bez internetového připojení.

Mezi známé technologie na Voice Recognition, které dokáží pracovat i bez internetového připojení patří Pocketsphinx od CMUSphinx a Google Voice Recognition od Googlu.

Pocketsphinx funguje bez internetového připojení díky vlastnímu slovníku s výslovností a akustickým modelem, který byl se svojí malou velikostí vhodný pro mobilní zařízení.

Google Voice Recognition má stejné vlastnosti jako Pocketsphinx pouze tehdy, pokud má uživatel určitou verzi Androidu a má stážený anglický jazykový balíček. Mezi jeho hlavní nevýhodu však patří nízká konfigurovatelnost.

Bude potřeba zjistit kvalitu a přesnost obou výše zmíněných technologií. Jejich kvalita se ověří pomocí dvou vzorových aplikací[10][11], které nabízí Google i CMUSphinx na svých webových stránkách.

2.4.1.2 Převod textu do mluvené řeči

Tato technologie bude použita pro přehrání správné výslovnosti uživateli. Při výběru vhodného frameworku pro převod textu na mluvenou řeč jsem se rozhodl použít Google Text-To-Speech, jelikož už je v Androidu obsažen od jeho ranných verzí, tj. splňuje nefunkční požadavek N2.

2.4.1.3 Zobrazení grafů

Při hledání vhodného technologie pro zobrazování grafů v Androidu jsem narazil na spousty frameworků které by se dali použít. Většina z nich však měla nepřehlednou dokumentace a nebo ji neměla vůbec.

Nakonec jsem zvolil MPAndroidChart, který je mezi open-source frameworky nejpoužívanější a má i přehlednou dokumentace, kde jsou i vzorové příklady použití. Dokáže zobrazit základní typy grafů jako jsou koláčové, spojnicové a další. Zobrazené grafy se mohou přibližovat/oddalovat a dají se do nich přidat různé animace.

Návrh

3.1 Kvíz

Hlavní komponentou této aplikace budou kvízy, které se budou skládat ze čtyř typů úkolů. Kvíz pak bude uživateli dávat úkoly k následnému vyřešení, tj. bude rozhodovat jaký úkol je pro uživatele v danou chvíli nejvhodnější. Pro dokončení kvízu je potřeba vyřešit všechny úkoly. Mezi jednotlivými úkoly aplikace zobrazí informace o úspěchu či nespěchu řešení daného úkolu. Pokud uživatel nějaký úkol úspěšně nevyřešil, bude mít možnost ho jednou znovu opakovat.

3.1.1 Úkoly

Čtyři typy úkolů jsou ještě rozděleny do dvou kategorií. V první z nich jsou ty, které procvičují uživatelovu znalost výslovnosti a v té druhé jsou úkoly procvičující znalost překladu.

3.1.1.1 Úkoly procvičující výslovnost

Úkol na procvičování výslovnosti je pouze jeden. Tento úkol uživateli zobrazí anglické slovo. Uživatel pak nahraje svoji výslovnost zobrazeného slova. Poté mu aplikace zkontroluje jeho nahranou výslovnost. Pokud byla nahraná výslovnost správná, aplikace uživatele informuje o úspěšném splnění úkolu. Pokud však výslovnost správná nebyla, aplikace uživateli ukáže jeho nahranou a správnou výslovnost slova. Tyto výslovnosti se pak porovnávají a aplikace uživateli ukáže, ve kterých místech byla jeho výslovnost nesprávná.

3.1.1.2 Úkoly procvičující překlad

První úkol procvičující překlad uživateli zobrazí české nebo anglické slovo X a k němu jeden správný překlad a čtyři nesprávné překlady. Uživatel pak bude

3. NÁVRH

mít za úkol zvolit správný překlad. Pokud zvolí nesprávný, aplikace mu zobrazí správnou odpověď spolu s doplňujícími informacemi ke slovu X (překlad, výslovnost).

Druhý úkol zobrazí uživateli 5 anglických slov a jejich 5 překladů. Uživatel má pak za úkol nalézt ke každému anglickému slovu jeho překlad.

Poslední úkol procvičující překlad, zobrazí uživateli anglické slovo a jeho překlad. Zobrazený překlad bude mít proházená písmena a úkolem pro uživatele bude nalézt správné pořadí písmen.

3.2 Případy užití

Z předešlé analýzy jsem vytvořil případy užití, které byly použity pro další návrh a následnou implementaci aplikace. Některé komplexnější případy jsem rozepsal do scénářů níže.

3.2.1 UC1 - Vytvořit kvíz

1. Příklad užití začíná po výběru skupiny slov S, oblasti jazyka a kritéria výběru slov
2. Aplikace získá podmnožinu anglických slov A ze skupiny S podle vybraného kritéria.
3. Aplikace vytvoří množinu B, která bude tvořena z překladů slov v podmnožině A.
4. Aplikace vytvoří různé druhy úkolů, kde každý bude mít jedno správné řešení. Druh úkolu je určován podle dříve vybrané oblasti jazyka. Úkoly jsou tvořeny ze slov obsažených v množinách A a B.
5. Aplikace vytvoří kvíz z úkolů výše a uloží ho.

3.2.2 UC2 - Hrát kvíz

1. Příklad užití začíná po vytvoření kvízu.
2. Aplikace vybere úkol z kvízu.
3. Vybraný úkol je zobrazen a uživatel ho začne řešit, respektive jeho část, podle daného popisu.
4. Uživatel odešle řešení, respektive část řešení.
5. Aplikace ověří správnost řešení, respektive část řešení.
6. Aplikace uloží skóre za poslané řešení, respektive poslanou část řešení.
7. Uživatel splnil úkol.

8. Aplikace informuje uživatele o úspěšném či neúspěšném splnění úkolu. Po neúspěšném splnění bude mít uživatel možnost daný úkol později jednou znovu opakovat. Pokud však uživatel úkol splnil úspěšně, pokračuje se bodem 2, dokud ještě zbývá úkol, který je potřeba splnit.
9. Aplikace zobrazí uživateli výsledné shrnutí odehraného kvízu. Shrnutí bude obsahovat úspěch tohoto a všech již odehraných kvízů dané skupiny. Dále pak uživateli zobrazí zlepšení, respektive zhoršení od minulých odehraných kvízů.
10. Uživatel bude moci opakovat tento kvíz a nebo ho opustit.

3.2.3 UC3 - Vybrat úkol k procvičování

1. Příklad užití začíná při spuštění kvízu.
2. Aplikace vyhledá v kvízu úkol, který ještě nebyl uživateli zobrazen. Pokud takový úkol nenalezne, pokračuje se bodem 3.
3. Aplikace vyhledá v kvízu úkol, který již byl jednou neúspěšně řešen.
4. Aplikace vrátí nalezený úkol, popřípadě informuje o tom, že již není žádný úkol k vyřešení

3.2.4 UC4 - Nahrát výslovnost

1. Příklad užití začíná, když uživatel hraje kvíz a řeší úkol, který procvičuje výslovnost.
2. Uživatel stiskne tlačítko pro nahrání výslovnosti.
3. Aplikace informuje uživatele o tom, že ho poslouchá.
4. Uživatel nahraje svou výslovnost slova z úkolu.
5. Aplikace uloží nahranou výslovnost.

