Bakalářská práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Adaptace mobilní aplikace na základě zvukových vstupů

Dominik Brázdil

Studijní program: Otevřená informatika

Obor: Software

Červen 2019

Vedoucí práce: Ing. Jiří Šebek

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Brázdil** Jméno: **Dominik** Osobní číslo: **466281**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Studijní obor: **Software**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Adaptace mobilní aplikace na základě zvukových vstupů

Název bakalářské práce anglicky:

Pokyny pro vypracování:

1) Nastudujte různé možnosti zachycování zvuku v mobilu a jeho následného zpracování. [1,2,3]
2) Rozpracujte různé scénáře využití kontextových informací.
3) Navrhněte mobilní aplikaci, která bude měnit svou strukturu komponent na okolního ruchu, zvuku a jiných kontextových informacích.
4) Aplikaci otestuje na uživatelích a získejte zpětnou vazbu.
5) Vyhodnoťte spokojenost uživatelů a použitelnost aplikace.

Seznam doporučené literatury:

[1] Šebek, J. - Richta, K.: Usage of Aspect-Oriented programming in Adaptive Application Structure. New Trends in Databases and Information Systems: ADBIS 2016 Short Papers and Workshops, BigDap, DCSA, DC, Prague, Czech Republic, August 28-31, 2016, Proceedings
[2] Sohaib Ikram, Hafiz Malik: “Digital audio forensics using background noise“, Proceedings of the 2010 IEEE International Conference on Multimedia and Expo, ICME 2010, 19-23 July 2010, Singapore
[3] X. Lei, A. Senior, A. Gruenstein, and J. Sorenson, "Accurate and compact large vocabulary speech recognition on mobile devices," in Proc. Interspeech, 2013

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jiří Šebek, kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.02.2019** Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Jiří Šebek podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání	Podpis studenta

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Poděkování / Prohlášení

Děkuji vedoucímu své bakalářské práce Ing. Jiřímu Šebkovi za čas věnovaný udržování správného směru této práce a za praktické rady usnadňující její vypracování. Díky patří také mým rodičům, kteří se zasloužili o to, že jsem tuto práci mohl psát, a podporovali mě v tom.

Prohlašuji, že jsem následující práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze 20. 5. 2019

.....

Abstrakt / Abstract

Tato bakalářská práce je zaměřena na problematiku využívání zvukových informací z okolí člověka ke zlepšení jeho interakce se zařízením. Hlavní důraz je kladen na analýzu obecného uplatnění zvuku v mobilním zařízení a na jeho užitečnost při adaptaci uživatelského rozhraní. Dále pak je provedena rešerše dostupných technologií pro zpracování zvuku v mobilních zařízeních. Cílem práce je ověření dvou hypotéz. První z nich se zajímá o to, zda je samovolné vyjadřování uživatele možné použít pro detekci špatně navrženého rozhraní a pro jeho následné opravy. Druhá hypotéza zkoumá změnu kvality ovladatelnosti rozhraní při použití zvukové zpětné vazby uživateli. Součástí práce je návrh adaptivního systému a jeho implementace ve formě mobilní aplikace pro operační systém Android, která pro svou adaptaci využívá zpracování lidské řeči a analýzu jejího sentimentu. Systém také uživateli usnadňuje své ovládání poskytováním zvukové zpětné vazby při průchodu menu aplikace.

Klíčová slova: zvukový kontext, adaptivní uživatelské rozhraní, rozpoznávání řeči, analýza sentimentu, earcons

This bachelor thesis focuses on the problematics of using sound information in the surroundings of a person to improve his interaction with a device. The main focus is put on an analysis of general sound usage in mobile phones and its usefulness for user interface adaptation. Furthermore, research on available sound processing technologies in mobile devices is conducted. The aim of this work is to prove two hypotheses. First one is investigating if spontaneous user expressions can be used for detecting errors in interface design and for its subsequent repairs. The other examines the change in manageability of the interface while providing sound feedback to the user. The work also includes adaptation system design and its implementation in the form of a mobile application for Android that uses sound recognition and sentiment analysis for self-adaptation. The system also provides sound feedback expressing menu levels in order to ease user its control.

Keywords: audio context, adaptive user interface, speech recognition, sentiment analysis, earcons

Obsah /

1 Úvod . . . 1
1.1 Motivace . . . 1
1.2 Cíle . . . 2
2 Rešerše . . . 3
2.1 Uživatelské rozhraní . . . 3
2.2 Adaptace rozhraní . . . 3
2.2.1 Přístupy k adaptaci . . . 4
2.2.2 Prvky adaptivního přístupu [3] . . . 4
2.3 Kontext . . . 5
2.4 Audio kontext . . . 6
2.5 Veličiny extrahované ze zvuku . . . 6
2.6 Řeč . . . 7
2.6.1 Speech enhancement . . . 7
2.6.2 Speech recognition . . . 9
2.6.3 Sentiment analysis . . . 11
2.7 Ostatní zvuky . . . 12
2.7.1 Odstranění řeči . . . 13
2.7.2 Environmental sound recognition . . . 13
2.8 Přehrávání zvuku . . . 14
2.8.1 Text-to-speech synthesis [9] . . . 14
2.8.2 Auditory icons and earcons . . . 16
3 Analýza a související práce . . . 17
3.1 Možnosti využití zvuku na platformě android . . . 17
3.1.1 Environmental sound recognition . . . 17
3.1.2 Speech recognition . . . 19
3.1.3 Sentiment analysis . . . 20
3.2 Zvolené přístupy . . . 22
3.3 Menu na platformě android . . . 24
4 Návrh . . . 26
4.1 Architektura . . . 26
4.2 Rozdělení komponent . . . 27
4.3 Procesy programu . . . 29
4.3.1 Vytváření menu . . . 29
4.3.2 Výběr tlačítka . . . 30
4.3.3 Rozpoznávání řeči . . . 30
5 Implementace . . . 32
5.1 Nepřetržité rozpoznávání řeči . . . 32
5.2 Vytváření a úprava menu . . . 33
5.3 Vracení se v menu . . . 34
5.4 Adaptace . . . 35
5.5 Přehrávání earconů . . . 36
6 Testování . . . 37
6.1 Test rozpoznávání sentimentu a adaptace . . . 37
6.1.1 Výsledky a odpovědi . . . 38
6.2 Test zvukové zpětné vazby . . . 39
6.2.1 Výsledky a odpovědi . . . 40
7 Závěr . . . 42
8 Budoucí práce . . . 44
A Slovník zkratek . . . 45
B Ukázky kódu . . . 46
C Reference . . . 47
D Obsah CD . . . 52

Tabulky / Obrázky

6.1. Tabulka hlasových reakcí.....38
6.2. Tabulka odpovědí 1 38
6.3. Tabulka odpovědí 2 40

2.1. Adaptační přístupy 5
2.2. Zvýrazňování řeči.....8
2.3. Rozpoznávání řeči 10
2.4. Odstranění řeči 13
3.1. Environemtal sound recognition 18
3.2. Kontextuální lexikony 22
3.3. Typy menu.....23
3.4. Android menu 24
4.1. Model komponent 28
4.2. Sekvenční diagram vytváření menu 29
4.3. Sekvenční diagram výběru tlačítka.....30
4.4. Sekvenční diagram rozpoznávání řeči 31
5.1. Formát menu 34
6.1. Graf dob přístupu 40

Kapitola 1
Úvod

1.1 Motivace

Mobilní telefony, zvláště chytré mobilní telefony, se stávají rok od roku větším trendem a jejich vlastnictví je dnes již ve spoustě zemí samozřejmostí. V roce 2016 vlastnilo mobilní telefon 63% populace, z čehož přibližně 40% se týká chytrých telefonů. Toto zastoupení neustále roste a pro rok 2019 bylo odhadnuto, že majitelů chytrých telefonů bude 2,7 miliardy [6]. S tímto trendem také souvisí přecházení většiny služeb do oblasti mobilních aplikací a webů, jejichž poměr užívání vedou s 90% mobilní aplikace [7].

Z těchto dat vyplývá potřeba přikládat mobilním telefonům více pozornosti ve všech ohledech, jejich neustálému vývoji a úpravám podle požadavků uživatelů. Tyto úpravy se často zaměřují na usnadnění uživateli práci se zařízením a jejich vzájemnou komunikaci. Tato komunikace běžně probíhá přes uživatelské rozhraní (UI) telefonu nebo aplikace. Jedním z problémů, jaké úpravy pro usnadnění komunikace zvolit, je mnohdy už zjišťování jejich potřeby, protože pokud vše funguje, jak má, uživatel často neví, že by nějakou změnu chtěl, popřípadě není schopen tuto změnu definovat [4].

Jedním z řešení je použití adaptivního UI, které pro získávání informací o uživatelových potřebách využívá senzory telefonu, které mohou být zdrojem velice relevantních dat, protože svá mobilní zařízení zpravidla nosíme stále při sobě a sbíraná data mohou tedy velice přesně vypovídat o našem okolním kontextu.

Většina z těchto senzorů je v tomto ohledu již nějakým způsobem prozkoumána a dobře využita (GPS, eyetracking, síťový kontext, ...) [5], ale mikrofon a zvuk obecně je stále zdroj informací, kterému se nevěnuje moc pozornosti a který by také mohl přispět ke zlepšení interakce člověka se zařízením. Přitom sluch a řeč jsou dva ze základních lidských smyslů a člověk je používá ke komunikaci naprosto přirozeně a neustále. Stroje jen mají stále potíže se zpracováním zvuku a jeho analýzou, ale s rychlým vývojem, který dnes zažíváme, pro ně bude tento prvek stále důležitější a užitečnější.

Stroje ale nemusí jen poslouchat, mohou k nám také mluvit. S jejich vyjadřováním se potkáváme téměř každý den. Už jen když vezmeme v potaz všechna cinknutí a pípnutí ze svých telefonů, která nás upozorňují, že se něco děje, všechna zvuková upozornění z našich domácích spotřebičů a nebo když Siri odpovídá na položenou otázku. Dnes se značná část naší technologie vyjadřuje zvukem [8], ale tyto zvuky nemusejí sloužit jen k jednoduchému oznamování, ale mohou nás právě také navádět a ulehčit nám manipulaci s nimi.

1.2 Cíle

Cílem této práce je nastudovat techniky využití zvuku v mobilních aplikacích, možné problémy spojené s ovládáním mobilních aplikací a navrhnout postupy adaptace a vylepšení uživatelského rozhraní mobilního zařízení s využitím hlasového nebo šumového zvukového kontextu v okolí zařízení. Adaptace by měla uživateli zpříjemnit interakci s prostředím, dalším krokem je tedy vývoj jednoduché mobilní aplikace, na které se přívětivost adaptací otestuje.

Kapitola 2
Rešerše

2.1 Uživatelské rozhraní

Účelem uživatelského rozhraní je zprostředkovat veškerou komunikaci mezi člověkem a zařízením, které toto rozhraní poskytuje. Mělo by uživateli usnadňovat práci se zařízením a využívání jednotlivých funkcí aplikace, zapouzdřit funkcionalitu programu a skrýt ji za snadno ovladatelné komponenty.

Může nabývat mnoha forem, z nichž nejběžnější jsou hardwarové součásti zařízení jako je displej nebo reproduktor, grafické komponenty aplikace nebo zvuk aplikací vydávaný.

2.2 Adaptace rozhraní

Uživatelské rozhraní se může měnit několika různými způsoby za účelem přizpůsobení se konkrétnímu uživateli nebo okolí a tím zjednodušit svoji ovladatelnost. Změny se mohou lišit na základě akcí, jimiž byly vyvolány, typu vstupních dat, která pro provedení této změny byla použita nebo se mohou dělit podle délky trvání těchto změn, či doby, kdy byla změna navržena [1]. Různých definic a rozdělení adaptací je tudíž několik druhů. Pro účely této práce byl vybrán nejčastější typ rozdělení [2]:

- *Adaptabilní* - Adaptabilní systémy jsou ty, které uživateli umožňují si vybrat, nebo nastavit různé způsoby zobrazení a ovládání z těch, které jsou již zabudované v systému. Adaptabilní uživatelské rozhraní lze tedy definovat jako systém, ve kterém je změna iniciována akcí koncového uživatele prostřednictvím výběru z různých funkcí, které tento systém nabízí. Charakteristika změn je založena na informacích o uživateli, které byly známy už před momentem interakce se zařízením, a odhaduje se, že zůstanou neměnné po dobu této interakce.
- *Adaptivní* - Adaptivní systémy naopak mění prvky své struktury a funkcionality za účelem přizpůsobení se podle uživatelových potřeb, které se mohou i v průběhu času měnit. Tyto systémy jsou založeny na takovém principu, aby

byly schopné tyto změny a požadavky samostatně identifikovat a podle nich zvolit a provést příhodou adaptaci. Adaptivní uživatelské rozhraní jsou schopna upravovat svou strukturu na základě sledování stavu uživatele, stavu zařízení a prostředí, ve kterém se zrovna nachází, za běhu podle adaptační strategie.

Rozhodování, který z těchto přístupů je lepší a přijatelnější pro uživatele, není vůbec snadné. Samotné definice se mohou v mnoha případech překrývat. Adaptabilní systémy mají značnou výhodu v tom, že uživatel má plnou kontrolu nad tím, co se s aplikací děje a může si vše nastavit podle vlastních potřeb. Na druhou stranu, adaptivní systémy mohou pomoci v případech, kdy je rozhraní moc komplexní, nebo tehdy, když uživatel ani neví o možnostech zlepšení či usnadnění své komunikace se zařízením, protože nevyžadují žádný zásah z uživatelovy strany.

2.2.1 Přístupy k adaptaci

Kombinací těchto metod můžeme vytvořit několik variant, jak přistupovat k adaptování a zpříjemňování rozhraní uživateli:

- *Adaptabilní (Manuální)* - Uživatel ovládá proces a vykonává všechny úkony.
- *Adaptabilní s podporou systému (Uživatelský výběr)* - Uživatel ovládá prvky adaptace, které jsou zabudované v systému.
- *Adaptivní s ovládáním uživatele (Uživatelské schvalování)* - Systém provádí samotnou adaptaci pod uživatelovým dohledem. Systém provede určitou akci a upozorní uživatele na její možnosti, ze kterých může vybrat.
- *Adaptivní (Plně adaptivní)* - Celý proces adaptace je zaštiťován systémem, který rozhoduje o akci na základě nějakého modelu, kontextu a historii uživatelských rozhodnutí. V tomto případě je často kvůli nedostatku testovacích dat těžké vybrat optimální adaptaci.

Na Obrázku 2.1 máme shrnuty základní výhody a nevýhody každého z těchto přístupů.

2.2.2 Prvky adaptivního přístupu [3]

- *Složky adaptace* - prvky interakce (rozhraní), které budou adaptací změněny, např. zobrazené informace, interakční komponenty, metody či zařízení.
- *Spouštěče adaptace* - faktory rozhodující o provedení adaptace, např. charakteristika uživatele nebo kontext aplikace.
- *Cíle adaptace* - konkrétní veličiny, které chceme adaptací vylepšit nebo upravit, např. minimalizovat počet chyb, zlepšit ovladatelnost nebo efektivitu.

Types	*Advantages*	*Disadvantages*
Adaptable interface	Simpler Easy to learn Used in systems easy-to-use Can be organized by the user	Requires learning Don't available for complex systems
Adaptable with system support interface	Very efficient Support to the user Easy to adapt	Need Learning Can harm privacy
Fully adaptive interface	Not disturbing Predicting human behaviour Based on user's profile Reduces the user's cognitive load	Most Invasive Can harm privacy Unwanted information
Adaptive interface with user control	Provides means to enable and disable adaptation Control over privacy Adaptation in the form of a proposal Helped to remember	Disturbing/Distracting Can harm privacy Need Learning Confused notices

Obrázek 2.1. Výhody a nevýhody adaptačních přístupů [3]

- *Pravidla adaptace* - Definují postupy, podle kterých se budou adaptační složky měnit na základě adaptačních spouštěčů.

