



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE VZDÁLENÉHO A PASIVNÍHO MONITOROVÁNÍ ŘEČI

REMOTE AND PASSIVE SPEECH MONITORING APPLICATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jiří Klimeš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Daniel

Kováč

BRNO 2023

Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Jiří Klimeš

ID: 212561

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Aplikace vzdáleného a pasivního monitorování řeči

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je realizace mobilní aplikace na platformě Android. Tato aplikace bude schopna analyzovat zvukový záznam a extrahované parametry bude odesílat s pomocí internetového připojení na vzdálený server, kde se budou výsledky ukládat. Aplikace bude využita k monitorování řeči pacientů s různými neurodegenerativními onemocněními. Budou popsány možnosti pro vývoj aplikací na operační systém Android. Proběhne návrh prototypu aplikace s následnou implementací s využitím vhodného programovacího jazyka. V rámci práce také proběhne analýza možností nahrávání výstupního signálu mikrofonu během telefonního hovoru a případná integrace do aplikačního softwaru.

DOPORUČENÁ LITERATURA:

- [1] HORTON, John. Android Programming for Beginners. 3rd edition. Packt, 2021, 742 s. ISBN 9781800563438.
- [2] GERBER, Adam a Clifton CRAIG. Learn Android Studio: Build Android Apps Quickly and Effectively. Apress, 2015. ISBN 978-1-4302-6602-0.

Termín zadání: 6.2.2023

Termín odevzdání: 26.5.2023

Vedoucí práce: Ing. Daniel Kováč

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Poruchy motorické realizace řeči u pacientů s Parkinsonovou nemocí, souhrnně označované jako hypokinetická dysartrie, se vyskytují již v rané fázi této nemoci. Jazyk hraje významnou roli při klasifikaci řečníků na zdravé a ty s dysartrií. Autor v práci vysvětluje, jaké prvky řeči jsou nejčastěji postižovány. Dále popisuje, jak fungují mobilní aplikace na operačním systému Android a zda je možné jejich využití pro pasivní a vzdálené monitorování řeči. V práci je podrobně popsána problematika nahrávání telefonních hovorů a jakými způsoby je možné takové řešení realizovat. Následuje návrh takové aplikace a její realizace.

KLÍČOVÁ SLOVA

Parkinsonova nemoc; hypokinetická dysartrie; akustická analýza; vzdálené monitorování; pasivní monitorování; mobilní aplikace; operační systém Android

ABSTRACT

Motor speech disorders in patients with Parkinson's disease collectively referred to as hypokinetic dysarthria, occur in the early phase of the disease. Language plays an essential role in classifying speakers into healthy and those with dysarthria. Author explains which aspects of speech are most often affected. Then explains how mobile applications work on the Android operating system, and if it is possible to use them in passive and distant speech monitoring. The topic of voice call recording is then described and how is it possible to implement this solution. Such application is then designed and developed.

KEYWORDS

Parkinson's disease; hypokinetic dysarthria; acoustic analysis; remote monitoring; passive monitoring; mobile applications; Android operating system

BIBLIOGRAFICKÁ CITACE

KLIMEŠ, Jiří. Aplikace vzdáleného a pasivního monitorování řeči [online]. Brno, 2023 [cit. 2023-05-23]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/151126>.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Daniel Kováč.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Jiří Klimeš*

VUT ID studenta: *212561*

Typ práce: *Bakalářská práce*

Akademický rok: *2022/23*

Téma závěrečné práce: *Aplikace vzdáleného a pasivního monitorování řeči*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy, VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 26. května 2023

podpis autora

*Autor podepisuje pouze v tištěné verzi.

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Danielu Kováčovi za trpělivost, ochotu a podnětné návrhy k práci. Dále za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování této bakalářské práce.