3.2.5 UC5 - Rozpoznat výslovnost

1. Příklad užití začíná, po nahrání uživatelské výslovnosti.
2. Aplikace se pokusí rozpoznat uživatelskou výslovnost.
3. Aplikace vrátí seznam nejpravděpodobněji řečených slov. Tento seznam bude seřazen sestupně podle pravděpodobnosti, tj. nejpravděpodobější slovo bude jako první na seznamu.
4. Aplikace získá výslovnost prvního slova ze seznamu a vrátí ji.

3. NÁVRH

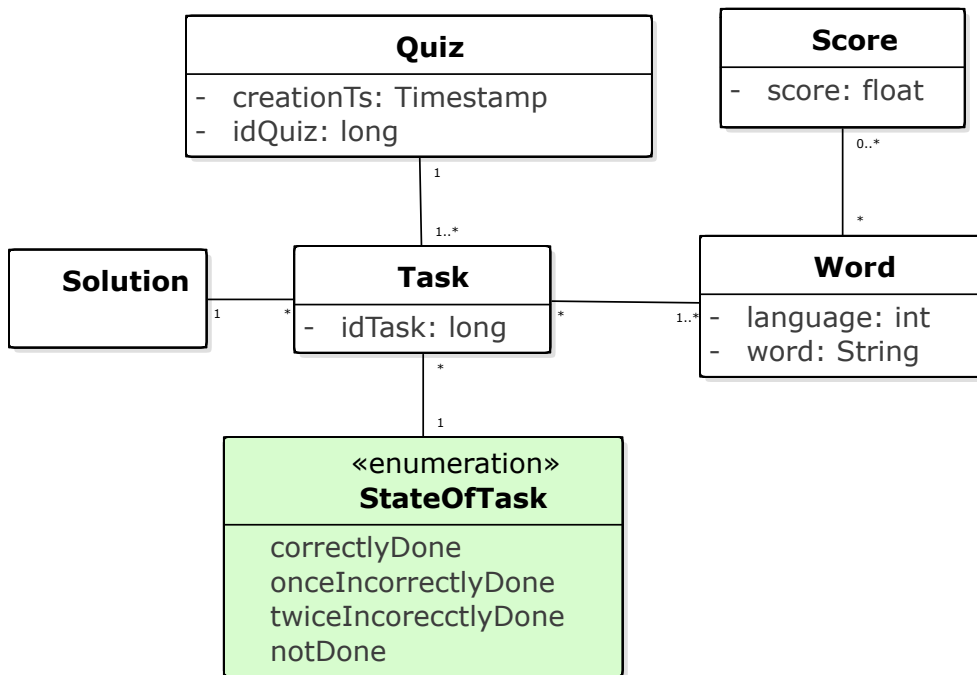
- alternativní scénář:
 1. Scénář začíná, pokud se aplikaci nepodaří rozpoznat uživatelskou výslovnost.
 2. Aplikace informuje uživatele o tom, že selhalo rozpoznání jeho nahrané výslovnosti.
 3. Uživatel znovu nahraje svoji výslovnost.
 4. Scénář dále pokračuje 1. krokem hlavního scénáře.

3.2.6 UC6 - Přehrát výslovnost

1. Příklad užití začíná, když si chce uživatel přehrát výslovnost slova.
2. Aplikace uživateli zobrazí ikonu u anglických slov.
3. Uživatel klikne na ikonu u slova.
4. Aplikace přehraje výslovnost daného slova.

3.3 Doménový model

Doménový model níže popisuje základní entity aplikace a vztahy mezi nimi. Základní entitou je Kvíz, který je složen z úkolů. Úkoly jsou tvořené Slovy, které pak uživatel procvičuje v rámci řešení daného úkolu. Každý úkol má pak jedno možné Řešení. V průběhu řešení úkolu uživatel získává skóre pro slova obsažená v právě procvičovaném úkolu.

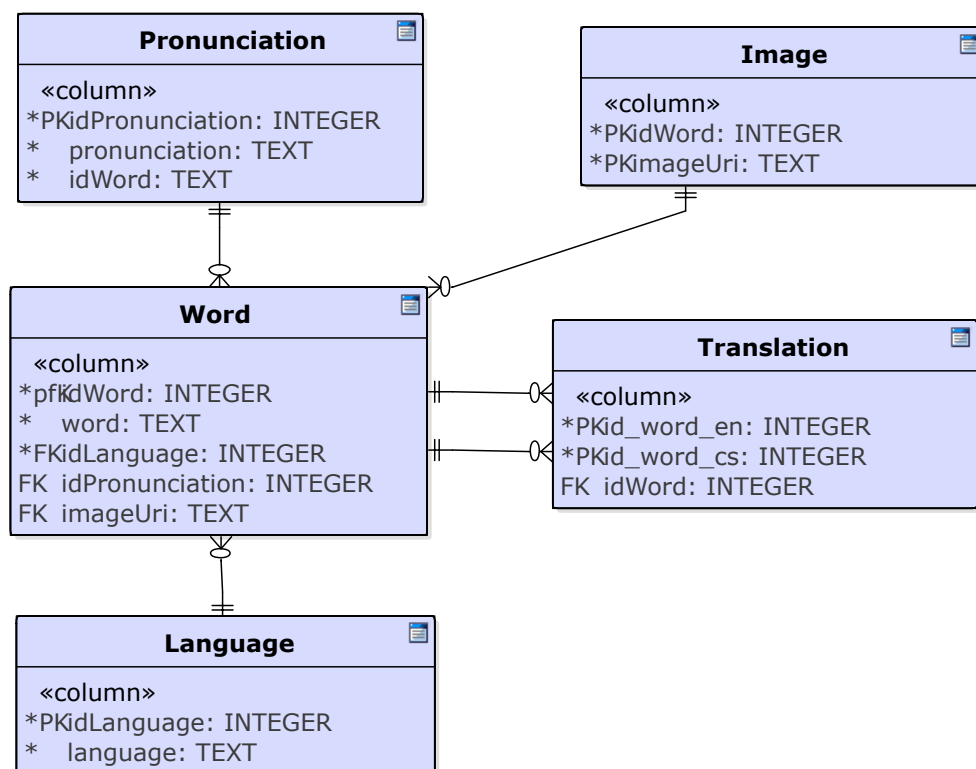


Obrázek 3.1: doménový model

3.4 Databázový model

V databázovém modelu níže jsou zobrazeny základní entity vytvořeny ze zdrojů dat a jejich vzájemné vztahy.