Spouštěči adaptace běžně bývají informace získávané za běhu aplikace z okolí zařízení. Souhrnu těchto informací se říká kontext aplikace nebo zařízení.

2.3 Kontext

Pojem kontextu obecně lze definovat mnoha různými způsoby v závislosti na tématu, k jakému se má definice vztahovat. Obecně lze za kontext považovat veškerá data a informace popisující zkoumané prostředí. V souvislosti s mobilními zařízeními lze kontext definovat jako [15]:

„Kontext je informace, která může být použita k charakterizaci situace, ve které se nachází osoba, místo nebo objekt, který je považován za relativní pro interakce mezi uživatelem a aplikací, včetně uživatele a aplikace samotné.“

Druhy prostředí, ze kterých pocházejí kontextové informace specifikující okolí by se daly rozdělit do těchto kategorií [16]:

- *Výpočetní prostředí* - dostupné procesory, dosažitelná zařízení pro uživatelský vstup a výstup, dostupnost sítě, rychlost sítě, výpočetní cena a podobné
- *Uživatelské prostředí* - umístění, pohyb, lidé v okolí a sociální kontext
- *Fyzické prostředí* - například teplota, osvětlení, řeč, hluk

Dnešní mobilní zařízení jsou vybavena mnoha různými senzory, které jsou schopné jednoduše zachytávat tyto informace a je tedy možné využívat je k úpravám prostředí aplikace podle aktuálních podmínek tak, aby byla pro uživatele v tomto momentu co nejpřívětivější a dostatečně jednoduchá na ovládání.

Jedním z nejbohatších zdrojů kontextových informací v mobilních zařízeních je zvukové okolí uživatele. Zvuk je kolem nás přítomný téměř neustále, používáme ho ke komunikaci mezi sebou navzájem, k detekování akcí probíhajících v našem okolí. Jednotlivé zvuky máme spojené s konkrétními událostmi, lidmi a věcmi a jsme schopni mezi nimi spolehlivě rozlišovat. Používáme ho dokonce i pro naše potěšení. A přece pro stroje jsou tyto schopnosti často ještě hodně nepřesné, málo použitelné nebo vůbec nemožné. Vývojem těchto dovedností u strojů, a zvláště mobilních zařízení, můžeme výrazně přispět k lepší uživatelské zkušenosti a k usnadnění procesů, které za nás mohou stroje vykonávat.

2.4 Audio kontext

Za zvukový kontext můžeme považovat veškeré zvukové okolí zařízení. Může to být kontext spojený s lidskou řečí, ať už to je hlas uživatele nebo jiných lidí, nebo charakteristické zvuky spojované s určitým významem či zdrojem nebo jen šum vycházející z obecného ruchu okolí. Zachycování těchto zvuků není s dnešní technologií mikrofonů komplikované, větší problémy přináší jejich charakterizování a klasifikace, aby z nich zařízení získalo potřebnou informaci a mohlo je nadále zpracovávat. Pro tyto účely se ze zvuku extrahuje nemálo hodnot, které daný zvuk specifikují a odlišují ho od ostatních.

2.5 Veličiny extrahované ze zvuku

Obecně mohou být tyto veličiny rozdělené do dvou kategorií: časové veličiny (time-domain) a spektrální veličiny (frequency-domain) [8].

Uvedu dvě z nejpoužívanějších časových veličin:

- *Krátkodobá energie (Short-time energy)* - představuje přibližnou reprezentaci změn amplitudy zvuku v čase
- *ZCR (Zero-crossing rate)* - Průchod nulou (Zero-crossing) nastává, když po sobě jdoucí záznamy mají opačné znaménko, tedy ZCR je průměrná hodnota počtu výskytů, kdy signál projde nulovou hodnotou, za krátký časový interval.

Podobně uvedu několik spektrálních veličin. Tyto veličiny jsou obvykle extrahovány za použití Rychlé Fourierovy transformace (FFT) na krátkých časových úsecích signálu a následně zpracovány podle požadované veličiny. Často používané veličiny mohou být tyto:

- *MFFCs (Mel-sequency cepstral coefficients)* - tyto koeficienty dohromady skládají mel-frequency cepstrum a jsou běžně využívané k rozpoznávání řeči a klasifikaci zvuků. Obyčejně se využívá třináct koeficientů.
- *Poměr pásmové energie (Band energy ratio)* - Poměr energie v konkrétním frekvenčním pásmu ku celkové energii.
- *Spektrální tok (Spectral flux)* - Používá se k měření rozdílu spektrální amplitudy mezi dvěma po sobě jdoucími rámci.
- *Statistické momenty (Statistical Moments)*
  - *Spektrální centroid (Spectral Centroid)* - měří jas zvuku. Čím vyšší centroid, tím vyšší jas.
  - *Pásmová šířka (Signal bandwidth)* - měří rozsah frekvencí signálu.
  - *Spektrální plochost (Spectral flatness)* - říká, jak moc je zvuk podobný tónu, nebo naopak je podobný šumu.
  - *Spektrální roll-off (Spectral roll-off)* - specifikuje frekvenci, při níž magnituda frekvenční odezvy dosáhne určitého procenta celkové magnitudy. Obyčejně se používá hranice 95%.
- *LPCCs (Linear prediction cepstral coefficients)* - Myšlenka lineární predikce spočívá v tom, že současný vzorek může být odhadnut jako lineární kombinace předchozích, což vytváří spolehlivější veličinu, která není tolik náchylná na náhlé změny.

Pro každý z jednotlivých aspektů zvukového kontextu existují specifická odvětví, která se zabývají jejich extrahováním ze zvukového signálu a jejich následnou analýzou. Tyto aspekty se dají rozdělit na prvky spojené s lidskou řečí a prvky obsažené v jiném typu zvukového kontextu jako jsou zvuky prostředí a šum.

2.6 Řeč

Řeč je v oblasti zvuku nejpřínosnějším zdrojem informací. Slouží jako nejpřirozenější prostředek komunikace mezi lidmi, a tedy by se mohla stát také ideálním způsobem komunikace mezi člověkem a strojem, proto je důležité naučit stroj řeč zachytit, rozpoznat jednotlivá slova a porozumět jejich významu a důsledkům, která vyvozují.

2.6.1 Speech enhancement

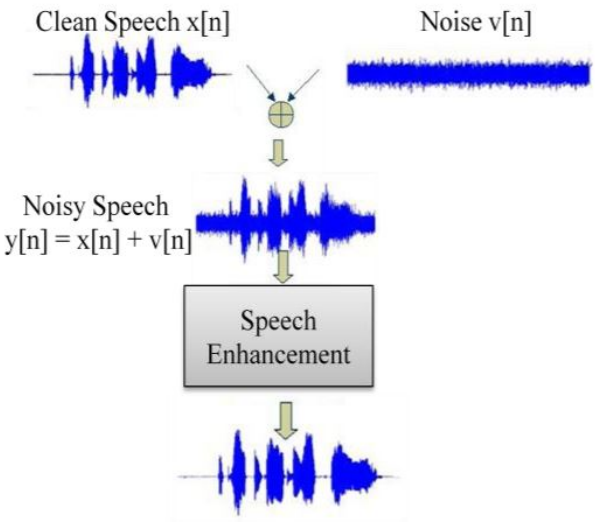
Techniky zvýraznění řeči vyvstávají v souvislosti se zpracováváním audio signálu řeči. Záznamy řeči jsou obvykle znečištěny hlasy jiných lidí, okolním ruchem nebo

ozvěnou. Tyto okolnosti značně snižují kvalitu nahrávek, srozumitelnost řeči a ztěžují proces automatického rozpoznávání řeči. Pro tyto účely je tedy užitečné umět tyto šumy v signálu efektivně odstraňovat. Obecná definice pro extrakci zvukového signálu by mohla znít takto [12] : *Obnovení signálu jednoho nebo více zvukových zdrojů ze zachyceného signálu obsahujícího jiné zvukové zdroje nebo ozvěnu.*

Proces zvýrazňování řeči má za úkol vylepšit zachycený signál zredukováním nebo plným odstraněním rušivého signálu. Tento hluk se dá rozdělit do tří základních skupin, pro které se používají rozdílné metody pro zvýrazňování řeči [13].

- *Periodický hluk* - Hluk, jehož signál se pravidelně periodicky opakuje. Efektivním přístupem pro jeho odstranění je použití různých druhů filtrace - stacionární, adaptivní nebo filtrace na základě přechodu domén (transform domain filters)
- *Širokopásmový šum* - Hluk, který je rozprostřen po velké části slyšitelného pásma. Techniky pro jeho redukci jsou nejčastěji spektrální odečítání (SS) a adaptivní rušení (adaptive cancellation).
- Jiná řeč - Pokud je v signálu obsaženo více mluvčích, pro jejich separaci je možné využít opět techniku přechodu domén nebo, pokud známe výšku jejich hlasu, lze využít Diskrétní Fourierovu Transformaci (DFT).

Je známo, že hluk se dá od signálu odečíst na základě skutečnosti, že charakteristiky hluku se s časem mění málo oproti charakteristikám hlasu. Na tomto faktu jsou založeny techniky zvýraznění řeči. Základní princip procesu zvýrazňování je vidět na Obrázku 2.2.



Obrázek 2.2. Proces zvýrazňování řeči [14]

Techniky mohou být rozděleny do skupin podle počtu vstupních signálů (tedy počtu mikrofonů nahrávajících signál) do single-channel a multi-channel větví [14].

- *Single-channel* - Vstupem je jediný mikrofon, kvůli čemuž jsou tyto metody výkonnostně omezené a často snižují srozumitelnost mluvené řeči. Jsou naopak ale cenově dostupnější. Nejvyužívanějšími přístupy jsou různé variace spektrálního odečítání.
  - *Spektrální odečítání* - Základní technika využívající aditivitu hluku.
  - *Spektrální odečítání s modelem odčítání (SSOM)* - Tato technika je vhodná pro odstraňování hudby v pozadí žádaného signálu.
  - *Nelineární spektrální odečítání (NSS)*
- *Multi-channel* - Vstupem je zdroj z více mikrofonů (obyčejně dvou), které jsou využívané pro získání pomocných hlukových signálů a jejich prostorových vlastností, na jejichž základě je následně zvýraznění prováděno.
- *Adaptivní potlačení hluku (ANC)* - Nejefektivnější a nejčastěji používaná metoda využívaná běžně v systémech sluchátek. Její funkcionalita spočívá v generování signálu shodného s rušivým signálem ale s opačnou fází, který vyruší efekt hluku. Rušivý signál se získává z pomocných vstupů, z nichž některé jsou blíže zdroji ruchu a v některých je ruch zachycen jen slabě.
- *Multisensor Beamforming* - Technika využívající geometrické rozložení mikrofonů a směr vstupu jednotlivých signálů, s pomocí kterých je odstraněn rušivý signál nekorespoundující fázi žádaného signálu.

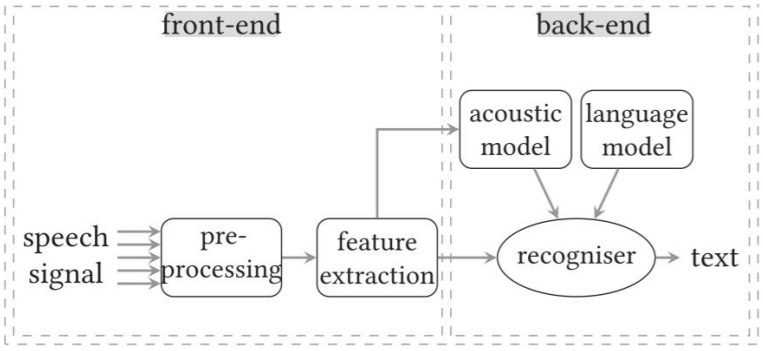
Zvýraznění řeči nachází největšího využití v procesu rozpoznávání řeči.

2.6.2 Speech recognition

Snad nejdůležitějším prvkem pro zachytávání zvukového kontextu je rozpoznávání řeči. Tímto nástrojem je možné rozpoznat a zaznamenat mluvené slovo v okolí zařízení, což má nespočet využití a poskytuje velice silný nástroj obzvláště v mobilních zařízeních. Tento proces můžeme definovat takto [11]:

"Rozpoznávání řeči (SRS) je proces automatického extrahování slov ze signálu řeči pomocí algoritmu. Je to schopnost stroje nebo programu identifikovat slova a fráze v mluvené řeči a převést je do strojem čitelného formátu."

Tento obor využívá širokou škálu technologií a zasahuje do několika jiných oborů. Je úzce spjat například s akustikou, fonetikou, teorií informací, lingvistikou, statistikou nebo strojovým učením a neuronovými sítěmi. Technik používaných k rozpoznávání je několik, každá má své výhody a nevýhody. Pokusíme se shrnout, co mají společného a pak si nastíníme funkcionalitu některých z nich.



Obrázek 2.3. Proces rozpoznávání řeči [11]

Základní struktura procesu je složená jen z několika bloků, jak můžeme vidět na Obrázku 2.3.

Hlas se nejprve nahraje prostřednictvím mikrofonu, který ho převede na elektrický signál. Systém zanalyzuje vstupní signál a vyextrahuje z něj potřebné veličiny (popsány výše) popisující danou stopu. Stejný proces byl proveden na mnoha záznamech lidské řeči, ze kterých byl vytvořen a naučen akustický model. Zpravidla jde o přiřazení jednotlivých fonémů jejich zvukovým stopám a jeho vytváření je založeno na statistických metodách analyzujících četnost výskytů fonémů v dané stopě. Pro spolehlivé rozpoznávání je také důležitý jazykový model, který naopak sleduje výskyty už celých slov a sousloví a zaznamenává jejich statistiky. Spojením těchto modelů vzniká výslovnostní slovník, který přiřazuje jednotlivým slovům jejich výslovnost (složením fonémů, ze kterých je slovo složeno).

Výpočetní síla je následně využita pro porovnání výslovnostního slovníku i obou modelů a charakteristiky nahrané stopy. Vyhledávací a porovnávací techniky se pak už mohou lišit podle velikosti modelů, optimálního rozsahu vstupní řeči, charakteristik jednotlivých jazyků a dalších parametrů.

- *Akusticko-fonetické rozpoznávání* - Proces je prováděn na úrovni fonémů. Každý z jednotlivých fonémů je definován seznamem akustických vlastností, které jsou viditelné v hlasovém signálu. Po separaci zvukových veličin se oznámkují fonémy pomocí segmentace signálu, z jejichž sekvencí jsou následně odhadnuta vyřčená slova.
- *Vzorové rozpoznávání* - Tento přístup formuluje vzory a pravidelnosti v řeči pomocí učících algoritmů na ohodnocených vzorcích. Rozpoznávaná řeč je potom porovnávána s nalezenými vzory.
- *Předlohové rozpoznávání* - V tomto přístupu se také využívají vzory nalezené v řeči, ale tentokrát k vytvoření slovníku, který slouží jako reference pro porovnání se vstupním hlasem. Pak se jen vybere nejlépe pasující předloha. Je to

nejstarší a nejméně efektivní metoda, použitelná pro malou škálu nejlépe od sebe oddělených slov. Dnes se používá spíše pro ověřování mluvčího.