Obsah

ABSTRAKT	3
KLÍČOVÁ SLOVA.....	3
ABSTRACT.....	3
KEYWORDS.....	3
BIBLIOGRAFICKÁ CITACE	4
ÚVOD	9
1. APLIKACE VZDÁLENÉHO A PASIVNÍHO MONITOROVÁNÍ ŘEČI	10
1.1 PARKINSONOVA NEMOC	10
1.2 HLASOVÉ ÚSTROJÍ.....	10
1.3 VADY ŘEČI U PARKINSONOVY NEMOCI	11
1.3.1 Artikulace.....	11
1.3.2 Prozódie.....	11
1.3.3 Fonace	11
1.4 EXISTUJÍCÍ PRÁCE NA TOTO TÉMA	11
1.4.1 Aplikace <i>iPrognosis</i>	12
2. MOBILNÍ APLIKACE.....	13
2.1 NATIVNÍ A WEBOVÉ APLIKACE	13
2.2 OPERAČNÍ SYSTÉMY MOBILNÍCH TELEFONŮ	14
2.2.1 <i>iOS</i>	15
2.2.2 <i>Operační systém Android</i>	15
2.3 APLIKACE PRO OS ANDROID.....	16
2.3.1 <i>Koncepce aplikací systému Android</i>	16
2.3.2 <i>Virtuální stroj</i>	16
2.3.3 <i>Princip nejnižších privilegií</i>	16
2.3.4 <i>Oprávnění a žádosti o přístup</i>	17
2.3.5 <i>Publikace aplikace</i>	17
2.4 PROGRAMOVACÍ JAZYKY.....	17
2.4.1 <i>Jazyk C/C++</i>	18
2.4.2 <i>C#</i>	18
2.4.3 <i>Java</i>	18
2.4.4 <i>Kotlin</i>	19
2.5 ALTERNATIVNÍ MOŽNOSTI VÝVOJE	19
2.5.1 <i>Python</i>	19
2.5.2 <i>BeeWare</i>	20
2.5.3 <i>Kivy Framework</i>	20
2.5.4 <i>Jython</i>	20
2.5.5 <i>Podpora knihoven a přidavných modulů</i>	20
2.5.6 <i>Chaquopy</i>	21
2.6 VÝSLEDNÁ VOLBA PROGRAMOVACÍHO JAZYKA	21
3. NÁVRH APLIKACE	22
3.1 PŮVODNÍ NÁVRH APLIKACE.....	22

3.2	NAHRÁVÁNÍ TELEFONNÍCH HOVORŮ	24
3.2.1	<i>Android 9 a méně</i>	24
3.2.2	<i>Android 10 a více</i>	24
3.2.3	<i>Speciální případy</i>	24
3.2.4	<i>Shrnutí</i>	25
3.3	ŘEŠENÍ PROBLEMATIKY NAHRÁVÁNÍ HOVORŮ	26
3.3.1	<i>Aplikace zprostředkující telefonní hovor</i>	26
3.3.2	<i>Zabudovaná funkce nahrávání hovorů</i>	26
3.3.3	<i>Accessibility API</i>	26
3.4	NOVÝ NÁVRH APLIKACE	27
3.5	ANALÝZA ŘEČI	28
3.5.1	<i>Dostupné modely strojového učení pro Android</i>	28
3.5.2	<i>Praat</i>	28
3.5.3	<i>Parselmouth</i>	28
4.	VÝVOJ APLIKACE	30
4.1	ANDROID STUDIO	30
4.1.1	<i>Typy souborů</i>	30
4.2	UŽIVATELSKÉ ROZHŘANÍ	31
4.2.1	<i>Rozhraní napsané v XML</i>	31
4.2.2	<i>Jetpack Compose</i>	31
4.2.3	<i>XML vs Jetpack Compose</i>	31
4.3	OPRÁVNĚNÍ APLIKACE	31
4.3.1	<i>Požadovaná oprávnění</i>	31
4.4	NAHRÁVÁNÍ ZVUKU	32
4.4.1	<i>Objekt Media Recorder</i>	32
4.4.2	<i>Objekt AudioRecord</i>	32
4.4.3	<i>Externí knihovna Android Wave Recorder</i>	33
4.5	PŘIPOJENÍ K INTERNETU	33
4.6	DATABÁZE ROOM	33
4.7	PLÁNOVÁNÍ PRÁCE POMOCÍ OBJEKTU WORK MANAGER	34
4.8	ŘEČOVÁ CVIČENÍ	34
4.9	ANALÝZA ŘEČOVÉHO SIGNÁLU	34
4.9.1	<i>Chaquopy</i>	35
4.9.2	<i>Analýza na straně serveru</i>	35
4.9.3	<i>Webové rozhraní Flask</i>	36
4.10	LOKALIZACE	36
4.11	RESPONZIVITA	36
4.12	STRUKTURA PŘILOŽENÉHO ZDROJOVÉHO KÓDU	37
	ZÁVĚR	38
	LITERATURA	40
	SEZNAM PŘILOŽENÝCH SOUBORŮ	44