3. NÁVRH



Obrázek 3.2: databázový model

3.5 Agregace dat

Všechna data potřebná pro tuto aplikaci, která se získají ze zdrojů popsaných v analýze, je potřeba spojit dohromady a výsledkem naplnit databázi.

Výsledek spojení dat bude jeden soubor, který bude obsahovat anglická slova, jejich překlady a výslovnosti. Výsledný soubor bude obsahovat na jednom řádku 3 položky: anglické slovo, všechny jeho překlady a všechny jeho výslovnosti. Tyto položky pak budou odděleny středníkem.

Datová množina překladů je jako slovník, kde jeden řádek obsahuje anglické slovo a jeden jeho překlad. Jeho řádky se sloučí, tak aby bylo anglické slovo na každém řádku unikátní. Po sloučení se tento soubor použije jako výsledný soubor do kterého se doplní výslovnosti. Tímto souborem pak bude naplněna databáze.

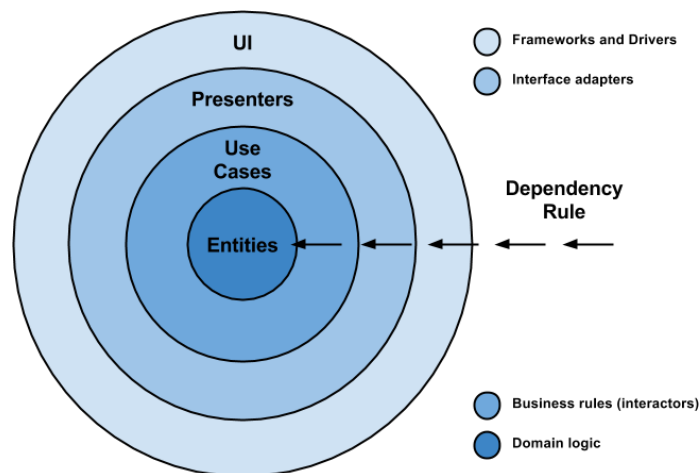
3.6 Architektura

Android Library poskytuje mnoho volnosti v rozhodování, jak navrhnout aplikaci pro Android. Tato volnost však může vést k nekonzistentním názvům komponent, velkému množství tříd nebo také k absenci jakékoliv architektury. Z těchto problémů pak vznikne aplikace, která se bude těžce rozšiřovat, testovat a udržovat.[12]

Jako inspiraci pro vhodné použití architektonických konceptů a nástrojů jsem získal z repositáře[13] od Googlu, ve kterém demonstruje ve vzorových Android projektech, jak se vyhnout problémům řečeným výše.

3.6.1 Clean Architecture

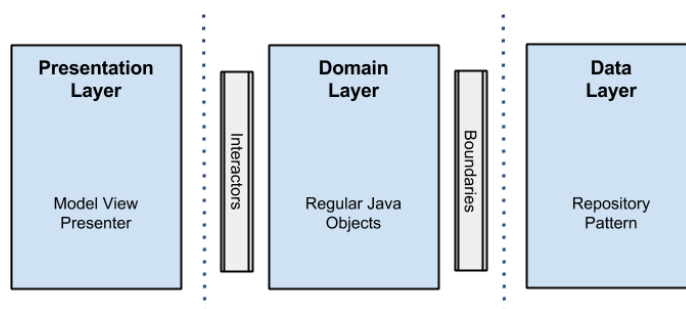
Pro tuto aplikaci jsem zvolil architekturu Clean Architecture[14]. V této architektuře je aplikace rozdělena do mnoha vrstev cibulového tvaru, jako je to znázorněno na obrázku.3.3



Obrázek 3.3: reprezentace Clean Architecture

Její hlavní účelem je oddělení odpovědností, tj. každá vrstva této architektury má svůj specifický účel a pracuje nezávisle na ostatních. Díky tomu se pak dosáhne dobré rozšiřovatelnosti a udržovatelnosti výsledné aplikace.

Dále pak tato architektura popisuje závislosti mezi jednotlivými vrstvami. Všechny závislosti mohou směřovat pouze z vnějších vrstev do vnitřních. Jelikož je vrstva z business logikou uprostřed, tak neví nic o vnějších vrstvách a díky tomu může být bussiness logika testována bez jakýchkoliv závislostí. Tím se pak hlavní funkcionalita může testovat izolovaně od zbytku aplikace.



Obrázek 3.4: návrh architektury

Mezi její nevýhody však patří velké množství kódu, který neprovádí žádnou hlavní funkcionalitu. Aby se dosáhlo výše zmíněné nezávislosti mezi vrstvami, musí každá z vrstev mít svůj vlastní datový model, takže pro odeslání datového modelu z jedné vrstvy do druhé se musí použít mapper, který přetransformuje jeden datový model do druhého.

3.6.2 Vrstvy aplikace

Aplikaci bude rozdělena do čtyř vrstev: doménovou, datovou, zařízení a prezentační.

3.6.2.1 Doménová vrstva

Doménová vrstva v sobě bude obsahovat veškerou business logiku, která se bude skládat z Interactorů (třídy obsahující funkcionalitu) a rozhraní, které budou sloužit ke komunikaci s vnějšími vrstvami. Tato vrstva nebude mít žádnou závislost na Android Library.

3.6.2.2 Datová vrstva

Všechna lokální nebo vzdálená data potřebná v aplikaci budou procházet přes tuto vrstvu. V datové vrstvě budou implementovány repository (rozhraní definované v doménové vrstvě). Repository se postarají o získání, respektive uložení dat z různých zdrojů.

3.6.2.3 Vrstva zařízení

Vrstva zařízení implementuje rozhraní z doménové vrstvy, pomocí kterých probíhá komunikace se zařízeními uvnitř mobilního telefonu. Zařízením je např. mikrofon nebo reproduktor.

3.6.2.4 Prezentační vrstva

Tato vrstva bude používat návrhový vzor Model View Presenter. View obsahuje pouze věci souvisejících s grafikou a nebude obsahovat žádnou logiku aplikace. Každému View je přiřazen Presenter, který komunikuje s business logikou přes Interactors, které provádí svojí práci na vlákne v pozadí. Díky tomu se zbytečně neblokuje hlavní vlákno a tak může stále reagovat na uživatelské akce (dotyky, ...). Interactor pak oznámí Presenterovi výsledek své práce a ten ji sdělí uživateli přes view.

3.6.3 Moduly aplikace

Aplikace bude rozdělena do 4 modulů. Každý modul bude reprezentovat jednu z vrstev zmíněných výše. Datový, prezentační modul spolu s modulem zařízení budou závislé na doménovém modulu.

3.6.4 Balíčky aplikace

Dobrá organizace kódu do balíčků je jedna z klíčových bodů dobré architektury. Struktura balíčku je totiž první věc se kterou se programátor setká při prohlížení zdrojového kódu.

Obecně se v javě rozlišují dva různé způsoby, jak organizovat strukturu balíčků a těmi jsou package by layer a package by feature.