- *Vektorová kvantizace (VQ)* - Při potřebě snížení objemu dat, potřebných k automatickému rozpoznávání je vhodné použít přístup vektorové kvantizace. Její efektivnost závisí především na výběru kódové knihy, která měří vzdálenosti mezi jednotlivými rámci rozdělených z řeči.
- *Dynamické borcení času (DTW)* - DTW je nástroj nacházející podobnosti mezi dvěma rozdílně rychlými signály, které jsou časově posunuté. S jeho využitím se rozpoznávací systém stává účinnějším pro sledy několika slov.
- *Statistická metoda* - V procesu rozpoznávání řeči se naráží na mnoho nepravidelností kvůli okolnímu šumu, různým mluvčím a dalším negativním vlivům. V tomto přístupu jsou slova vybírána na základě pravděpodobnosti vypočítané z charakteristiky vstupního signálu. Nejčastější technikou pro jeho charakterizování jsou Skryté Markovovy Modely (HMM). Jejich parametry reprezentují vlastnosti hlasového signálu, které se s časem mění. I v dnešní době je tato metoda považována za nejefektivnější přístup, jak dosáhnout rychlého a přesného rozpoznání řeči.
- *Umělé Neuronové sítě (ANN)* - V posledních letech jsou neuronové sítě v mnoha různých odvětvích. V rozpoznávání řeči dosahují skvělých výsledků v komplikovaných případech, ale nejsou ještě tak efektivní jako například HMM pro rozsáhlé jazykové modely. Neuronová síť simuluje lidský mozek a neuronové procesy v něm. Obsahuje velké množství uzlů (neuronů), které jsou poskládané do několika vrstev, mezi kterými je pomocí nelineárních funkcí přelévána váha na jednotlivé uzly. Počáteční váhy pro jednotlivé neurony jsou rozděleny při učení neuronové sítě na testovacích datech, což je velice dlouhý proces.

Textový záznam řeči se může využít rovnou například pro zadávání příkazů nebo diktování zpráv, ale lze ho také rozebírat po obsahové stránce a získávat z něj kontextové informace související se stavem uživatele, jako jsou například emoce, které mohou mít vysokou výpovědní hodnotu.

2.6.3 Sentiment analysis

Analýza sentimentu je rostoucí odvětví, které má za úkol automaticky určit sentiment obsažený v psaném textu. Sentiment samotný lze popsat jako pozitivní nebo negativní vyjádření pomocí jazyku [17]. Klasickým příkladem využití analýzy sentimentu je určování, jestli je komentář nebo hodnocení zveřejněné u zboží v internetovém obchodě pozitivní nebo negativní. Klasifikace sentimentu obvykle

vychází z informací získaných z pozitivních či negativních slov v textu, z jejich kontextu a slovní struktury textu. Proces analýzy lze rozdělit do tří kroků:

- rozhodování, jestli je text subjektivně zabarvený
- určování polarity zabarvení, tedy jestli vyjadřuje pozitivní nebo negativní názor
- určování míry tohoto zabarvení

Základem pro kvalitní klasifikaci sentimentu je dostatek již ohodnocených dat, na jejichž základě lze potom určit i ohodnocení nového textu. Rozhodujícím prvkem tohoto úkolu je informace, odkud data pochází. Podle něj lze rozlišit dva přístupy k analýze sentimentu: strojové učení a s pomocí lexikonu.

V případě strojového učení je vytvořen klasifikační systém, který je schopný určit polaritu nového textu. Tento systém je naučen na již ohodnocených zdrojích a textech, čemuž se v oboru strojového učení říká supervised learning, protože systém předem ví, jak jsou jednotlivé ukázky ohodnoceny. V tomto procesu se naučí rozpoznávat vlastnosti přiřazené jednotlivým polaritám (někdy je k možnostem klasifikace pozitivní a negativní přidána i varianta neutrální). Výhodou strojového učení je jeho vysoká přesnost a relativně rychlý proces učení. Nevýhodou naopak je závislost systému na trénovacích datech, takže systém nemůže být jednoduše použit na nový typ zdroje trénovacích dat.

Při využití lexikonu jsou hodnoty sentimentu odvozeny od polarity jednotlivých slov v textu, které jsou získány z již existujícího slovníku spojujícího každé slovo s jeho polaritou. Při analýze nového textu jsou všechna slova v něm ohodnocena pomocí tohoto slovníku a jejich hodnoty jsou potom zpracovány jedním z dostupných algoritmů, což ve výsledku vyjádří hodnotu sentimentu celého textu. Tento přístup je na rozdíl od strojového učení mnohem snazší využít v nových prostředích bez nutnosti vytváření nového slovníku.

Problém podobný analýze sentimentu je analýza emocí [19]. Jedná se o velice podobný klasifikační proces přiřazující slova nebo celé texty jedné ze sedmi základních lidských emocí [18]: hněv, strach, překvapení, znechucení, radost, smutek a opovržení. To může sloužit k přesnější identifikaci stavu uživatele, a tak lze zpřesnit řešení jeho využití.

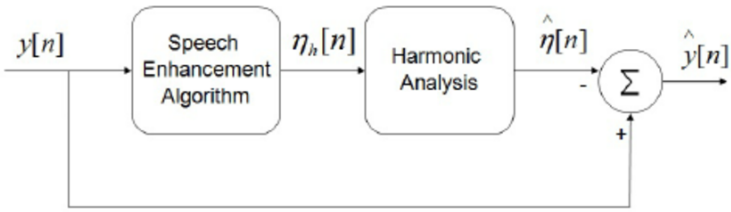
2.7 Ostatní zvuky

Protože je řeč nejtypičtějším zvukovým přenosem informací, věnuje se jí také nejvíce pozornosti. Ve zvukovém spektru to ale není zdaleka jediný prvek, který běžně slýcháme. Zvuky našeho okolí každým rokem ve městech zesilují a upozorňují nás

na události probíhající kolem nás. Jejich detekce a rozlišování je pro stroj užitečnou informací, kterou může využít pro lepší porozumění uživatelových potřeb.

2.7.1 Odstranění řeči

Proces zvýrazňování řeči a její separace ze zvukového signálu je běžným problémem, který je dnes úspěšně řešen složitými metodami, jejichž výsledky jsou vynikající a spolehlivé. Už méně častým požadavkem je odstranění řeči ze zvukového signálu za účelem využití jiného zvukového kontextu, než je řeč. Metodiky zvýrazňování řeči sice často fungují na základě separace řeči od zbytkového zvuku, ale většinou v něm zanechávají zbytkový signál řeči, který je pro rozpoznávání a analýzu zvuků prostředí rušivý. V práci [20] byl navržen systém pro jeho odstranění přidáním analýzy harmonie za algoritmus zvýraznění řeči, jak je vidět v blokovém schématu Obrázku 2.4.



Obrázek 2.4. Princip odstranění řeči [20]

2.7.2 Environmental sound recognition

V oblasti rozpoznávání zvuku je kromě řeči a hudby důležitou součástí i rozpoznávání zvuků okolí (ESR), které je již poslední částí skládanky k tomu, aby stroje byly schopné vnímat celou škálu zvuku jako člověk. Okolními zvuky se myslí jak přírodní, tak člověkem produkované zvuky, které potkáváme v běžném životě, tedy veškeré zvuky kolem nás kromě lidské řeči a hudby. Této oblasti je věnována pozornost teprve v průběhu posledního desetiletí, kdy je běžně dostupná vysoká výpočetní síla a ESR je možné provádět s relativně vysokou přesností [23]. Využití schopnosti zachytit a rozeznat jednotlivé zvuky v okolí a případně je přiřadit i k události s nimi spojené je hned několik. Může pomoci hluchým nebo špatně slyšícím lidem reagovat na události nebo nebezpečí probíhající kolem nich [21], detekovat porouchanost stroje ze zvuku, který vydává při práci, v chytrých domech nebo dokonce pro rozpoznávání zvířecích druhů nebo typu automobilů [22].

Použití technik z rozpoznávání řeči nebo zvuku bývá v případě ESR hodně nepřesné, protože jejich signál je obecně stacionární (jeho vlastnosti se v čase nemění) a harmonický. Oproti tomu jsou jakékoliv jiné zvuky obvykle nepravidelné,

nestacionární a nedají se, jako například řeč, rozdělit na malé základní jednotky. Množství zvuků obecně je také mnohem větší a jsou navzájem více rozdílné.

Uvedu zde základní tři techniky pro jejich zpracování [23]. Všechny z nich na svém konci využívají klasifikační model (běžně realizovaný pomocí neuronových sítí) pro získání nejlepší shody s konkrétním zvukem.

- *Framing-based* - Signál je rozdělen na malá okna, ze kterých jsou extrahovány zvukové veličiny, které jsou použity pro trénování sítě i následné testování. Jednotlivá okna jsou potom přiřazena k patřičným zvukům. Problémem tohoto přístupu je zvolení vhodné velikosti okna.
- *Sub-framing-based* - Namísto oken je signál rozdělen na menší části, ze kterých jsou extrahovány veličiny a jejich složením je teprve provedena klasifikace patřičného okna.
- *Sequential* - Rozdělení je provedeno do ještě menších segmentů (20-30ms), které se navzájem z poloviny překrývají, díky čemuž je tato metoda schopná zachytit dlouhodobé rozdíly v signálu a vztahy mezi segmenty.

Každá z metod využívá pro klasifikaci zvuků veličiny extrahované ze signálu, kde každá veličina vyjadřuje jinou vlastnost zvuku a je vhodná pro klasifikaci jiných typů zvuků. Proto je výběr veličin použitých v procesu ESR velice důležitou součástí vytváření takového systému.

2.8 Přehrávání zvuku

Usnadnění a zpříjemnění uživateli práci se zařízením je možné dosáhnout i s využitím přehrávání zvuků vyjadřujících upozornění, nápovědu nebo zpětnou vazbu aplikace. Uživatelské rozhraní může být zejména na mobilních zařízeních prostorově značně omezené, takže namísto přeplňování obrazovek nepřehlednými vizuálními komponentami, poskytování informací uživateli prostřednictvím zvuku se nabízí jako další přirozená alternativa, která bývá často uživateli jasnější a snáze zapamatovatelná.

2.8.1 Text-to-speech synthesis [9]

Syntéza hlasu (TTS) je obor zabývající se simulací syntetického hlasu podle textového zdroje, který tento hlas předčítá a slouží jako přirozenější rozhraní v komunikaci člověka se strojem. Z pohledu uživatele je nezbytné, aby byl syntetizovaný zvuk podobný skutečnému lidskému hlasu. V posledních letech je tento obor mnohem více zkoumán, díky čemuž se ve většině projektů tohoto faktu povedlo uspokojivě docílit.

Proces syntézy je většinou složen ze 3 částí [10]:

- *Normalizace (předzpracování)* - v této části se provádí rozpoznávání kontextu a významu slov, aby se předešlo špatné výslovnosti těch, které mohou být vysloveny více způsoby. Tím mohou být například číslovky, data nebo homografy (slova, která se stejně píší, ale jinak vyslovují). Dosahuje se toho statisticky s využitím Skrytých Markovových Modelů (HMM) nebo Hlubokých Neuronových sítí (DNN).
- *Převod slov do fonémů* - slova se mohou rozdělit do grafémů (typicky jednotlivá písmena nebo slabiky) a pak převést na fonémy podle určitých pravidel s nimi spojených. Stroj je pak schopen tento proces provést pro jakákoliv slova, i když je nezná, ale problémy pak způsobují veškeré výjimky ve výslovnosti jednotlivých jazyků.
- *Převod fonémů na zvuk* - k této poslední části se dá přistupovat třemi různými způsoby:
  - *Spojováním* - Získané fonémy se přiřadí k nahrávkám fonémů získaných ze záznamů mluvících lidí, které byly rozděleny. Výhodu tohoto přístupu je vysoká podobnost výsledné řeči s lidskou, ale bývá omezen na jeden lidský hlas a jazyk.
  - *Formanty* - Tento přístup nechává stroj zvuk samotné řeči syntetizovat generováním frekvencí o různých hlasitostech. Díky tomuto mohou jednoduše přečíst jakákoliv slova z libovolných jazyků a libovolným hlasem. Avšak tento hlas stále zní značně roboticky.
  - *Artikulací* - K dosažení řeči se použije aparát podobný lidskému hlasovému ústrojí, aby se dosáhlo co největší podobnosti. Toto pole je stále v raném rozvoji a nemá mnoho reálných použití.

Jelikož se kvalita syntézy za posledních několik let značně zvýšila zejména v ohledu srozumitelnosti a přirozenosti, je tato technologie již běžnou součástí mnoha systémů v různých odvětvích. Příkladem mohou být navigace, hlasoví asistenti, roboti, dokonce většina operačních systémů v mobilech má vestavěnou funkci text-to-speech.

Technologie syntézy řeči v této práci není použita, protože přímočaré metody, jak uživateli usnadnit komunikaci se zařízením, jsou v mobilech již implementovány (Voice Assistant) a hlavní zaměření práce bylo směřováno do jiných oblastí zvuku.

2.8.2 Auditory icons and earcons

Kromě řeči lze také z mobilních zařízení přehrávat neverbální zvuky charakterizující specifickou činnost, stav nebo událost v aplikaci a poskytovat tak informaci uživateli méně rušivým a více intuitivním způsobem. V uživatelském rozhraní lze tyto zvuky využít pro sdílení potvrzení nebo zamítnutí provedeného kroku uživateli, například při vyplňování formuláře, nebo pro informování, kde se v aplikaci nachází, například při procházení menu aplikace.

Neverbální zvuky lze rozdělit do dvou kategorií [24]: audio ikony, které reprezentují skutečné události z našeho světa, a earcony, které jsou abstraktní synteticky vytvořené nebo hudební zvuky. Z hlediska intuitivnosti a jasnější poskytnutí zamýšlené informace jsou audio ikony vhodnější a u uživatelů dosahují rychlejších reakcí. Na druhou stranu oproti earconům se nedají použít v každé situaci, protože k většině akcí a událostí v oblasti mobilních aplikací se nedají přiřadit reálné zvuky tak, aby vysvětlovaly právě probíhanou aktivitu.

Earcony mohou být různě skládány a upravovány za účelem vyjádření rozdílu informace jimi nesené. Například je možné měnit jejich barvu, rozsah, rytmus, výšku nebo intenzitu a tak vyjádřit hierarchii mezi nimi [25].

Kapitola 3
Analýza a související práce

3.1 Možnosti využití zvuku na platformě android

V rámci vývoje mobilní aplikace může být využití technologií spojených se zvukem někdy značně obtížné. Obtíží může být například nedostatečný výkon pro náročné procesy používané ve většině těchto technologií. Dnešní chytré telefony sice disponují již překvapivou výkonnou silou, která udržuje mobilní telefon v plynulém chodu, ale ve srovnání s webovými servery, na kterých jsou zpravidla tyto procesy spouštěny, zdaleka nedosahují dostatečných výsledků. Některé serverové služby jsou sice veřejně přístupné, ale většina si za své služby účtuje poplatky.

3.1.1 Environmental sound recognition

Rozpoznávání zvuků okolí je poměrně nové odvětví a na chytrých telefonech není ještě nějak běžně využíváno. Jeho využití v mobilních zařízeních je přitom velice rozsáhlé. Lze ho uplatnit například v rozeznávání blížících se aut k jdoucímu uživateli [26] nebo identifikování zvuku dronů pohybujících se v okolí zařízení [27], což může být v posledních letech aktuální problematika. V případě adaptování uživatelského rozhraní se znalost okolního zvukového kontextu nabízí ke stanovování určitých scén a prostředí, ve kterých se uživatel právě nachází, na jejichž základě by se přizpůsobovala aplikace uživatelovým současným potřebám spojeným právě s tímto prostředím.

- Příkladem může být detekce prostředí hromadné dopravy, ve kterém by se uživateli mohly zobrazovat nabízené navazující spoje, současná poloha nebo často zde využívané komponenty rozhraní.
- Jiným případem může být rozlišování, zda uživatel sedí za volantem. Spousta dopravních nehod bývá způsobena využíváním mobilního telefonu v průběhu řízení a řidiče není snadné tento zvyk odnaučit. Aby se zkrátil čas uživatele věnovaný telefonu, aplikace by se při řízení přepnula do zjednodušeného režimu, který by nabízel jen základní funkcionalitu aplikace.
- Míra hluku kolem nás roste každým rokem a s tím souvisí i ztráta schopnosti soustředění [28]. V hlučných prostředích by se aplikace mohla přizpůsobovat

ztrátě pozornosti uživatele a udržovat ho v právě prováděném úkonu zvýrazňováním komponent. Využívání sluchátek při transportu je běžnou záležitostí většiny lidí, což snižuje jejich schopnost reagovat na případná okolní nebezpečí a události. Aplikace by mohla tyto události zachytávat a přehráváním upozornění uživateli do sluchátek ho o nich uvědomovat.

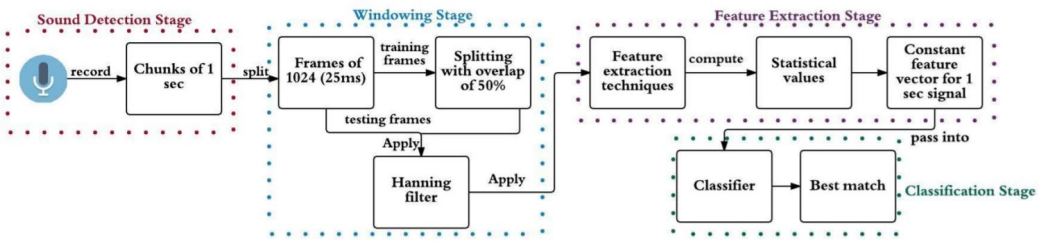
- Aplikace by mohla napomáhat při detekci aktuálního počasí rozeznáváním zvuků deště, větru, krup nebo hromu a sbírat tak data pro přesnější upozorňování uživatele na počasí v jeho přímém okolí.

Existují určitá řešení rozpoznávání zvuků okolí dosahujících použitelných výsledků, nejsou ale veřejně přístupné nebo nejsou vhodné pro implementaci do mobilního operačního systému.

TarsosDSP [31] je framework pro Android operační systém, který je schopný spolehlivě detekovat základní zvuky v okolí jako je tlesknutí, hvízdnutí nebo jiný výrazný signál. Pro rozpoznávání kontextuálních zvuků však není určený a v širším měřítku zvuků podává špatné výsledky.

EARS [29] je implementace systému pro real-time zpracování okolního zvuku a jeho klasifikaci s využitím konvolučních neuronových sítí pro SoC (System on Chip) systémy bez potřeby serverové části pro učení sítě. Implementován je v jazyce Python a pro jeho specifikaci na SoC prostředí nelze snadno využít pro mobilní aplikace.

Jiné práce [21, 26, 30] se zabývají problémem implementace takového systému do operačního systému Android a popisují postupy použité pro jeho vytvoření, ale nezveřejňují svůj zdrojový kód, a tedy není způsob, jak je snadno pro adaptaci přepoužít. Pro zreplikování takového systému by bylo nutné naimplementovat jednotlivé kroky klasifikace zvlášť: nahrávání zvuku, rozdělení nahrávek do malých částí, extrahovat zvukové veličiny z každé části do vektoru a tento vektor zanalyzovat klasifikačním systémem. Kroky jsou názorně vidět na Obrázku 3.1.



Obrázek 3.1. Proces klasifikace zvuků [21]

Z důvodu komplexnosti tohoto procesu, nedostatku veřejně přístupných řešení a nejistotě, jestli by byla výsledná přesnost použitelná pro adaptační proces, jsem se rozhodl zaměřit se na řečovou oblast audio kontextu.

3.1.2 Speech recognition

Rozpoznávání řeči je celkově již zaběhlejším nástrojem. V oblasti chytrých telefonů je často již implementováno a denně používáno. Navzdory tomu není zprovoznění nepřetržitého rozpoznávání, aby mohl být zaznamenán veškerý řečový kontext v okolí zařízení v průběhu jeho používání, v mobilu úplně snadné. Pokud se chceme držet neplacených řešení, existuje pár cest, kterými se můžeme vydat:

- *Android SpeechRecognizer* - Funkcionalita nativně vestavěná v operačním systému Android. Tato třída je určena pro diktování textových zpráv nebo zadávání hlasových povelů, proto je rozpoznávání omezeno na několikasekundové mluvení a po krátké chvíli bez zaznamenání řeči se vypíná. Pravděpodobně z důvodu vysokých nároků na spotřebu baterie nelze přepnout do režimu nepřetržitého rozpoznávání. Výhodou této třídy je vysoká přesnost rozpoznávání, které je dosaženo především díky přeposílání mluveného signálu na server, kde je řeč zpracována a výsledky jsou posílány zpět do zařízení.
- *PocketSphinx* [32] - Open source nástroj pro zpracovávání hlasového signálu zejména pro rozpoznávání řeči. Tento nástroj umožňuje kupříkladu zachytávání jen několika konkrétních slov definovaných jednoduchou gramatikou, rozpoznávání vyřčení specifické fráze nebo kompletní nepřetržité rozpoznávání řeči. Rozpoznávací systém je možné si nastavit podle vlastních potřeb a lze použít na Android zařízeních v offline módu bez nutnosti streamování dat na server, což značně snižuje náročnost aplikace na spotřebu baterie. Přizpůsobit vlastním potřebám lze nastavením vlastního jazykového a akustického modelu pro přesnější charakterizaci poslouchaného jazyka. Vytváření těchto modelů vyžaduje však velké množství dat s nahrávkami požadované řeči. PocketSphinx v sobě zahrnuje základní modely pro několik řečí dokonce upravené pro používání na nízko-výkonnostních zařízeních, ale pro dosažení alespoň srovnatelných výsledků se serverovým rozpoznáváním vyžadují zásah do parametrů algoritmu a znalosti v oblasti SRS.
- *Sphinx4* [33] - Nástroj z projektu CMUSphinx, pod kterým je veden i nástroj PocketSphinx. Sphinx4 je hlasové rozpoznávání psané v Javě, tudíž využitelné na platformě Android, které navíc umožňuje i funkcionalitu rozpoznávání mluvčího. Oproti PocketSphinx je více modifikovatelné, poskytuje adaptaci modelů a jiné funkcionality nepřímo spojované s rozpoznáváním řeči. Bohužel je také

náročnější na výkon, a tedy primárně určené pro desktopové nebo serverové využití, ale pro mobilní zařízení není ideálním nástrojem.

V oblasti placených nebo do určité míry nehrazených nástrojů je výběr podstatně širší a výsledky rozpoznávání přesnější. Pro účely nastínění všeobecného přehledu dostupných STT (speech-to-text) nástrojů uvedu několik z nich včetně základních informací:

- *Microsoft Azure* [34] - Komplexní machine-learning framework od firmy Microsoft poskytující nesčetné nástroje včetně převodu řeči na text, který je považován ze jeden z nejlepších v současné době. Zdarma poskytuje 5 hodin rozpoznávání za měsíc.
- *Google API* [35] - Nástroj od firmy Google, který nabízí rozpoznávání až 120 různých jazyků s možností automatického rozpoznání a nahrávání dlouhých zvukových záznamů. Neplacené je při prvních 60 minutách nahrávek.
- *IBM Watson* [36] - Firma IBM poskytuje také soubor nástrojů se strojovým učením, ve kterém nabízí převod textu na řeč, užitečného především při zpracování velkého množství dat. Placený od první minuty.

3.1.3 Sentiment analysis

Diktování textových zpráv nebo ovládání základních funkcionalit hlasem je klasickým případem využití schopnosti řeč rozpoznat a zaznamenat, ale řeč se dá také využít k mnohem sofistikovanějším účelům. Z vlastností signálu hlasu je možné detekovat různé vady řeči jedince [39] nebo napomáhat při zlepšování výslovnosti cizího jazyka [40]. Dnes se již dokonce projevují snahy naučit počítač zpracovávat rozpoznaný text a pochopit jeho význam. Pokud by mohl chytrý telefon z řeči jeho uživatele zachytit nejenom příkazy, které má v sobě zakódované, ale také pochopit její smysl a význam, mohl by mnohem lépe porozumět současným potřebám uživatele a podle nich se přizpůsobit a reagovat na ně.

Naučit počítač chápat význam slov může sice ještě několik let trvat, ale emocionální stav uživatele je získatelná informace, která hodně vypovídá o uživatelově rozpoložení a jeho potřebách. Mobilní aplikace může využívat znalost toho, co uživatel právě cítí, ke své adaptaci:

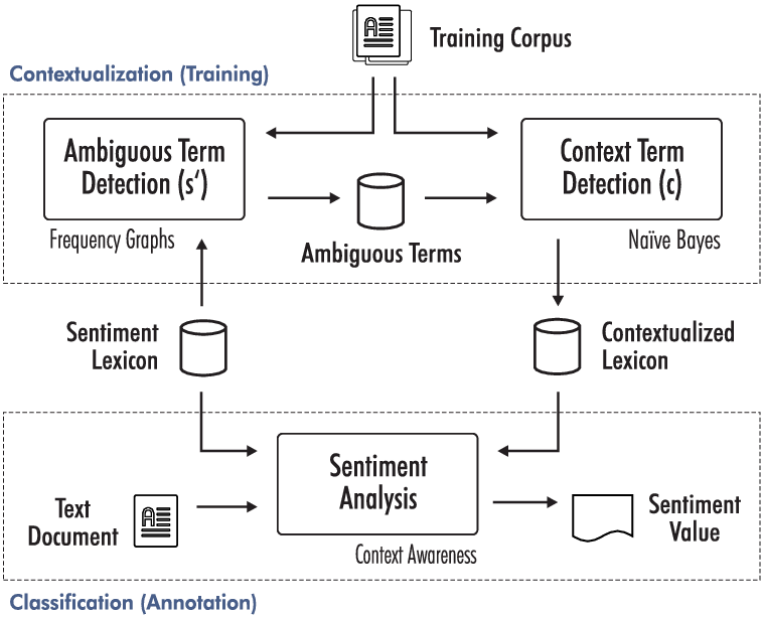
- Aplikace se může snažit uživateli zmírnit dopad negativních nálad zpříjemněním svého prostředí, například změnou barev, které mají prokazatelný vliv na lidskou náladu.

- Rozeznání momentů, kdy je uživatel ve stresu, například v nějakém stresovém prostředí, může aplikace využít pro snížení složitosti rozhraní a upřednostnění pravděpodobně hledaných komponent a tím odstranit možnou příčinu stresu.
- Větší výkyv nálady bývá na internetu někdy spojen s hodnocením například nějakého produktu, filmu nebo práce. Tyto emoce by mohla aplikace vložit do automaticky vytvářených šablon recenzí a tím uživateli usnadnit jejich psaní.
- Negativní emoce a projevy mohou být spojeny se špatným návrhem či rozložením uživatelského rozhraní. Uživatel může být s aplikací nespokojený někdy až do té míry, že bude svoje emoce slovně vyjadřovat, což může aplikace použít jako zdroj informací pro analýzu toho, v jakém bodě je rozhraní uživateli nepříjemné a snažit se ho podle toho náležitě adaptovat.

Existují práce zabývající se analýzou aktuálních emocí [38] nebo míry stresu [37] uživatele přímo z veličin zvukového signálu. Takovýto přístup je úkolem pro strojové učení a do budoucna může dosahovat velice přesných výsledků. Jelikož jsme ale schopni hlasový signál převádět na text, je přímočařejším a jednodušším přístupem analyzovat emocionální informace přímo z něj podle statistik a jimi vytvořených slovníků.

Pro základní implementaci analýzy sentimentu nebo přiřazování emocí k textu je dostačující vzít jeden z veřejně dostupných lexikonů, které přiřazují slova jednotlivým skupinám, a rozpoznaná uživatelova slova ohodnocovat na těchto lexikonech. Slova se mohou buď jednoduše přiřazovat jejich vahám, které se sečtou pro každé slovní vyjádření, z čehož dostaneme již celkem přesnou představu o sentimentu samotného textu. Nebo lze analyzovat i celá slovní spojení a celkový kontext slov mezi sebou, což vede k přesnějšímu rozeznávání neobvyklých slovních obratů jako je například ironie, rozeznávání dvojznačnosti některých slov, detekování používaných slovních spojení a dalších výjimečných případů.

Obrázek 3.2 shrnuje proces vytváření a aplikace kontextuálně vnímavých lexikonů, které jsou příkladem využití kontextových informací. Systém nejprve detekuje dvojznačné výrazy v trénovacích datech, pak najde kontextové výrazy pro každý z nich, a nakonec zahrne kontext do procesu analýzy sentimentu.



Obrázek 3.2. Využití kontextuálních lexikonů pro analýzu sentimentu [47]

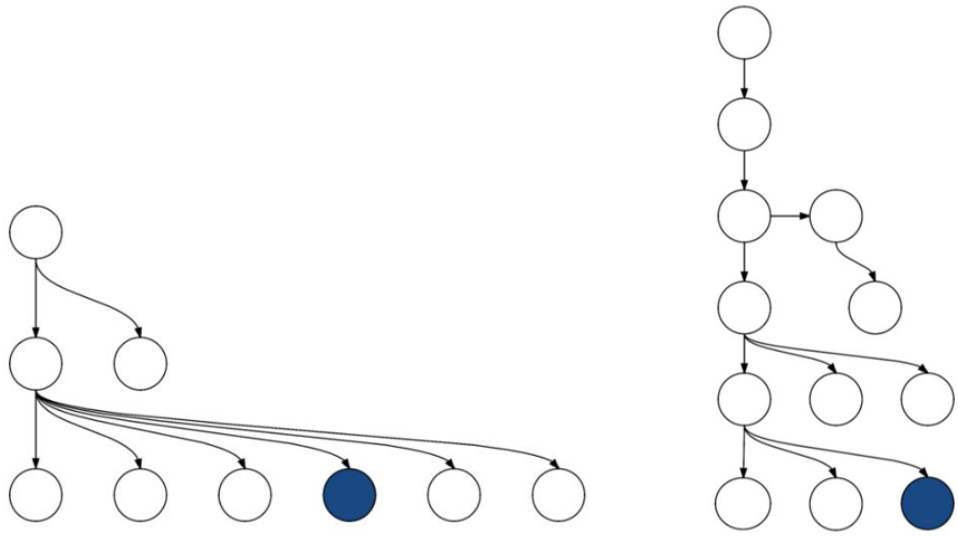
3.2 Zvolené přístupy

Pro účely této práce byl zvolen přístup k adaptaci na základě analýzy nespokojenosti uživatele s nabízeným rozhraním, která probíhá za běhu mobilní aplikace. Cílem je jednak zjistit, jestli lze tímto způsobem nespokojenost detekovat, tedy zda existují případy, kdy uživatel vyjádří své negativní emoce nahlas při ovládání aplikace, a také otestovat, jestli následná navržená adaptace obecně zmírňuje míru nespokojenosti.

Častým problémem při návrhu aplikace bývá hloubka, šířka a celkové rozložení libovolných menu, která slouží k pohybu mezi funkcionalitami aplikace. Tato menu mohou být uživatelsky nepřívětivá, protože se v nich uživatel ztrácí, nemůže nalézt hledané prvky kvůli jejich špatnému rozložení pod rodiči, velkému počtu prvků pod jedním rodičem (Obrázek 3.3 vlevo), přílišné hloubce stromu prvků menu (Obrázek 3.3 vpravo) nebo obecně velkému počtu prvků v menu.

Problémy s tímto spojené jsou často špatně detekovatelné, protože uživatel prvek běžně nakonec najde, jen mu to zabere víc času, a to není žádný zásadní problém omezující funkcionalitu aplikace. Způsoby, jakými by menu mělo přaskládávat svou strukturu pro zlepšení použitelnosti, jsou již analyzovány a testovány v jiných pracích [44, 45] a v této práci tomu nebude přikládán velký důraz, ačkoliv bude navržen postup využití získaných řečových kontextových informací pro jeho úpravu.