3.6.4.1 Package by layer

Při package by layer každý balíček obsahuje položky, které spolu obvykle nesouvisí. Výsledkem je pak nízká soudržnost a modularita spolu s vysokou provázaností mezi jednotlivými balíčky. To však způsobí, že upravovat jednu funkcionalitu znamená hledat položky napříč mnoha balíčky a její smazání nelze téměř nikdy provést v jednom kroku.[15]

3.6.4.2 Package by feature

Způsob package by feature se pokouší umístit všechny položky související pouze s jednou funkcionalitou do jednoho balíčku. Výsledkem takového rozdělení je vysoká soudržnost a modularita spolu s nízkou provázaností mezi ostatními balíčky.[15]

Pro tuto aplikaci jsem použil právě tento způsob, protože díky vysoké soudržnosti uvnitř balíčku bude vše přehledně na jednom místě. Jako názvy funkcionalit budou použity funkční požadavky, tj. název balíčku bude např. x.y.z.createQuiz a tento balíček createQuiz pak bude obsažen ve více modulech.

3.6.5 Resources aplikace

Resources jsou přidáné soubory a statický obsah jako např. obrázky, layouty obrazovek, které používá aplikace a další. Těchto souborů může být velké množství, a proto budou všechny rozděleny do složek s názvem obrazovky. Díky tomu se pak urychlí hledání určitého resource.

3.7 Uživatelské rozhraní

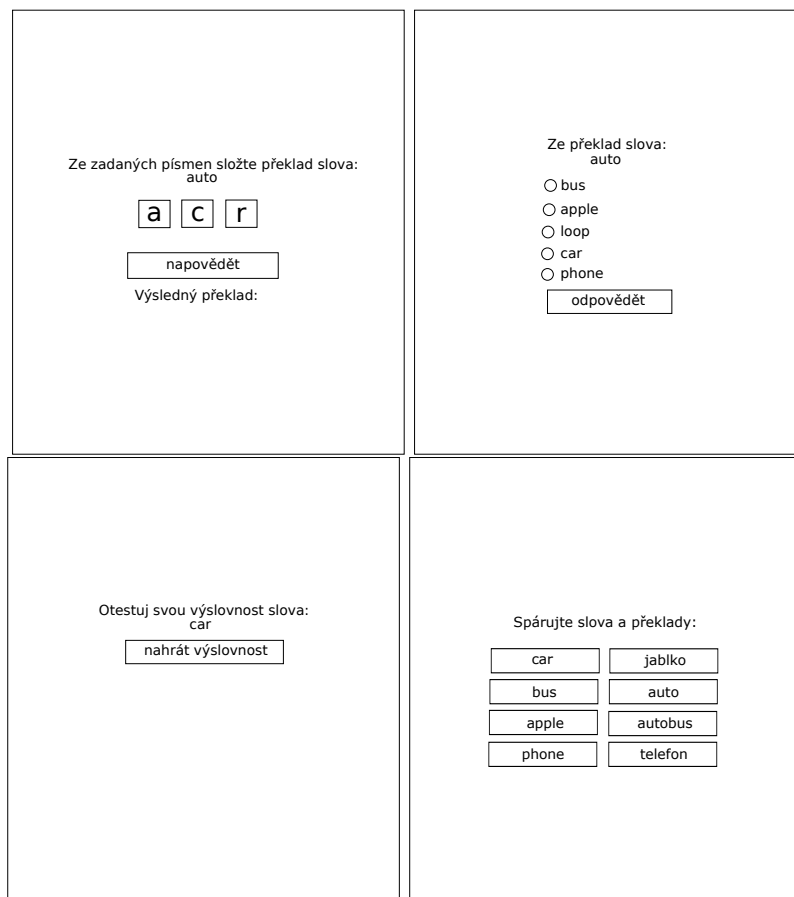
Uživatelské rozhraní pro aplikaci bylo navrženo tak, aby bylo dostatečně intuitivní a všechny jeho části byly responzivní, a aby bylo připraveno na různé typy displejů. Při návrhu pro Android se musí dávat pozor na velikost a rozestup jednotlivých prvků rozhraní - kvůli dotykovým displejům nesmí být příliš malé, aby nedocházelo k překlikům. Jednotlivé obrazovky popsané níže jsou namapovány na funkční požadavky:

3.7.1 Obrazovky k F4:

- Obrazovka O1: Na této obrazovce si budou moci uživatelé vybrat skupinu slov, kterou chtějí procvičit a kritérium, podle kterého se vytvoří kvíz. Po vybrání skupiny slov a kritéria se zobrazí dialog, ve kterém uživatel dostane možnost vybrat si oblast jazyka kterou chce procvičovat, tj. překlad, výslovnost, nebo obě najednou.

3.7.2 Obrazovky k F2:

- Obrazovka O2: Na této obrazovce se zobrazují úkoly. Po splnění úkolu se uživateli zobrazí dialog s informací o správnosti, resp. o nesprávnosti poslaného řešení. V případě nesprávného řešení se v dialogu objeví doplňující informace o slovech, které obsahuje daný úkol.



Obrázek 3.5: obrazovky úkolů

3.7.3 Obrazovky k F7 a F8:

- Obrazovka O3: uživateli zobrazí seznam skupin slov a tlačítko pro její vytvoření. Každá položka v seznamu bude obsahovat dvě ikony, které po kliknutí provedou akci pro skupinu slov. Ikona tužky bude sloužit pro úpravu skupiny a ikona popelnice pro smazání skupiny.
- Obrazovka O4: zobrazí seznam všech slov skupiny, kde každá položka bude obsahovat ikonu popelnice pro smazání daného slova. Pod seznamem je tlačítko pro přidání nových slov.
- Obrazovka O5: zobrazí seznam všech slov vyjma těch ze skupiny slov, které budou mít na každé položce checkbox pro vybrání slova. Uživateli se pak po kliknutí na tlačítko "přidat" přidají vybraná slova do skupiny.

3. NÁVRH

3.7.4 Obrazovky k F6:

- Obrazovka O6: zobrazí vstupní pole pro zadání jména pro novou skupinu slov. Pod tímto polem zobrazí seznam všech slov, kde každá položka bude obsahovat checkbox pro výběr slova. Dále pak obrazovka obsahuje tlačítko pro potvrzení vytvoření skupiny.

3.7.5 Obrazovky k F5:

- Obrazovka O7: zobrazí seznam anglických slov, obsažených ve skupině a jejich všechny překlady. Po kliknutí na tlačítko detail se zobrazí obrazovka s detailem.
- Obrazovka O8: zobrazí detail slova, tj. všechny dostupné informace o daném slově. Pokud některé informace dostupné nejsou, jako např. překlad, tak se nezobrazí žádný seznam.