Pro analýzu sentimentu ze zachyceného hlasu by bylo postačující využití sentimentového lexiconu a ohodnocování výrazů podle jednotlivých slov, protože



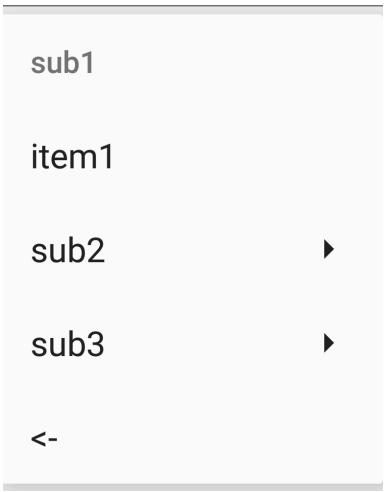
Obrázek 3.3. Problémy souvisejícími s menu strukturou: obsáhlá úroveň (vlevo), hluboký strom (vpravo) [45]

očekávaná slovní vyjadřování uživatele při průchodem menu nebudou pravděpodobně moc obsáhlá, takže detekování případných ironií nebo jiných pro analýzu rušivých řečových prvků bude obtížné. V oblasti analýzy sentimentu existují ale frameworky potýkající se alespoň se základními z nich a jejich použití by nemělo podávat horší výsledky než základní přístup. Framework VADER-Sentiment-Analysis [42] nabízí jednoduché použití a je schopný správně analyzovat i slangové výrazy, zápory ve větách a přidává určitou logiku k celkové skladbě vět. Jeho využití bude vhodné pro aplikace na mobilním zařízení, jelikož není výpočetně náročný a pro náš případ krátkých mluvených výrazů podává uspokojující výsledky. Přístup klasifikátorů a strojového učení je pro mobilní zařízení výpočetně příliš náročný a kontextově vnímavé lexikony, jelikož jejich vytváření je vcelku nová praxe, jsou ještě obecně málo přístupné. Pro rozlišování, zda je uživatel spokojený nebo nespokojený, bude analýza sentimentu více než dostačující.

Avšak pro porovnání s přístupem klasifikace do více tříd než pozitivní a negativní a pro zjištění, jestli by se daly takovéto informace získávat, bude využito rozdělení výroků mezi základní lidské emoce. NRC [43] je lexikon přiřazující slova k 8 základním emocím vytvořený ohodnocováním skupinou lidí. S jeho využitím můžeme alespoň přibližně určit emocionální hodnotu mluvené řeči, ale výpovědní hodnota nebude příliš vysoká a bude sloužit především pro účel odhadu emoce nejčastěji vyjadřované při nespokojenosti s rozhraním.

Nepřehlednosti příliš hlubokého menu může být náročné se zbavit, pokud jsou prvky v něm důležité a jejich počet se nedá redukovat. Seskupením prvků pod

méně rodičů by se menu nezpřehlednilo, protože by byly úrovně příliš obsáhlé a název rodiče příliš obecný pro snadné nalezení žádaného prvku. Proto se můžeme pokusit alespoň pomoci uživateli zorientovat se ve struktuře stromu. V případě hlasově prezentovaných menu, která nejsou viditelná (například telefonní volby), byl navržen přístup přehrávání zvukových zpětných vazeb, earconů, k napovídání uživateli, v jaké úrovni menu se právě nachází [41]. Ke specifikaci zvukem, o jakou úroveň se jedná, byly využity vlastnosti zvuku jako jsou například výška nebo barva. Uživatel s jejich použitím prokázal lepší orientaci v menu a rychlejší reakce při výběru položek. V případě vizuálně přístupných menu může být situace podobná. Menu na obrazovkách jsou sice většinou dobře viditelná, takže uživatel by neměl mít potřebu nějaké nápovědy, co na obrazovce právě vidí. Jenže zobrazovány jsou vždy jen prvky jedné úrovně, jak je vidět na Obrázku 3.4, a pokud je strom menu dostatečně hluboký a z názvu prvků to není zřejmé, uživatel často ztratí přehled, jak hluboko se ve stromě nachází. V tom případě nemusí vědět, jaké úkony má provést pro to, aby se přiblížil k hledanému prvku.



Obrázek 3.4. Android menu

3.3 Menu na platformě android

Vybrat vhodný typ menu pro implementaci adaptivní struktury může být problematické. Pro správné otestování adaptivních metod je důležité, aby se struktura menu mohla měnit v průběhu jeho procházení a aby bylo možné libovolně procházet veškeré úrovně stromu včetně vracení se zpět do vyšších úrovní. Na platformě android existují tři základní typy menu:

- *Options menu* - Menu v pravém horním rohu aplikace určené pro možnosti spojené s navigací v aplikaci. Je spojené s Activity třídou, která zastupuje určité

aktivity v aplikaci a moc často se nemění, takže načítání options menu probíhá jen zřídka.

- *Context menu* - Menu přiřazované jednotlivým stránkám aplikace, které slouží pro výčet možností s úpravou elementů na této stránce. Je spojené s View třídou, která je častým prvkem obrazovky a její načítání je běžné. Otevírá se dlouhým podržením elementu na stránce.
- *Pop-up menu* - Menu, které může být specifické pro každý jeden prvek v aplikaci a slouží pro upřesnění akce spojené s tímto elementem.

Bohužel ani jedno z těchto menu nepodporuje funkci vracení se o úroveň zpět ve stromu menu a pokud ji chce programátor využívat, musí si menu trochu přizpůsobit a reagovat na události jím vyvolané, aby podobnou funkcionalitu nasimuloval. Co se týče modifikování menu za běhu aplikace, context nebo pop-up menu jsou přívětivější vzhledem k tomu, že jejich načítání je prováděno častěji a změny by byly hned viditelné. U option menu se ale dá docílit stejného efektu uchováním si odkazu na něj a vzhledem k tomu, že toto menu je lépe přístupné a vyvolává více událostí, ukázalo se pro tuto simulaci jako nejvhodnější. Konkrétněji jsou tyto postupy a problémy popsány v sekci Implementace.

Kapitola 4
Návrh

Tato kapitola obsáhne počáteční proces vytvoření testovací mobilní aplikace pro tuto práci, čímž je návrh. Návrh systému představuje rozvržení a promyšlení jeho funkcionality ještě před samotnou implementací. Správným návrhem se lze vyvarovat mnoha problémů, které na začátku nemusí být viditelné, a může výrazně zkrátit dobu implementace.

4.1 Architektura

Aplikace bude rozdělena do několika základních částí podle funkcionality. Tím se docílí modulárního rozvržení, které zpřehledňuje návrh systému a usnadňuje jeho rozšiřování.

Způsob, jak jsou základní funkce od sebe rozděleny a jakým mezi sebou jednotlivé části komunikují, určuje architektura aplikace. Architektonické vzory, kterých se může mobilní aplikace držet, bývají v běžných případech tyto tři:

- *Model-View-Controller (MVC)* - Základní vzor, kterým lze od sebe vrstvy aplikace oddělit. Model zastupuje datovou vrstvu a komunikaci s ní skrze ukládání a načítání dat. View je vrstva zobrazující rozhraní uživateli včetně potřebných dat z Modelu. Controller reaguje na veškeré akce uživatele a obstarává logiku celé aplikace, jejíž důsledky promítá do dat Modelu. Nevýhodou tohoto vzoru bývá, že Controller je realizován aktivitou nebo fragmentem, které často nabydou velkých rozměrů a ztrácí svou přehlednost.
- *Model-View-Presenter (MVP)* - Tento vzor se od MVC tolik neliší. Hlavní rozdíl tkví ve zrušení vazby mezi View a Modelem. O jejich propojení se stará Presenter, který je při akci notifikován od View a rozhoduje, která data z Modelu budou ve View zobrazena. Stejný problém nafukování třídy jako u MVC aktivity může nastat i u Presenteru, pokud si vývojář nedá pozor na jeho správné rozdělování.
- *Model-View-ViewModel (MVVM)* - Rozdělení vrstev je v tomto vzoru opět podobné. View upozorňuje ViewModel o akcích uživatele jako v případě Presenteru. U MVVM ale ViewModel nemá přímou vazbu na View, pouze poskytuje

seznam daných událostí, na které může reagovat i více View najednou. Komunikace a zobrazení dat zde probíhá přes obousměrné vazby mezi UI komponentami a jejich obrazy v kódu. Tímto přístupem se docílí větší nezávislosti jednotlivých vrstev a usnadní jejich testování. Na druhou stranu ale vyžaduje komplexnější XML definice a více počáteční práce se zavedením architektury.

Jelikož tato aplikace slouží jen pro testování několika funkcionalit, nebude nijak zvlášť obsáhlá a nepředpokládá se, že by byla v budoucnu hodně rozšiřována. Vzor MVVM projeví své výhody zejména při větších aplikacích, které poskytují rozsáhlé rozhraní. V naší aplikaci bude rozhraní obsahovat jenom několik komponent a vhodnější proto bude výběr jiného vzoru. Avšak logika skrytá v adaptačním procesu nebude triviální a u MVC vzoru by její implementace vedla k nafouknutí hlavní aktivity. Proto byla vybrána architektura MVP, která zajistí dostatečnou modularitu systému a nevyžaduje zbytečně moc implementačního úsilí.

4.2 Rozdělení komponent

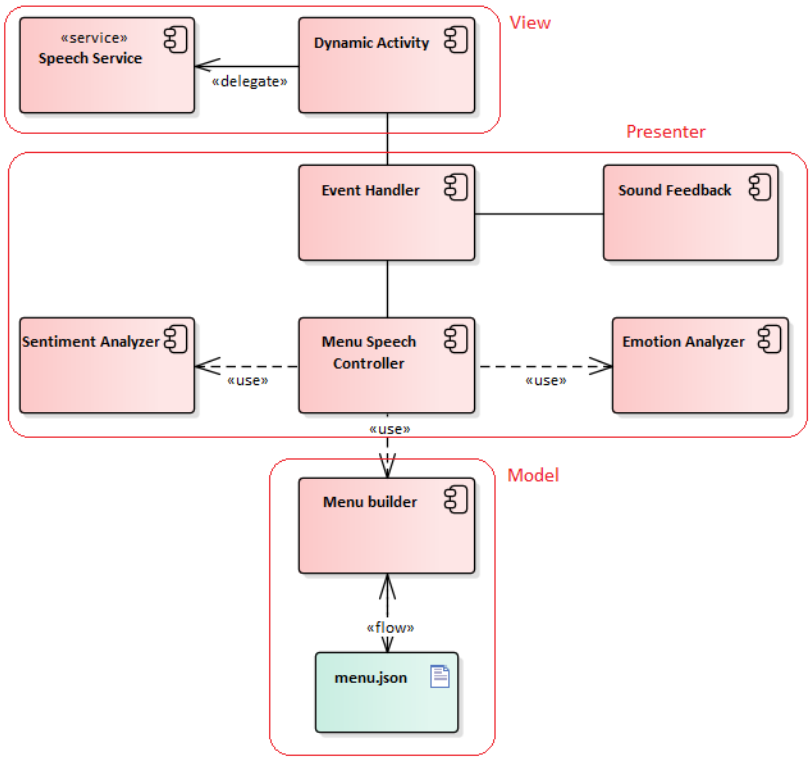
Samotný vzor popisuje základní myšlenku přístupu k rozdělení vrstev. Ta ale musí být pro správný návrh ještě podrobněji rozvedena rozdělením do specifických komponent. Pro každou funkcionalitu jednu.

Základní komponentou mobilní aplikace bývá již zmíněná aktivita, která je její nezbytnou složkou a slouží k přijímání uživatelových příkazů a zobrazování změn uživateli. Aktivita má možnost spouštět a zastavovat službu k rozpoznávání řeči, která po celý čas rozpoznávání běží na pozadí aplikace a nenarušuje její chod, dokud nezachytí výskyt řeči. Společně by se daly jako třídy poskytující uživatelské rozhraní zařadit pod View.

Po jejím zachycení se řeč převede na text a výsledek přepošle do třídy EventHandler, která zodpovídá za zpracovávání veškerých událostí vyvolaných v aktivitě. Reaguje konkrétně na události spojené s akcí uživatele ovládajícího menu a na detekci mluvených slov. Díky těmto událostem je pak schopný nasimulovat chybějící funkcionalitu menu potřebnou pro naše testování. Při průchodu menu uživatelem provolává EventHandler třídu SoundFeedback, která se stará o přehrávání zvukové zpětné vazby v podobě earconů.

Záznam rozpoznané řeči je zpracováván ve třídě MenuSpeechController. Ta zaštiťuje algoritmus adaptace menu. K tomu využívá třídy SentimentAnalyzer a EmotionAnalyzer, které poskytují rozhraní k příslušným službám ohodnocení textu. Tyto třídy implementují logiku celé aplikace a rozhodují o změnách a zobrazovaných datech v ní, společně proto tvoří návrhovou část Presenter.

Při rozhodnutí o potřebné adaptaci menu je tento úkol přenechán třídě MenuBuilder, která prvky menu podle pokynů přeskládá a výsledné menu propaguje uživateli. MenuBuilder mimo to slouží i pro sestavení menu při spuštění, kdy je struktura menu nahrána ze souboru ve formátu JSON, takže lze menu při testování snadno zaměňovat přidáváním nových souborů a výsledek adaptace pak zase persistovat zpět do souboru. MenuBuilder zařizuje nahrávání a poskytnutí dat pro View a spolu se soubory definujícími menu tvoří část zvanou Model.



Obrázek 4.1. Rozdělení komponent aplikace

Na schématu 4.1 je znázorněn diagram jednotlivých komponent aplikace. Služba rozpoznávání řeči je ovládána přímo z hlavní aktivity aplikace, protože služby běžící na pozadí musejí být s nějakou aktivitou spojeny a jejich funkčnost končí s ukončením aktivity. Tato služba může být snadno nahrazena implementací jiné rozpoznávací služby, pokud výsledek rozpoznávání vrací jako String slov převedených na text přes broadcast zprávy. Přecházení mezi službami při testování různých metod pro převod řeči na text znamená tedy pro vývojáře změnu jen jednoho řádku kódu.

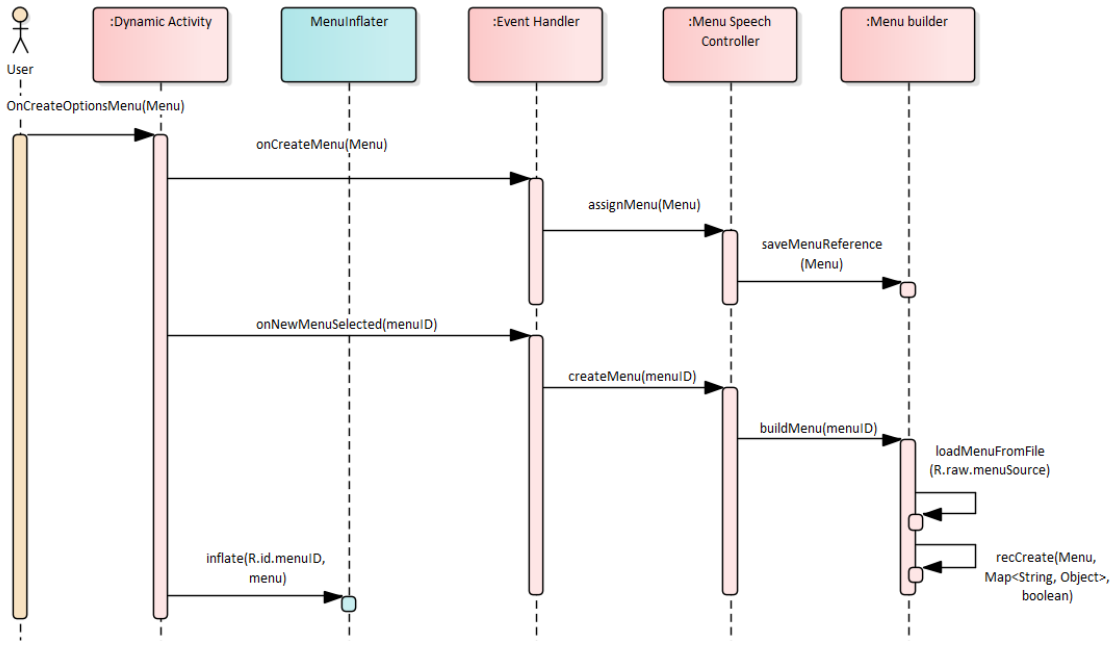
Různé služby na vyhodnocování sentimentu a emocí z textu naopak poskytují výsledky často v různých formách a přechod mezi nimi tedy vyžaduje větší zásah do algoritmu implementovaného ve třídě MenuSpeechController.

Načítání menu ze souboru a jeho zpětné zapisování je užitečné při dlouhodobé adaptaci, protože při jakékoliv změně v jeho struktuře je nutné tuto změnu uchovat do příštího spuštění aplikace, kdy bude znovu načteno. Přináší to s sebou i výhodu při testování, kdy vývojář nemusí požadované menu náročně vytvářet v kódu, ale může ho zapsat přímo do souboru v intuitivním JSON formátu a změnit zdroj pro načítání.

4.3 Procesy programu

4.3.1 Vytváření menu

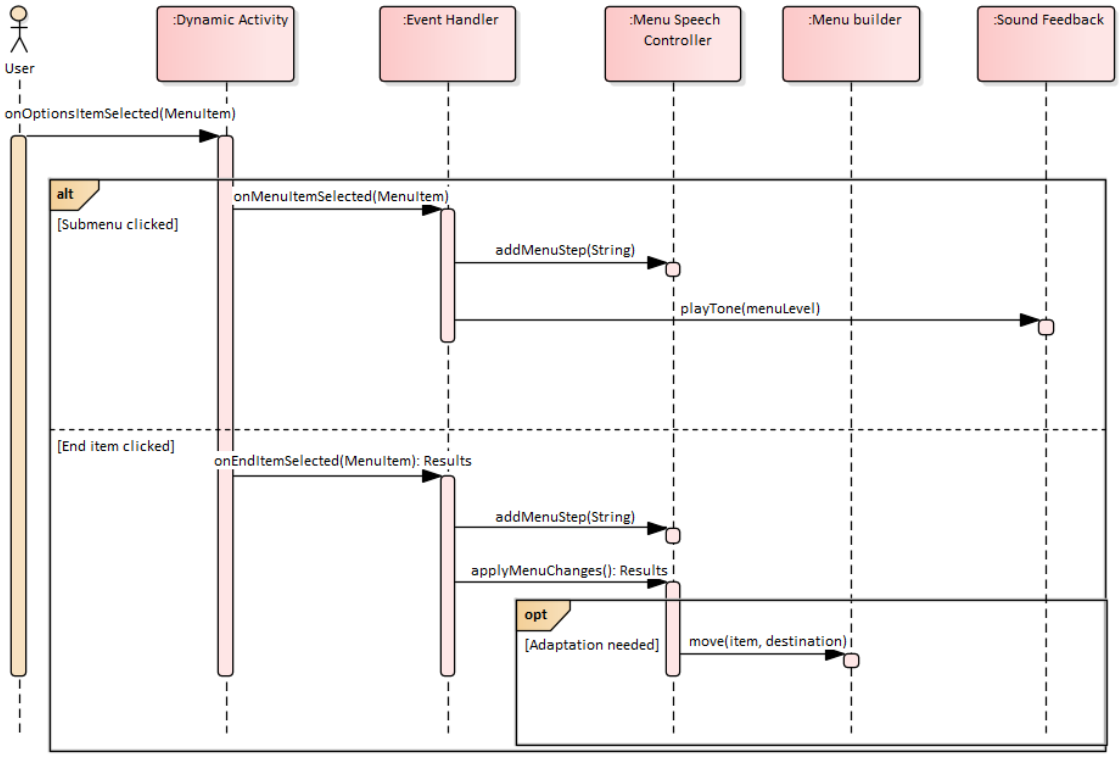
Menu je vytvářeno jen při spuštění aplikace provoláním metody onCreateOptionsMenu(), která je volaná automaticky při existenci tohoto menu v aplikaci. Díky tomu je množství procesů čtení souboru z paměti omezeno na minimum a při běhu aplikace je menu uchováno v rychlé cache paměti. Zdrojový soubor pro menu lze snadno měnit v hlavní aktivitě a pro účel testování lze měnit i přímo z aplikace za běhu výběrem v drop-down menu. Celá struktura menu je obsažená v tomto souboru, ale volání inflate() na instanci MenuInflater přijímá jako argument odkaz na XML soubor, ve kterém se na platformě Android menu běžně definují, ale pro naše použití je tento zápis moc obsáhlý a pro ruční vytváření struktury menu příliš komplikovaný. Avšak pro takto již definovaná menu je stačí vložit jako argument metody inflate() a jejich prvky se ve výsledném menu zobrazí.



Obrázek 4.2. Sekvence kroků při vytváření menu

4.3.2 Výběr tlačítka

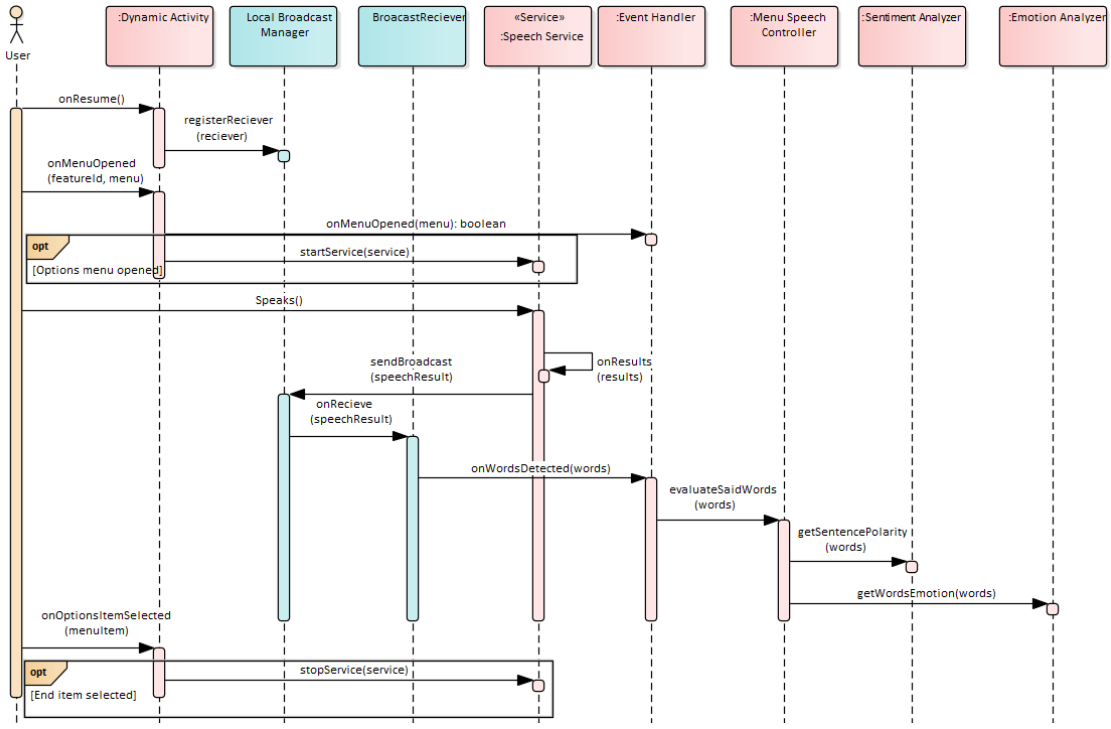
Při každém stisku na libovolný prvek menu je informace propagována do MenuSpeechController, aby bylo zachyceno, kde se uživatel právě nachází, a aby se mohl postup průchodu menu případně použít pro algoritmus adaptace. EventHandler také přepošle současnou hloubku menu do SoundFeedback, aby se přehrál zvuk příslušný této úrovni. Při výběru konečného prvku menu a tedy jeho zavření se provede test na potřebu adaptace a ta se případně v menu realizuje přesouváním prvků pomocí metody move() v MenuBuilder. Rozlišování, o jaký typ prvku se jedná, probíhá na úrovni aktivity podle jejich parametrů, protože Android mezi nimi nerozlišuje a vyvolává stejnou událost pro všechny.



Obrázek 4.3. Sekvence kroků při výběru tlačítka

4.3.3 Rozpoznávání řeči

Proces rozpoznávání řeči se spouští, jakmile uživatel otevře menu, a ukončuje se hned při výběru koncového prvku. Tím je tato služba používána jen tehdy, když je potřeba, což má vliv na úsporu baterie, protože výpočty spojené s rozpoznáváním mohou být velice náročné. Aby mohlo rozpoznávání běžet na pozadí a umožňovat ovládání aplikace v jeho průběhu, je spouštěno jako služba Androidu, a tudíž musí být veškerá komunikace s aplikací prováděna skrze broadcast zprávy. Textový



Obrázek 4.4. Proces rozpoznávání řeči

formát řeči je potom předán frameworkům k analýze, jejíž výsledky jsou ukládány po celou dobu, kdy je menu otevřené.

Kapitola 5
Implementace

V této kapitole jsou nastíněny části implementace testovací aplikace, které jsou nějakým způsobem zajímavé z hlediska vývojáře nebo způsobovaly v průběhu vývoje komplikace. Je určena především pro čtenáře se základními znalostmi programování a výše zmíněných technologií.

Pro účely implementace bylo využito vývojové prostředí Android Studio od firmy JetBrains a z důvodu zaměření aplikace na operační systém Android byl použit programovací jazyk Java.

5.1 Nepřetržité rozpoznávání řeči

Jak již bylo zmíněno v analýze, umožnit rozpoznávání řeči po celou dobu průchodu menu uživatelem není na mobilním zařízení jednoduché. Nejpřístupnějším řešením se zdál být framework PocketSphinx, který by prováděl rozpoznávání po neomezenou dobu přímo na zařízení. Toto řešení bylo zprovozněno a kvůli omezenému výpočetnímu výkonu telefonu byl použit značně prořezaný jazykový model. I přes snahu optimalizovat modely a přizpůsobit je prostředí nebyly výsledky rozpoznávání uspokojivé a nebylo možné je využít. Služba však byla v aplikaci ponechána pro umožnění dalších vylepšení. Z tohoto důvodu byla implementována další služba, která pro rozpoznávání využívá v systému Android vestavěnou třídu SpeechRecognizer, která provádí výpočty na serveru a výsledky jsou tudíž mnohem přesnější. Tato třída je určena primárně pro zadávání hlasových příkazů a diktování krátkých zpráv, proto se po chvíli ticha vždy nahrávání hlasu zastaví. Tento fakt lze obejít detekcí tohoto zastavení a spouštěním služby stále dokola. Třída ovšem nevyvolává žádnou událost při svém ukončení, proto je nutné měřit čas, po jaký je stále ještě aktivní, a po jeho uplynutí službu zavolat znovu.

```
CountDownTimer noSpeechCountDown = new CountDownTimer(5000,1000){
        @Override
        public void onFinish()
        {
            isCountDownOn = false;
            restartRecognizer();
        }
```

Ukázka kódu 1: Odpočítávání doby rozpoznávání

Spouštění časovače je prováděno pokaždé, jakmile je služba zavolána a začne vyčkávat na řeč, a ukončován je v případě, že vypršel čas, vyskytla se chyba, nebo byl zachycen záznam řeči.

```
public void onReadyForSpeech(Bundle params) {
        isCountDownOn = true;
        noSpeechCountDown.start();
        Log.d(MainActivity.TAG, "onReadyForSpeech");
}
```

Ukázka kódu 2: Spuštění časovače

Dalším problémem je zachytávání okamžiku, kdy je celé menu otevřeno nebo zavřeno, aby mohlo být rozpoznávání podle toho spouštěno a ukončováno. Existují sice metody onMenuOpened() a onPanelClosed(), které mají sloužit tomuto účelu, jsou ale bohužel volány pro libovolné menu komponenty v aktivitě, což zahrnuje i jednotlivá submenu. Není ani snadné z nich vyvodit nějakou souvislost, protože počet jejich volání není jednoznačný.

Android však rozlišuje mezi hlavním menu a jeho submenu a proto lze rozeznat otevření hlavního menu pomocí funkce getClass() zavolané na prvku menu.

```
if (menu != null && menu.getClass().toString().contains(".MenuBuilder"))
```

Ukázka kódu 3: Rozeznání otevření menu

Při splnění této podmínky došlo jistě k otevření hlavního menu a služba může být spuštěna. Detekce úplného zavření menu je ještě obtížnější, protože metody k tomu určené jsou staré a v dnešní verzi systému nejsou provolávány, přičemž předchozí přístup použitý při otevírání nelze zopakovat, protože hlavní menu je zavřeno i v případě přechodu z první úrovně menu do druhé, kdy je otevřeno příslušné submenu.

Jediným způsobem je reagování na stisk koncového prvku menu, při kterém se menu zavře. To nemusí být úplně spolehlivé, protože menu se může zavřít i kliknutím mimo strukturu menu. Jelikož je aplikace vytvářena pouze na testování, lze tuto nedokonalost tolerovat.

5.2 Vytváření a úprava menu

Menu jsou v systému Android reprezentována běžně XML soubory definujícími strukturu menu. Proces jejich upravování není zcela běžný a nejjednodušším přístupem se ukázalo být načítání struktury z JSON formátu.

```
{"submenu1":{
    "submenu2":{
      "item1":""
    },
    "submenu3":{
      "item2":"",
      "item3":""
    }
  }
}
```

Obrázek 5.1. Příklad zápisu menu struktury ve formátu JSON

Ten může být načten s pomocí knihovny Gson do Map struktury, se kterou lze snadno pracovat a pozměňovat ji. Vytváření menu pak probíhá rekurzivním procházením této struktury s využitím nativních metod menu pro přidání prvku nebo submenu.

```
void recCreate(Menu menu, Map<String, Object> map, boolean backItem){
        for (String title: map.keySet()) {
            if (map.get(title) instanceof String) {
                menu.add(0, title.hashCode(), 0, title);
            }
            else recCreate(
                    menu.addSubMenu(0, title.hashCode(), 0, title),
                    (Map<String, Object>) map.get(title),
                    true);
        }
        if (backItem) menu.add("<-");
}
```

Ukázka kódu 4: Vytváření menu

Podobně jsou implementovány i metody pro přesun jednotlivých prvků napříč strukturou. Díky použité rekurzi může aplikace pracovat s libovolně hlubokými nebo širokými menu.

5.3 Vracení se v menu

Pro hlubší menu je nutné, aby se v nich uživatel mohl libovolně pohybovat a vracet. Tato funkcionalita není na Androidu podporována u základních menu, protože nejsou obecně určena pro velké množství prvků. Avšak při našem testování musejí být použita dostatečně hluboká menu, aby se docílilo přesnějších testovacích výsledků, jak je přesněji popsáno v sekci Testování. Proto bylo naimplementováno řešení, které tuto možnost uživateli poskytuje.