3.7.6 Obrazovky k F3:

- Obrazovka 9: Obrazovka bude sloužit pro zobrazení statistiky. V horní části obrazovky je menu s dvěma tlačítky pro různé statistiky: celkovou a skupin slov. Statistika skupin slov zobrazí uživateli nejprve seznam všech skupin, kde si vybere jednu u kterých chce zobrazit statistiku. Uživateli se pak zobrazí úspěchy v odehraných kvízech dané skupiny pro obě oblasti jazyka. U celkové statistiky aplikace zobrazí celkové úspěšnosti skrze všechny odehrané kvízy.

3.8 Technologie pro zjednodušení vývoje

Níže je uvedený seznam obsahující oblasti, ve kterých se zjednoduší samotný vývoj a technologie, které pro to budou určeny.

1. Mapování objektů a dotazování se databáze
2. Závislosti mezi komponentami a jejich testování
3. Multithreading
4. Manipulace s obrázky

3.8.0.1 Mapování objektů a dotazování se databáze

Jelikož musí aplikace fungovat bez internetového připojení, viz nefunkční požadavek N1, tak budou muset být data uložena v lokální databázi. Je tedy potřeba zvolit vhodný ORM framework, který zjednoduší dotazování do takové databáze a zároveň se postará o automatické namapování záznamů do objektů.

Jako ORM framework jsem zvolil Room od Google, který splňuje všechny vlastnosti popsané výše.

3.8.0.2 Závislosti mezi komponentami a jejich testování

K vyřešení těchto problémů se použije vhodný Dependency Injector DI, který se postará o eliminaci nebo alespoň zmenšení nezbytných závislostí. Zároveň zvýší testovatelnost komponent, jelikož se pomocí DI mohou závislosti vložit do určitých komponent, tak je možné jednoduše vytvořit mock implementaci těchto závislostí a tím otestovat jednu komponentu nezávisle na celé aplikaci.[16]

Jako Dependency Injector jsem zvolil Dagger 2 od Googlu. Tento DI generuje všechny potřebný kód k jeho fungování a nepoužívá reflexi, která je obecně pomalá[17], zvláště pak pro mobilní zařízení, a proto se tento DI hodí na vývoj aplikací pro Android.

3.8.0.3 Multithreading

Při vývoji aplikací pro Android si musí dát programátor pozor na přetížení hlavního vlákna aplikace, které se primárně stará o vykreslování uživatelského rozhraní a jeho interakci s aplikací. Pokud se hlavní vlákno přetížá, tak daná aplikace zamrzne - tj. nebude vůbec reagovat na uživatelské akce, které provádí na obrazovce.

Aby se vyhnulo přetížení hlavního vlákna, je zapotřebí převést určité výpočetní operace do jiného vlákna. V této aplikaci se bude tedy každý Interactor z doménové vrstvy provádět na odlišném vlákně.

Pro jednodušší práci s vlákny se použije RxJava, což je implementace Reactive Extensions[18].

3.8.0.4 Manipulace s obrázky

V jednom typu úkolu z kvízu si budou moci uživatelé místo textových odpovědí zobrazovat obrázky vystihující daný text, tj. pokud aplikace zjistí, že má obrázek přiřazený k určitému slovu, tak ho uživateli zobrazí. Pokud ne, zobrazí mu text.

V aplikaci tedy bude nutné použít framework, který dokáže transformovat obrázek do určitých rozměrů a zachovat přitom jeho poměr mezi výškou a šířkou.

Pro tento účel jsem zvolil Glide, který dokáže transformovat obrázky, ale stará se i o jejich cachování. Glide ukládá obrázky po jejich transformaci do interní paměti mobilního zařízení, takže až si aplikace vyžádá zobrazení toho samého obrázku, tak už si ho vezme z interní paměti a nebude ho muset znovu transformovat do zobrazitelné podoby.

Implementace

4.1 Implementace prototypů

Nejprve jsem vytvořil dva prototypy pro rozpoznávání hlasu a poté jeden prototyp pro převod textu na mluvenou řeč. U prototypu bylo potřeba zvolit nejvhodnější jazykový modul. Všechny prototypy vycházely z ukázkových projektů od tvůrců frameworků.

Každý z prototypů byl otestován a podle jejich výsledků se rozhodlo, jaký framework se použije v aplikaci.

4.1.1 Testování prototypů pro rozpoznávání řeči

Oba prototypy po zapnutí čekají na uživatelský hlas. Po získání hlasového vstupu vrací seznam slov. Tento seznam obsahuje slova, která nejspíše řekl uživatel na vstup. Každému slovu je přiřazena jeho pravděpodobnost správnosti. Z tohoto seznamu prototypy vyberou nejpravděpodobnější slovo a jemu naleznou odpovídající výslovnost ve slovníku cmudict a tu vypíše na standardní výstup.

Vstupem pro testy byl seznam slov zobrazený níže, který byl složený z jednoslabičných až čtyřslabičných slov a dále dvě zvukomalebná. Test ověřoval pouze správné rozpoznání řečených slov ze seznamu prototypem. Testy byly opakovány u obou prototypů třikrát. Výsledky jsou zobrazeny v tabulkách níže.

Seznam testovaných slov:

1. through - jednoslabičné slovo
2. teacher - dvouslabičné slovo
3. umbrella - tříslabičné slovo
4. helicopter - čtyřslabičné slovo

4. IMPLEMENTACE

5. year - zvukomalebné slovo se slovem ear

6. ear - zvukomalebné slovo se slovem year

4.1.1.1 Test Google Voice Recognition

slovo	rozpoznáno
through	ano
teacher	ano
umbrella	ano
helicopter	ano
year	ano
ear	ne

Tabulka 4.1: Výsledky testu pro Google Voice Recognition

4.1.1.2 Test PocketSphinx

slovo	rozpoznáno
through	ano
teacher	ano
umbrella	ne
helicopter	ne
year	ano
ear	ne

Tabulka 4.2: Výsledky testu pro PocketSphinx

4.1.1.3 Zhodnocení testů

Z testů jasně vyplynulo, že Google Voice Recognition dosahuje lepších výsledků než PocketSphinx, proto bude v aplikaci použit Google Voice Recognition.

4.1.2 Prototyp pro převod textu do mluvené řeči

Pro prototyp sloužící k převádění textu do mluvené řeči jsem použil Text-To-Speech demo od Googlu. Toto demo na vstup vyžaduje text a na výstup přehraje zvuk jako mluvenou řeč. V prototypu jsem porovnal jazykové moduly od Googlu a Pico.

Po otestování obou dvou modulů jsem se rozhodl použít ten od Google, protože měl výrazně reálnější hlasový projev než modul od Pico.

4.2 Implementace aplikace

Po vybrání vhodného frameworku pro rozpoznávání řeči a vhodného modulu pro převod textu do mluvené řeči se naimplementovala aplikace podle návrhu. Náhledy obrazovek z implementované aplikace se nachází v příloze B.

Testování

Testování aplikace probíhalo ve dvou fázích. V první z nich se testovalo podle testovacích scénářů (TS), které ověřovali splnění funkčních požadavků a v té druhé se testovalo uživatelské rozhraní. Obě fáze proběhly na více zařízeních (emulátorech a reálných zařízeních). Zařízení byly vybírány podle verze Androidu, rozlišení a zahuštění pixelů na obrazovce.

5.1 Testování funkčních požadavků

- TS1 pro F4: Zvolte oblast jazyka a kritérium pro výběr slov. Vyzkoušejte všechny kombinace.
 - Očekávaný výsledek: Uživatel si mohl navolit všechny kombinace.
 - TS2 pro F1, F2, F5: Vyberte skupinu pro procvičování. Spusťte kvíz a splňte všechny jeho úkoly. Po ukončení kvízu přejděte na obrazovku O5, kde si prohlédnete některá slova a jejich detaily.
 - Očekávaný výsledek: Uživatel úspěšně odehrál kvíz a poté si prohlédl slova právě procvičené skupiny.
 - TS3 pro F3: Přejděte na obrazovku O2. Prohlédněte si statistiky svých odehraných kvízů.
 - Očekávaný výsledek: Uživateli se zobrazily statistiky jeho odehraných kvízů.
 - TS4 pro F6: Přejděte na obrazovku O3. Stiskněte tlačítko pro vytvoření kvízu. Vyplňte jméno skupiny. Zvolte aspoň 6 slov, které budou přidány do nově vytvořené skupiny. Stisknutím tlačítka potvrďte vytvoření skupiny.
 - Očekávaný výsledek: Uživatel úspěšně vytvořil novou skupinu, která se mu zobrazí v seznamu všech skupin.

5. TESTOVÁNÍ

- TS5 pro F7 a F8: Přejděte na obrazovku O6. Zvolte libovolnou skupinu ze seznamu. Smažte z této skupiny dvě slova. Smazaná slova do ní opět přidejte.
- Očekávaný výsledek: Uživatel úspěšně smazal slova ze skupiny a poté je tam úspěšně přidal.

Po úspěšném provedení těchto scénářů se ověřilo naplnění všech funkčních požadavků. Při provádění testů se však ukázalo, že na zařízení s Androidem verze 5.1.1 je nastavena původní hodnota pro dobu čekání na další slova v Google Voice Recognition příliš vysoká. Uživatel tak může nabýt dojmu, že se nic nestalo a znovu se pokusí nahrát svoji výslovnost.

5.2 Testování uživatelského rozhraní

U těchto testů se kontrolovalo, jestli se na různých displejích budou vhodně zobrazovat všechny položky uživatelského rozhraní - tj. jestli jsou položky dostatečně velké pro dotyky a zároveň aby mu nějaká z položek nezmizela z obrazovky, např. při přetočení displeje.

Po provedení testů se zjistil problém se zobrazením položek na malých displejích, kde text u některých položek zmizel z obrazovky místo toho, aby se text zalomil. Tato chyba byla hned po zjištění opravena.

Možné vylepšení aplikace

Aplikace v budoucnu nemusí podporovat výuku pouze anglického jazyka, ale může se rozšířit i na jiné jazyky. K jiným jazykům by stačilo nalézt slova, jejich překlady, výslovnosti a obrázky, které je vystihují.

Dalším možným zlepšením je zjednodušit tvorbu skupiny, která má velké množství slov a to tím, že by si uživatel nahrál textový soubor s množinou slov do aplikace a ta by mu z těch slov vytvořila skupinu, tj. uživatel by si nemusel všechna slova přidávat po jednom. K tomu by se dal využít již vytvořený program, který byl použit dříve pro nahrání dat do databáze, jelikož už v sobě tuto funkcionalitu obsahuje.

Závěr

Vytvořená aplikace slouží ke zjednodušení výuky anglického jazyka, kde si její uživatelé mohou zlepšit pomocí kvízů svůj překlad, poslech a výslovnost slov. Výhodou oproti jiným výukovým aplikacím je, že ke svému fungování nepotřebuje internetové připojení a dají se v ní procvičovat výše zmíněné oblasti jazyka najednou a uživatel tak nemusí používat více aplikací.

V první fázi této práce proběhla analýza, kde se nejprve analyzovala konkurence, ve které se detailně popsala funkcionalita aplikací sloužící pro výuku různých oblastí jazyka. Dále se z cílů a předešlé analýzy konkurence vytvořily funkční a nefunkční požadavky. Z nefunkčního požadavku N1 vyplynulo, že je třeba mít všechna data jako jsou anglická slova, jejich překlady a výslovnosti uložena přímo v aplikaci. Analyzovaly se tedy zdroje, ze kterých se dají získat taková data. Poté se zanalyzovaly technologie potřebné pro splnění funkčních požadavků F1 a F3.

Další fáze se věnovala návrhu, kde se nejdříve popsal kvíz jako důležitá komponenta aplikace. Popsalo se, co přesně bude tento kvíz a z jakých částí se bude skládat. Dále pak se určily případy užití, které byly vytvořeny z funkčních požadavků. K některým komplexnějším případům se napsaly scénáře. Poté se navrhly entity a databáze, jejímž výsledkem byl doménový a databázový model. Dále se rozhodlo, že všechna data získaná z různých zdrojů budou sloučena do jednoho souboru a následně pak nahrána do databáze. Poté se určila Clean Architecture jako architektura, která se bude dodržovat při vývoji, díky které pak výsledná aplikace bude snadno rozšiřitelná, udržovatelná a testovatelná.

Ještě před samotnou implementací se vytvořily prototypy, testující technologie pro rozpoznávání hlasu a převod textu to mluvené řeči, ze kterých se následně zvolil framework, který se použije v aplikaci. Poté se vytvořila prázdná databáze, která bude při implementaci aplikace vložena do aplikace. Dále pak byl vytvořen program, který spojil všechna potřebná data pro aplikaci do jediného souboru a následně tyto data nahrál do databáze.

Po vybrání frameworků popsaných výše a nahrání dat do databáze se

ZÁVĚR

naimplementovala aplikace podle návrhu.

Po implementaci probíhalo testování, které se skládalo ze dvou částí. V první části se otestovalo splnění funkčních požadavků a v té druhé se otestovalo uživatelské rozhraní.

Z výsledků z testování se ověřilo, že aplikace splňuje všechny funkční požadavky a nefunkční požadavky - tj. i cíle této práce.