Do každého submenu je přidán prvek navíc, který zastupuje funkci zpětného tlačítka, jak je možné vidět v Ukázce 5. Když uživatel menu prochází, jednotlivé kroky jsou ukládány do Stack struktury, aby prvky byly přístupné v pořadí od naposledy stisknutého do prvního. Při stisknutí zpětného prvku se vezme předposlední prvek, který reprezentuje o úroveň vyšší submenu, a provede se akce s ním spojená, což v případě submenu znamená jeho otevření.

```
public void onBackItemSelected() {
   menuLevel -= 2;
   menuStepsOrder.pop();
   if (menuStepsOrder.empty()) {
       returningToMenuTop = true;
       menuLevel += 1;
   }
  else menu.performIdentifierAction(menuStepsOrder.pop().getItemID(),0);
}
```

Ukázka kódu 5: Vracení se o úroveň výš

V případě vracení se do nejvyšší úrovně menu je nutné místo toho zavolat metodu openOptionsMenu() z hlavní aktivity, protože první úroveň menu nemá žádného rodiče.

5.4 Adaptace

Samotná informace o spokojenosti uživatele získaná z řeči není dostačující pro velké změny ve struktuře menu a měla by být kombinována s jinými kontextovými druhy informací. Proto byly navrženy dva přímočaré způsoby adaptace testující hlavně použitelnost této informace, ale samostatně mohou být pro uživatele příliš drastické.

První z nich provádí analýzu sentimentu od otevření menu a ukončuje ji při jeho zavření. Pokud byla v tomto intervalu zaznamenána řeč přesahující svou negativní hodnotou určitou hranici, je finální prvek, jehož zvolením bylo menu uzavřeno, přesunut o jednu úroveň výše. To má za úkol rozeznat prvky, které jsou špatně zařazené pod svými rodiči, a přesunutím ve stromu nahoru je dostat do širšího tematického okruhu, kam by už mohly patřit.

Druhý způsob pracuje také jen po dobu otevření menu a během něho přiřazuje hodnotu sentimentu jednotlivým rodičům v menu. Jestliže se při průchodu uživatelem na nějakém z nich dostal negativní sentiment přes danou hranici, zvolený koncový prvek se přesune do submenu definovaného tímto rodičem. Tento přístup také detekuje špatně zasazené prvky ve stromu, ale jejich přesunutí se snaží přizpůsobit očekáváním uživatele. Předpokládá se totiž, že uživatel bude projevovat

svou zlost z toho, že prvek nemůže najít, právě tam, kde hledaný prvek očekával. Adaptace tak může libovolně přeskládávat prvky napříč celým menu. Hranice, při jejímž překročení k adaptaci dojde, může být snadno změněna nastavením jedné hodnoty v kódu. Pro účel testování bylo zvolena hodnota -0.3 jako střední cesta mezi příliš častou adaptací a potřebou testování adaptace na několika jedincích. Za negativně zabarvené texty se považují texty s hodnotou nižší než -0.05 a v našem případě chceme adaptovat jen, pokud je uživatel rozčilen nebo hodně nespokojen. V případě reálného použití by byla vhodnější vyšší hranice, jinak by mohla příliš častá změna vést k ještě větší nespokojenosti z toho, že prvky nejsou na místech, na kterých si je uživatel pamatuje.

Způsoby adaptace lze v aplikaci snadno přidávat definicí nové enumerační hodnoty Mode ve třídě MenuSpeechController a implementací jeho logiky pod příslušnou větev switch rozdělovače v metodách analyzujících řeč a provádějících adaptaci v této třídě.

```
public enum Mode{
        MOVING_UP(1),
        MOVING_TO_SOURCE(2);
```

Ukázka kódu 6: Definování metod adaptace

5.5 Přehrávání earconů

Důležitým krokem je zvolení správných zvuků pro roli earconů v aplikaci, protože v sobě musí obsahovat informaci, která bude zvuk spojovat se současnou událostí, a zároveň by neměl být pro uživatele rušivým prvkem při ovládání aplikace.

V případě vyjadřování hloubky menu s jejich pomocí vyvstává problém, že předem není dána samotná maximální hloubka menu, a tudíž neznáme počet potřebných earconů pro pokrytí všech možností. Zároveň musí být earcony navzájem podobné, aby byl rozdíl v jedné z jejich vlastností co nejzřejmější.

Proto byl zvolen přístup generování zvuku za běhu tak, že současná hloubka bude promítnuta do frekvence (a tedy výšky) výsledného tónu. V tom uživatel snadno pozná rozdíl mezi tóny navzájem a je možné generovat tón pro libovolně hluboké menu, pokud zůstane v mezích slyšitelnosti lidského ucha.

Kapitola 6
Testování

Pro získání informací o použitelnosti navržených funkcionalit v systému mobilní aplikace bylo provedeno testování na uživatelích, kteří provedli zadané pokyny a odpověděli na otázky hodnotící celý proces. Podle výzkumu [46] stačí na testování použitelnosti systému 5 uživatelů a výsledky přibližně odpovídají skutečnosti. Avšak z důvodu nízkého počtu případů uskutečnění adaptace a pro získání větší jistoty ve výsledcích se zúčastnilo testů 8 lidí.

6.1 Test rozpoznávání sentimentu a adaptace

Pro otestování použitelnosti využití sentimentu řeči k detekci špatně navržených menu a jejich následné adaptace bylo vytvořeno hodně obsáhlé a až 5 úrovní hluboké menu, které v sobě obsahovalo špatně zařazený prvek. Cílem uživatele v testovacím procesu bylo nalézt a zvolit právě tento prvek. Uživatel strukturu menu před testem neznal a musel proto prvek v rozsáhlém menu dlouho hledat, což mělo za cíl umocnit jeho nespokojenost a vztek, které by mohl během hledání slovně vyjádřit. Po dobu hledání byl uživatel v místnosti sám, aby nebyla míra verbálního vyjadřování omezena kvůli přítomnosti jiné osoby, a ze stejného důvodu ani neznal cíl tohoto testování.

Poté zodpověděl uživatel otázky zkoumající postupně kvalitu rozložení testovaného menu, pocity a potřebu je vyjádřit během hledání a přívětivost adaptačních scénářů. V případě, že k adaptaci nedošlo například z důvodu, že uživatel nepromluvil, byl alespoň jeden ze scénářů nasimulován manuální úpravou menu, jako kdyby uživatel překročil hranici adaptace, aby ho mohl ohodnotit. U druhého to bohužel nebylo možné, protože adaptaci ovlivňuje čas uživatelova vyjadřování a je pokaždé jiná. Uživatelé odpovídali na otázky vyjádřením míry souhlasu na škále od 1 do 5, kde 1 znamená “vůbec ne” a 5 “rozhodně ano”. První adaptační scénář byl přesun prvku o jednu úroveň výše, druhý scénář byl přesun prvku ke zdroji vyjádření nespokojenosti.

Otázky:

- Byl hledaný prvek špatně umístěn?
- Cítil/a jste zlost při hledání prvku?
- Měl/a jste chuť vyjádřit pocity slovně při průchodu menu?
- Byl prvek po adaptaci podle prvního scénáře ve struktuře menu lépe zařazen?

Pokud došlo k adaptaci při testu:

- Byl prvek po adaptaci podle druhého scénáře ve struktuře menu lépe zařazen?

Z důvodu funkčnosti analýzy sentimentu a emocí pouze pro anglický jazyk byli pro testování vybíráni pouze lidé mluvící anglicky a nemluvící česky, protože u nich byla větší pravděpodobnost, že při vyjadřování použijí angličtinu.

6.1.1 Výsledky a odpovědi

	Počet (z 8)	v %
Promluvilo při hledání	4	50
Dosáhlo hranice adaptace	1	12.5

Tabulka 6.1. Hlasové reakce uživatelů

O1	O2	O3	O4	O5
4	2	5	5	4
3	3	4	3	-
4	2	1	5	-
4	2	4	4	-
5	3	3	3	-
5	4	5	2	-
2	1	1	3	-
5	3	3	5	-
4	2.5	3.25	3.75	4

Tabulka 6.2. Tabulka odpovědí 1

Odpovědi na otázku 1 potvrdily, že hledaný prvek byl v menu špatně zařazen a testování proběhlo tedy na správné situaci. V tabulce 6.1 vidíme, že více než polovina uživatelů promluvila v souvislosti s procesem hledání prvku. Když k tomu zvážíme průměr odpovědí otázky číslo 3, který ukazuje přiklánění uživatelů spíše

k vyjádření se než mlčení, můžeme říct, že při nespokojenosti je slovní vyjádření uživatele běžnou záležitostí.

Z emocionálního hlediska nedošlo k přiřazení vyřčených slov k žádné emoci ani v případě, kdy byl sentiment vyhodnocen jako negativní. Uživatelé sami uvedli v odpovědích na otázku 2, že zlost nebo hněv nebyla emoce, která by byla spojena s nespokojeností, jak by se dalo očekávat.

K adaptování menu došlo pouze v jednom případě. To je zapříčiněno tím, že uživatelé, kteří své pocity projevili, mluvili příliš potichu na to, aby slova mohla být zachycena, nebo vyjádřené výrazy nebyly negativního charakteru. První adaptační scénář musel být proto simulován a druhý byl ohodnocen jen jednou. Pro uživatele byly oba případy adaptace posunem k lepšímu rozložení menu, ačkoliv u druhého není výsledek příliš věrohodný.

6.2 Test zvukové zpětné vazby

Pro testování užitečnosti přehrávání earconů definujících současnou hloubku menu, kde se uživatel nachází, byla vytvořena dvě menu, která jsou si podobná strukturou, ale obsahují různé prvky. Obě jsou obsáhlá a hluboká až 7 úrovní. Úkolem uživatele bylo nalézt v každém menu postupně dva prvky, přičemž u jednoho průchodu byla zapnutá funkce přehrávání earconů. V obou případech byl měřen čas, za jaký uživatel oba prvky našel, aby mohla být porovnána míra zrychlení při zvukové nápovědě a jestli takto poskytnutá informace vůbec uživateli pomohla. Funkce byla zapínána u každého testu s opačným menu než u předchozího subjektu, aby se zmírnil dopad toho, že jsou menu rozdílná a rozdíl v čase průchodu může být ovlivněn rozvržením každého z nich. Nebylo možné testovat oba případy na stejném menu, protože by uživatel při druhém průchodu už znal jeho strukturu a byla by tak ovlivněna rychlost jeho rozhodování. Před testem byla všem vysvětlena funkce a význam tónů, aby jejich špatné pochopení nevedlo k zavádějícím nápovědám.

Poté uživatel opět odpověděl na otázky, které zkoumaly přehlednost testovaných menu, užitečnost earconů při průchodu a naopak jejich rušivost při něm. Opět volil míru souhlasu na škále od 1 do 5, kde 1 znamená “vůbec ne” a 5 “rozhodně ano”.

Otázky:

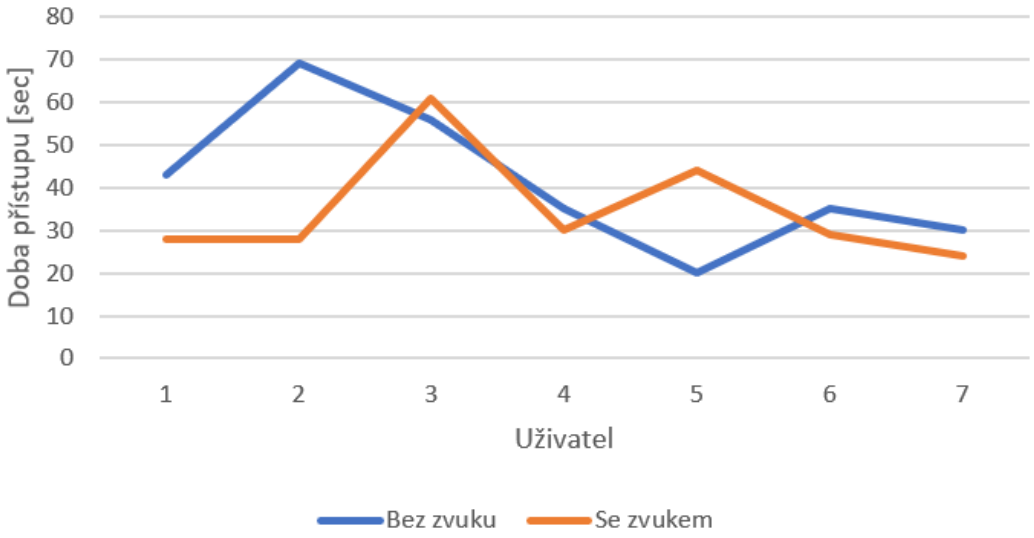
- Bylo menu nepřehledné?
- Pomohla Vám zvuková zpětná vazba udržovat si přehled, v jaké úrovni menu se nacházíte?

- Pomohla Vám zvuková zpětná vazba při nalezení žádaného prvku?
- Byla zvuková zpětná vazba rušivým prvkem při procházení menu?

6.2.1 Výsledky a odpovědi

O1	O2	O3	O4
2	4	1	1
2	5	1	3
2	2	1	5
1	4	2	5
1	3	1	4
2	2	1	1
1	5	5	1
2	3	2	5
1.625	3.5	1.75	3.125

Tabulka 6.3. Tabulka odpovědí 2



Obrázek 6.1. Porovnání dob hledání prvků při zvukové zpětné vazbě

První otázka nám opět pomohla potvrdit předpoklad, že menu byla v tomto případě intuitivně rozvržena a hledání prvků v nich by mělo být jednoduchou záležitostí pro uživatele.

Z grafu na Obázku 6.1 je vidět, že při zapnuté funkci zvukové zpětné vazby se rychlost nalezení prvků nějak konstantně nezlepšila. Velké výkyvy v grafu jsou

pravděpodobně zapříčiněny rozdílností obou menu a faktem, že uživatel se mohl v jednom z nich orientovat lépe než v druhém. Díky prohazování menu, s jakým byla funkce zapnuta, bylo ale možné porovnat obě linie při pohledu na celkový trend jejich hodnot.

Odpovědi na otázku 2 sice nasvědčují tomu, že díky zvuku si byli uživatelé lépe vědomi toho, jak hluboko v menu se zrovna nacházejí, ale průměr otázky 3 již více odpovídá naměřeným časům a ukazuje, že při hledání konkrétního prvku v menu se uživatelé orientovali především podle názvů jeho uzlů, které viděli, a znalost současné hloubky v menu jim nepomáhala. Vybrané earcony splnily svůj účel ve funkci poskytnutí informace, ale čtvrtá otázka ukazuje, že jejich výběr nebyl pro uživatele zrovna přívětivý a při hledání jim byly nepříjemné a rušily je.

Kapitola 7
Závěr

Cílem této práce bylo prozkoumat různé alternativy existujících řešení zpracování zvuku v chytrých telefonech a dále pak časté problémy spojené s ovládáním mobilních aplikací. Z nich měl být jeden problém zvolen, pro nějž se navrhnou adaptační scénáře využívající právě vybraný zvukový kontext zařízení jako spouštěč adaptace.

V Kapitole 2 byly popsány různé přístupy k využití zvuku včetně popisu technologií, které jsou v současnosti pro jejich realizaci používány. Ty byly rozděleny do dvou hlavních větví podle typu zdroje zvuku na řeč a okolní zvuky. Pro obě větve bylo nabídnuto několik způsobů jejich využití k adaptaci v reálných životních situacích.

Pro implementační část byla vybrána složka řeči a s ní spojený proces rozpoznávání řeči zejména kvůli spolehlivosti podávaných výstupů a nutnosti implementace na mobilním zařízení. Cílem bylo zjistit, jestli lze detekovat chybný návrh aplikace skrze samovolné slovní projevy uživatele. Jako cíl adaptace bylo zvoleno aplikační menu, se kterým souvisí navigační problémy uživatele v něm.