Literatura

- [1] GROUP, T.: Speak English Pronunciation. <https://play.google.com/store/apps/details?id=com.tflat.phatamtienganh>, 2017.
- [2] ELSA: ELSA Speak: Reduce your accent. <https://play.google.com/store/apps/details?id=us.nobarriers.elsa>, 2017.
- [3] Learn, F. E.: English Fun Easy Learn. <https://play.google.com/store/apps/details?id=com.funeasylearn.english>, 2017.
- [4] S.L., L. C.: Learn English Vocabulary. <https://play.google.com/store/apps/details?id=com.funeasylearn.english>, 2017.
- [5] flashcards, E.: English vocabulary : flashcard. <https://play.google.com/store/apps/details?id=vn.magik.english.flash.card>, 2017.
- [6] Labs, Z.: Andy - English Speaking Bot. <https://play.google.com/store/apps/details?id=com.pyankoff.andy>, 2017.
- [7] X-Apps: English Grammar and Vocabulary. <https://play.google.com/store/apps/details?id=com.grammar.en>, 2017.
- [8] slovníky, S.: Svobodné slovníky. <https://github.com/svobodneslovniky/svobodneslovniky>, 2017.
- [9] cmusphinx: cmudict. <http://svn.code.sf.net/p/cmusphinx/code/trunk/cmudict/cmudict-0.7b>, 2017.
- [10] Cmusphinx: pocketsphinx-android-demo. <https://github.com/cmusphinx/pocketsphinx-android-demo>, 2017.
- [11] Google: Adding Voice Capabilities. *Google [online]*, Březen 2014, [cit. 2017-5-2]. Dostupné z: <https://developer.android.com/training/wearables/apps/voice.html>

- [12] Google: Android Architecture Blueprints. *Google [online]*, Srpen 2012, [cit. 2017-8-2]. Dostupné z: <https://github.com/googlesamples/android-architecture>
- [13] Google: todo-mvp. <https://github.com/googlesamples/android-architecture/tree/todo-mvp/>, 2017.
- [14] Bob, U.: The Clean Architecture. *The Clean Architecture [online]*, Srpen 2012, [cit. 2017-5-2]. Dostupné z: <https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>
- [15] Cejas, F.: Architecting Android...The evolution. *Fernando Cejas [online]*, Březen 2014, [cit. 2017-6-6]. Dostupné z: <https://fernandocejas.com/2015/07/18/architecting-android-the-evolution/>
- [16] Jenkov, J.: Dependency Injection Benefits. *jenkov [online]*, Srpen 2012, [cit. 2017-10-2]. Dostupné z: <http://tutorials.jenkov.com/dependency-injection/dependency-injection-benefits.html>
- [17] Netcyrax: The lost droid and the magic Dagger: an intro to dependency injection for Android. *medium [online]*, Srpen 2012, [cit. 2017-11-3]. Dostupné z: <https://medium.com/rocknnull/the-lost-droid-and-the-magic-dagger-an-intro-to-dependency-injection-for-android-c686f4399117>
- [18] ReactiveX: ReactiveX. *ReactiveX [online]*, Srpen 2012, [cit. 2017-10-2]. Dostupné z: <http://reactivex.io/>