Pro aplikaci získaného řečového kontextu do adaptace byly implementovány nástroje rozpoznání řeči a analýzy sentimentu a emocí, s jejichž využitím byly vytvořeny dva adaptační scénáře, které upravují rozložení prvků menu. Mimo to byla v rámci zlepšení komunikace zařízení s uživatelem realizována zvuková zpětná vazba navádějící uživatele v hlubokých strukturách menu. Pro ověření hypotéz a použitelnosti navržených funkcionalit bylo provedeno uživatelské testování na jednoduché mobilní aplikaci.

Pro rozeznávání nespokojenosti uživatele z řeči testování ukázalo, že slovní vyjádření při špatném návrhu aplikace je použitelný zdroj informací, protože při procházení menu promluvila v souvislosti s jeho rozvržením polovina respondentů. Problémem však je přesné zachytávání této řeči, protože uživatel mluví sám pro sebe a často tedy nemluví dostatečně nahlas.

Analýza sentimentu se ukázala jako užitečný nástroj pro rozpoznání nespokojenosti uživatele v tom, že lze snadno nastavovat její citlivost a dobře zachytává negativní hodnotu slov. Oproti tomu analýza emocí by nebyla pro tento účel příliš vhodná, protože některé negativní výrazy jako třeba nadávky nejsou jednoznačně

s nějakou emocí spojeny a nespokojenost by nemusela být v tom případě vždy zachycena.

Navržené adaptační scénáře bylo díky malému počtu případů, kdy k jejich provedení došlo, těžké řádně otestovat. Díky nasimulování alespoň jednoho z nich byla adaptace menu přesouváním prvků při jejich špatném zařazení ve stromu výše vyhodnocena jako spíše přívětivá. Adaptaci přesouváním nevhodně zařazených prvků do místa, kde se uživatel nejvíce vyjadřoval, se podařilo otestovat jen na jednom uživateli z důvodů uvedených v sekci Testování. Avšak i jím byla hodnocena také spíše kladně.

Testování zvukové zpětné vazby související s hloubkou procházeného menu podalo výsledek, že tato funkce není velmi užitečná v pomoci při navigaci v hlubokých menu. Informace o současné hloubce byla tímto způsobem uživateli srozumitelně poskytnuta, ale nepomohla urychlit nebo usnadnit nalezení daného prvku.

Vcelku vzato, rešerše ukázala, že zvukový kontext má v oblasti adaptace velký potenciál. V rámci této práce byla vyvinuta aplikace přizpůsobující se okolní řeči, s jejímž využitím se podařilo ověřit hypotézu o uživatelském vyjadřování se ve stavu nespokojenosti. Uživatelé také otestovali adaptační scénáře spolu s funkcí přehrávání earconů. Ačkoliv výsledky nebyly zcela pozitivní, věřím, že přinesly užitečná zjištění a přispějí k dalšímu vývoji v této oblasti.

Kapitola 8
Budoucí práce

V rámci vylepšení současného řešení testovací aplikace může být provedena implementace českého jazyka, aby bylo umožněno snazší testování navržených adaptačních algoritmů díky možnosti testování na větší skupině lidí. Rozpoznávání řeči na platformě Android již tento jazyk podporuje, je proto nutné jen doimplementovat analýzu sentimentu a případně emocí s tímto jazykem do aplikace. Dále může být zdokonalena služba rozpoznávání řeči. Současné řešení je spíše pomocným řešením pro nezbytnou funkčnost a opakované volání služby má vysoké nároky na spotřebu baterie. Rozpoznávání pomocí knihovny PocketSphinx se nabízí jako méně náročná alternativa, ale vyžaduje vložení velkého úsilí do vytvoření vhodného jazykového a akustického modelu pro dosažení lepších výsledků.

Jako pokračování ve výzkumu této práce mohou být navrženy, implementovány a otestovány nové scénáře pro adaptaci menu nebo celé aplikace. Pro jejich realizaci může být sbírání kontextových informací rozšířeno o informace z jiných typů zdrojů než zvukových, aby mohla být adaptace přesněji cílena na potřeby uživatele a zvukový kontext mohl viditelněji projevit svou roli v jejím výsledku. Analýza emocí může být také ještě vylepšena aplikací algoritmu zohledňujícího kontext slov mezi sebou, aby bylo možné porovnat využití rozeznávání emocí s analýzou sentimentu v roli zdroje kontextu k adaptaci.

Co se týče pokračování ve snaze využít zvukový kontext, velký potenciál v sobě má odvětví rozpoznávání okolních zvuků (ESR), které s sebou přináší velké množství do teď neřešených adaptačních scénářů, z nichž některé byly v této práci zmíněny. Pro jeho implementaci by bylo potřeba extrahovat vybrané veličiny z nahraného zvuku a jejich hodnoty zpracovat v nějakém klasifikačním systému ideálně využívajícím neuronové sítě (např. WEKA). Překážkou zde může být i vysoká nepřesnost existujících systémů, kterou by bylo možné vylepšit omezením počtu rozpoznávaných zvuků jen na několik vybraných.

Příloha A
Slovník zkratek

AUI	Adaptive User Interface
SRS	Speech Recognition System
SS	Spectral Subtraction
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
SSOM	Spectral Subtraction with Over subtraction Model
NSS	Non-Linear Spectral Subtraction
ANC	Adaptive Noise Cancellation
VQ	Vector Quantization
DTW	Dynamic time wrapping
HMM	Hidden MArkov Model
ANN	Artificial Neural Network
DNN	Deep Neural Network
ESR	Environmental Sound Recognition
TTS	Text-to-speech

Příloha B
Ukázky kódu

Ukázka kódu 1: Odpočítávání doby rozpoznávání
Ukázka kódu 2: Spuštění časovače
Ukázka kódu 3: Rozeznání otevření menu
Ukázka kódu 4: Vytváření menu
Ukázka kódu 5: Vracení se o úroveň výš
Ukázka kódu 6: Definování metod adaptace

Příloha C
Reference

[1] MAKRIS, Nikolaos. *Creating Adaptable and Adaptive User Interface Implementations in Model Driven Developed Software* [online]. [cit. 2019-04-06]. Dostupné z: https://www.ru.nl/publish/pages/769526/4447603_makris_nikos_-1a.pdf

[2] GULLA, Francesca. *Design Adaptable and Adaptive User Interfaces: A Method to Manage the Information* [online]. 2014 [cit. 2019-04-06]. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-319-18374-9_5

[3] KARAGIANNIDIS, Charalampos. *Supporting Adaptivity in Intelligent User Interfaces: The case of Media and Modalities Allocation* [online]. 1995 [cit. 2019-04-06]. Dostupné z: http://ui4all.ics.forth.gr/UI4ALL-95/karagiannidis.pdf

[4] NIELSEN, Jakob. *First Rule of Usability? Don't Listen to Users* [online]. 2001 [cit. 2019-04-06]. Dostupné z: https://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users/

[5] MÄNTYJÄRVI, Jani. *Sensor-based context recognition for mobile applications* [online]. 2003 [cit. 2019-04-06]. Dostupné z: https://cris.vtt.fi/en/publications/sensor-based-context-recognition-for-mobile-applications-disserta

[6] *Number of mobile phone users worldwide from 2015 to 2020. Statista* [online]. [cit. 2019-03-29]. Dostupné z: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[7] *Mobile Marketing Statistics compilation. Smart insights* [online]. 11 Jul, 2018 [cit. 2019-03-29]. Dostupné z: https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/

[8] CHU, Selina. *Environmental Sound Recognition With Time–Frequency Audio Features* [online]. 2009 [cit. 2019-04-06]. Dostupné z: https://ieeexplore.ieee.org/abstract/document/5109766

[9] KARABETSOS, Sotiris. *Embedded unit selection text-to-speech synthesis for mobile devices* [online]. 2009 [cit. 2019-04-06]. Dostupné z: https://ieeexplore.ieee.org/abstract/document/5174430

[10] WOODFORD, Chris. *Speech synthesizers* [online]. 2018 [cit. 2019-04-06]. Dostupné z: https://www.explainthatstuff.com/how-speech-synthesis-works.html

[11] MEHMOOD, Faizan. *An Overview on Speech Recognition System and Comparative Study of its Approaches* [online]. 2014 [cit. 2019-04-06]. Dostupné z: https://www.researchgate.net/publication/312578015

[12] GANNOT, Sharon. *Audio Source Separation and Speech Enhancement* [online]. 2018, s. 4 [cit. 2019-04-13]. ISBN 9781119279884.Dostupné z: https://www.worldcat.org/title/audio-source-separation-and-speech-enhancement/oclc/1033578988

[13] CHAUDHARI, Amol. *A review on speech enhancement techniques*: 2015 [online]. [cit. 2019-04-13]. Dostupné z: https://ieeexplore.ieee.org/document/7087096

[14] KULKARNI, Devyani S. *A Review of Speech Signal Enhancement Techniques*: 2016 [online]. [cit. 2019-04-13]. Dostupné z: https://pdfs.semanticscholar.org/3849/45a8af8e8428990b6dd790007621560815af.pdf

[15] ABOWD, Gregory D., Anind K. DEY, Peter J. BROWN, Nigel DAVIES, Mark SMITH a Pete STEGGLES. *Towards a Better Understanding of Context and Context-Awareness.* 1999. Dostupné z: https://link.springer.com/chapter/10.1007/3-540-48157-5_29

[16] MUSUMBA. *Context awareness in mobile computing: A review*: 2013 [online]. [cit. 2019-04-13]. Dostupné z: https://ijmla.net/index.php/ijmla/article/download/5/19

[17] TABOADA, Maite. *Sentiment Analysis: An Overview from Linguistics*: 2016 [online]. [cit. 2019-04-13]. Dostupné z: https://www.annualreviews.org/doi/full/10.1146/annurev-linguistics-011415-040518

[18] NUMMENMAA, Lauri. *Opioid system and human emotions*: 2017 [online]. [cit. 2019-04-13]. Dostupné z: https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/bph.13812

[19] RAO, Yanghui. *Building Word-Emotion Mapping Dictionary for Online News* [online]. [cit. 2019-04-13]. Dostupné z: http://ceur-ws.org/Vol-917/SDAD2012_3_Rao.pdf

[20] IKRAM, Sohaib. *Digital audio forensics using background noise* [online]. 2010 [cit. 2019-04-13]. Dostupné z: https://www.researchgate.net/publication/221264335

[21] PILLOS, Angelos. *A real-time environmental sound recognition system for the Android OS* [online]. 2016 [cit. 2019-04-13]. Dostupné z: http://www.cs.tut.fi/sgn/arg/dcase2016/documents/workshop/Pillos-DCASE2016workshop.pdf

[22] WU, Huadong. *Vehicle sound signature recognition by frequency vector principal component analysis* [online]. 1998 [cit. 2019-04-13]. Dostupné z: https://ieeexplore.ieee.org/document/679823

[23] CHACHADA, Sachin. *Environmental sound recognition: A survey* [online]. 2013 [cit. 2019-04-13]. Dostupné z: https://www.researchgate.net/publication/261208284

[24] JOHANNESSON, Gabriel.*Auditory Icons vs Earcons in Games: What Makes the Most Efficient Non-verbal Sound Cue?* [online]. 2016 [cit. 2019-04-13]. Dostupné z: http://www.diva-portal.org/smash/get/diva2:1021223/FULLTEXT02.pdf

[25] WEBSTER, Mark C. *Using Non-Speech Sounds in Mobile Computing Devices* [online]. 2017 [cit. 2019-04-13]. Dostupné z: https://www.researchgate.net/profile/Stephen_Brewster/publication/238722278

[26] MIELKE, Matthias. *Smartphone application for automatic classification of environmental sound* [online]. 2013 [cit. 2019-04-18]. Dostupné z: https://ieeexplore.ieee.org/document/6613407

[27] MEZEI, József. *Drone sound detection* [online]. 2015 [cit. 2019-04-18]. Dostupné z: https://ieeexplore.ieee.org/abstract/document/7382945

[28] BANBURY. *Office noise and employee concentration: Identifying causes of disruption and potential improvements* [online]. 2010 [cit. 2019-04-18]. Dostupné z: https://www.tandfonline.com/doi/abs/10.1080/00140130412331311390

[29] *Environmental Audio Recognition System* [online]. 2017 [cit. 2019-04-18]. Dostupné z: https://github.com/karoldvl/EARS

[30] ROSSI, Mirco. *AmbientSense: A real-time ambient sound recognition system for smartphones* [online]. 2013 [cit. 2019-04-18]. Dostupné z: https://ieeexplore.ieee.org/document/6529487

[31] *TarsosDSP* [online]. 2015 [cit. 2019-04-18]. Dostupné z: https://github.com/JorenSix/TarsosDSP

[32] *PocketSphinx* [online]. [cit. 2019-04-18]. Dostupné z: https://cmusphinx.github.io/wiki/tutorialandroid/

[33] *Sphinx4* [online]. [cit. 2019-04-18]. Dostupné z: https://cmusphinx.github.io/wiki/tutorialsphinx4/

[34] *Microsoft Azure* [online]. [cit. 2019-04-18]. Dostupné z: https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

[35] *Google API* [online]. [cit. 2019-04-18]. Dostupné z: https://cloud.google.com/speech-to-text/

[36] *IBM Watson* [online]. [cit. 2019-04-18]. Dostupné z: https://www.ibm.com/watson/services/speech-to-text/

[37] KALAIVANI, Ashok. *Validation of Stress Assessment using Mobile Phone* [online]. 2016 [cit. 2019-04-18]. Dostupné z: https://www.questia.com/library/journal/1P3-4313929801/validation-of-stress-assessment-using-mobile-phone

[38] ETTORRE. *Speech Emotion Recognition Using MFCCs Extracted from a Mobile Terminal based on ETSI Front End* [online]. 2006 [cit. 2019-04-18]. Dostupné z: https://ieeexplore.ieee.org/document/4129151

[39] JHAWAR, Gunjan. *Speech disorder recognition using MFCC* [online]. 2016 [cit. 2019-04-18]. Dostupné z: https://ieeexplore.ieee.org/document/7754132

[40] ABDOU, Mahdy. *Computer Aided Pronunciation Learning System Using Speech Recognition Techniques* [online]. 2006 [cit. 2019-04-18]. Dostupné z: https://www.researchgate.net/publication/221484767

[41] BREWSTER, Stephen. *Using Non-Speech Sounds in Mobile Computing Devices* [online]. 2014 [cit. 2019-04-18]. Dostupné z: https://www.researchgate.net/profile/Stephen_Brewster/publication/238722278

[42] *VADER-Sentiment-Analysis* [online]. [cit. 2019-04-18]. Dostupné z: https://github.com/cjhutto/vaderSentiment

[43] *NRC Word-Emotion Association Lexicon* [online]. [cit. 2019-04-18]. Dostupné z: https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

[44] MILKA, Martin. *Aspektově orientovaný vývoj adaptivní struktury aplikace pro mobilní aplikace*. 2016. Bakalářská práce. ČVUT FEL.

[45] MISCHENKO, Nikita. *Aspektově orientovaný vývoj adaptivní struktury aplikace pro Java EE aplikace*. 2016. Bakalářská práce. ČVUT FEL.

[46] NIELSEN, Jacob. *How Many Test Users in a Usability Study?* [online]. 2012 [cit. 2019-05-04]. Dostupné z: https://www.nngroup.com/articles/how-many-test-users/

[47] WEICHSELBRAUN, Albert. *Extracting and Grounding Context-Aware Sentiment Lexicons* [online]. 2013 [cit. 2019-05-20]. Dostupné z: https://pdfs.semanticscholar.org/6c06/2cc749711ecfea25f1117a59ac1c1302cb9f.pdf

Příloha D
Obsah CD

Struktura adresářů :

- Texty - složka obsahující tuto práci ve formátu pdf a všechny její zdrojové tex soubory, obrázky a enterprise architect projekt
- testingApp.zip - testovací aplikace