Seznam použitých zkratk

API Application Programming Interface

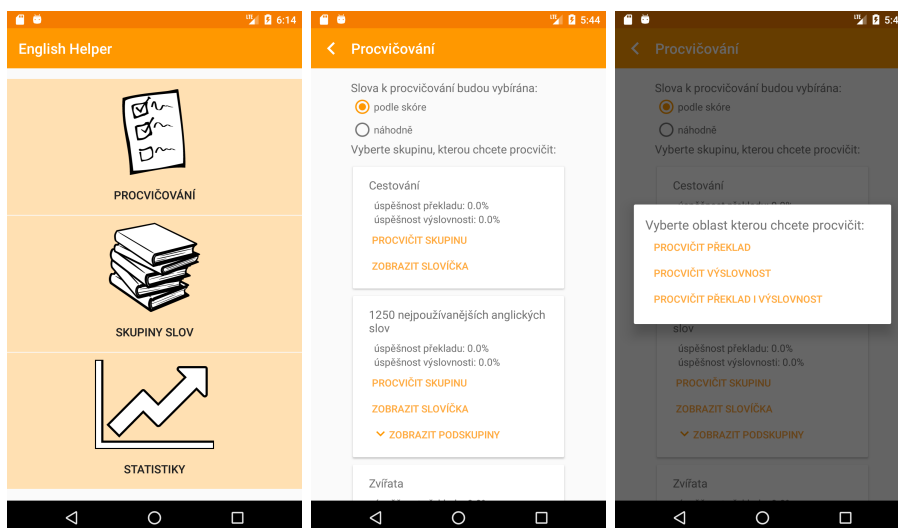
BNC British National Corpus

CEFR Common European Framework of Reference

ORM Object-relational mapping

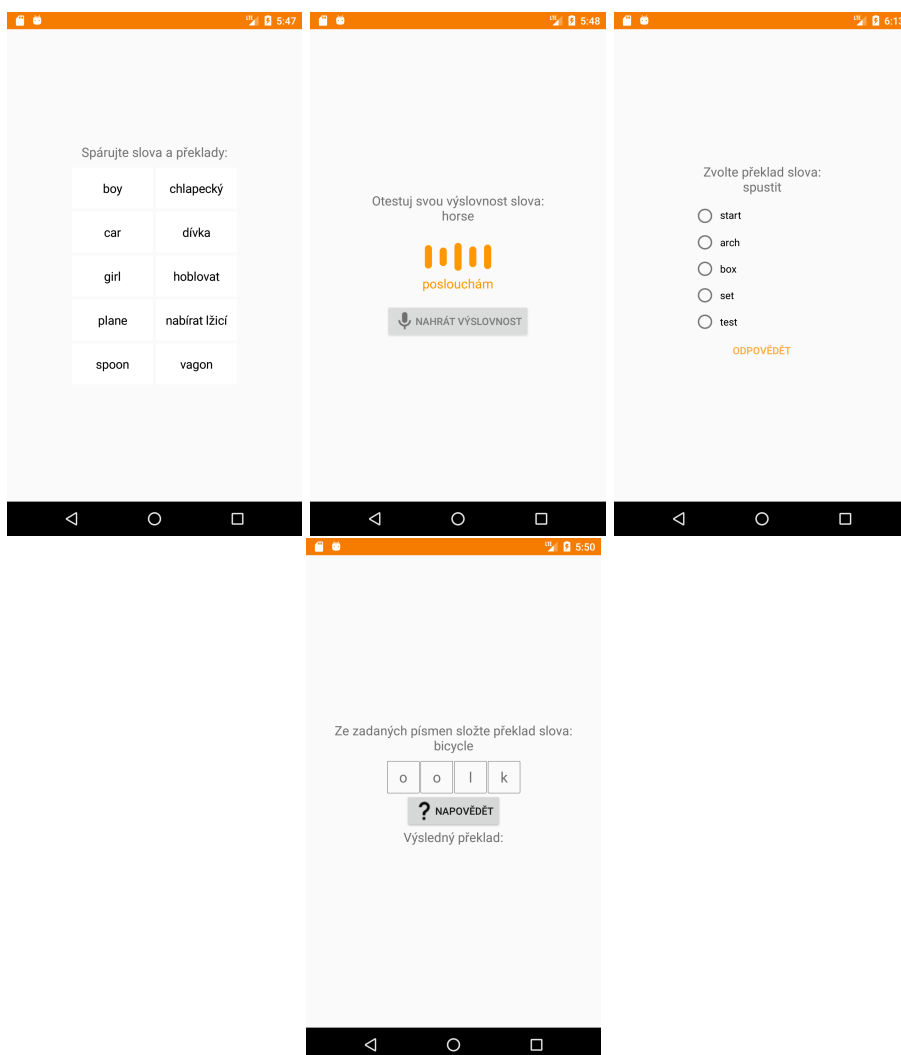
DI Dependency Injector

Náhledy obrazovek z výsledné aplikace

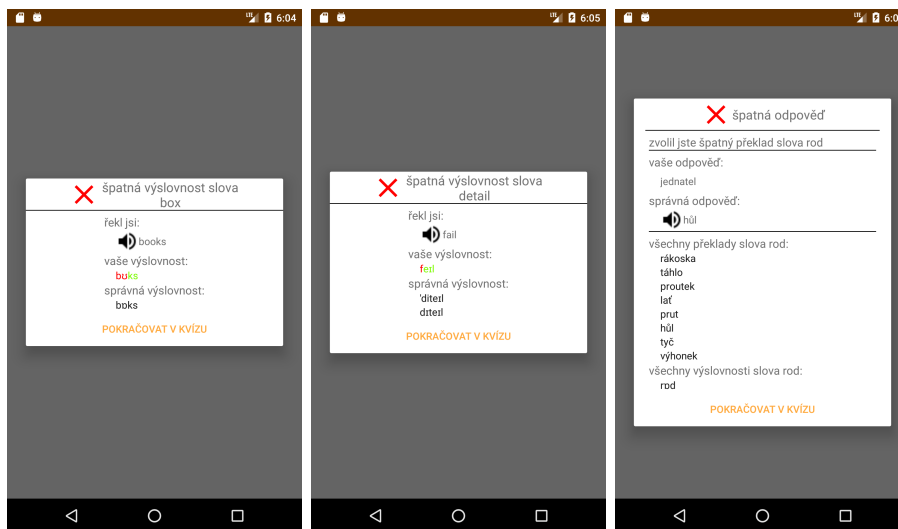


Obrázek B.1: obrazovky pro vybrání typu procvičování spolu s hlavní obrazovkou

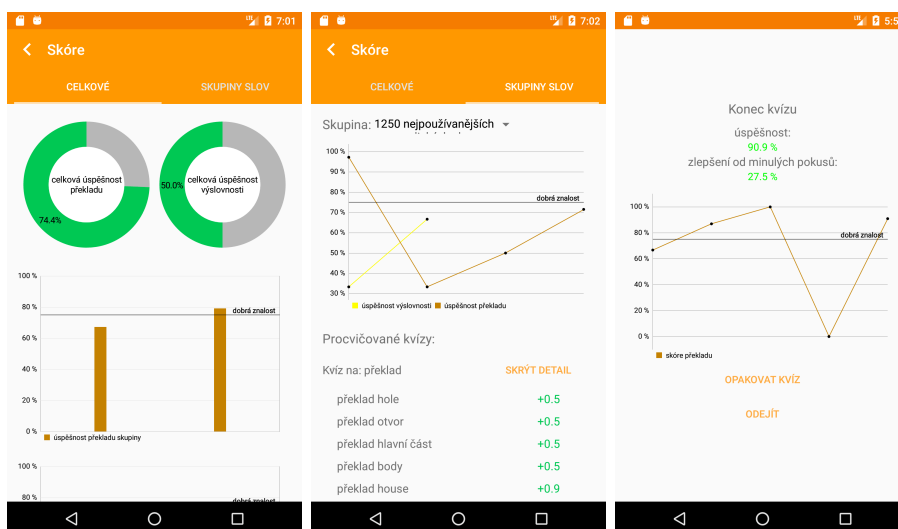
B. NÁHLEDY OBRAZOVEK Z VÝSLEDNÉ APLIKACE



Obrázek B.2: obrazovky úkolů

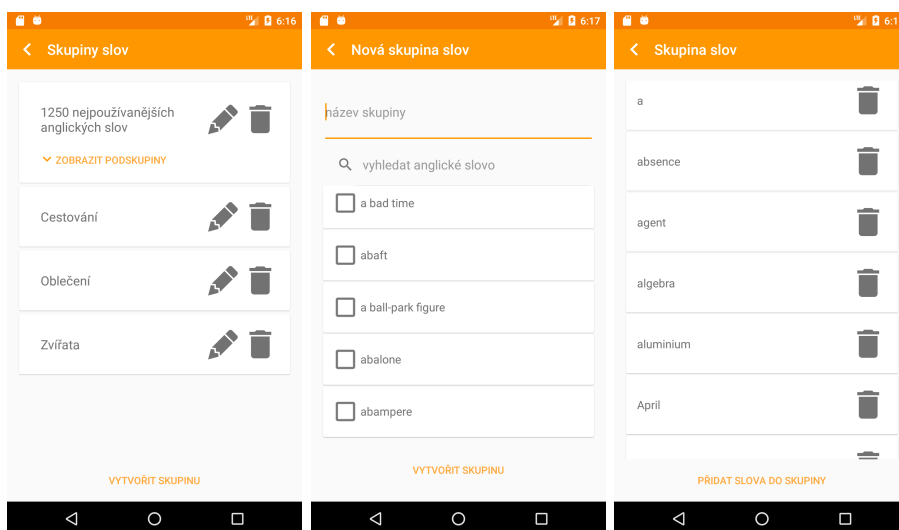


Obrázek B.3: obrazovky přechodů mezi úkoly



Obrázek B.4: obrazovky statistik a přehled odehraného kvízu

B. NÁHLEDY OBRAZOVEK Z VÝSLEDNÉ APLIKACE



Obrázek B.5: obrazovky pro editaci a tvorbu skupin

Instalační manuál

Aplikace lze naistalovat na mobilní zařízení s Androidem od verze 4.4. Pro instalaci výsledné aplikace postupujte podle návodu:

1. Přejděte ve svém mobilním zařízení do nastavení.
2. Nalezněte v nastavení sekci se zabezpečením a otevřte ji.
3. V této sekci povolte instalaci aplikací z neznámých zdrojů
4. Nahrajte aplikaci zabalenou jako .apk do svého zařízení. Nahrát aplikaci můžete dvěma způsoby:
 - Otevřete internetový prohlížeč a přejděte na stránku <https://goo.gl/3Bw3qR>, kde si stáhněte aplikaci do svého zařízení.
 - Z obsahu přiložené CD získáte soubor EnglishHelper.apk, který nahrajte do svého zařízení.
5. Vyhledejte právě nahraný soubor.
6. Po kliknutí na soubor EnglishHelper.apk se objeví dialog, který vám nabídne možnost instalace.
7. Zvolte možnost instalace.
8. Dále pokračujte podle instrukcí zobrazujících se při instalaci.

Obsah přiloženého CD

EnglishHelper.zip.....	zdrojové kódy implementace
EnglishHelper.apk.....	instalační balíček
thesis.pdf	text práce ve formátu PDF