ÚVOD

Parkinsonova nemoc je druhé nejrozšířenější neurodegenerativní onemocnění. Jedná se o nemoc, jež zásadně ovlivňuje život těch, jež jí trpí. Cílem této práce je prozkoumat, jak lze zlepšit život těchto lidí, pomocí moderních technologií, které většina dnešní populace využívá každý den. Jsou zkoumány mobilní telefony a jejich možné využití pro potřeby diagnostiky nemocí a sledování stávajícího stavu pacientů.

Ačkoliv jsou zařízení jako mobilní telefony rozšířeny mezi většinu dnešní populace, nejsou možnosti, jež nám telefony přináší, adekvátně využívány. Oproti ostatním zařízením patří mezi jejich silné stránky kompaktnost, velký výpočetní výkon anebo široké možnosti funkcí a využití, které jsou pro nás schopné zprostředkovat. Ačkoliv v posledních letech došlo ke značnému pokroku v oblasti používání mobilních telefonů pro zrakově, sluchově či jinak postižené uživatele, jedná se stále o první kroky, jež vedou ke zlepšení uživatelského zážitku pro lidi trpící nějakým postižením. Tento pokrok se také většinou zaměřuje na ty, u nichž již zdravotní omezení existuje. Moderní technologie je však možné využít i pro jejich prevenci či včasnou diagnostiku onemocnění.

Práce je koncipována do čtyř hlavních částí. První část se soustředí na vysvětlení tématu. Zaobírá se problematikou Parkinsonovy nemoci a vadami řeči s ní spojenými. Mezi ně patří problémy s artikulací, fonací, prozodií anebo respirační potíže. Součástí této kapitoly je představení, co je cílem této práce, kdo se již touto nebo podobnou problematikou zabíral v minulosti a jaké studie a výzkumy na toto téma již existují.

V druhé části je vysvětlena technická stránka mobilních aplikací a v čem vývoj mobilních aplikací spočívá. Jsou zde popsány možnosti a specifikace operačního systému Android a teoretická stránka vývoje mobilních aplikací. Součástí druhé kapitoly je také vysvětlení a odůvodnění volby konkrétního programovacího jazyka.

Třetí část práce podrobně popisuje úskalí zadání této práce. Jaké možnosti pro tvorbu takovýchto aplikací existují a jaké jsou naopak limitace. Její součástí je také teoretický návrh aplikace. Je vysvětlena problematika nahrávání telefonních hovorů a popsáno za jakých okolností to je možné.

Čtvrtá část se soustředí na samotný vývoj aplikace. Z jakých komponent a kódu, jež byl pro aplikaci napsán, se skládá. Jsou detailněji popsány způsoby, jak je možné jednotlivé požadavky na aplikaci splnit. Poslední část se také zabývá prezentací výstupů z této práce a je předloženo několik řešení, pro budoucí navázání na toto téma.

1. APLIKACE VZDÁLENÉHO A PASIVNÍHO MONITOROVÁNÍ ŘEČI

V této kapitole je zkoumána oblast využití mobilních aplikací pro účel monitorování řeči. Pasivní v tomto významu znamená monitorování řeči při činnostech, během kterých se uživatel aktivně a cíleně nesoustředí na svůj řečový projev, jako například při telefonních hovorech. Nejedná se tedy o aktivní účast pacientů na řečových a logopedických cvičeních, při kterých mohou podávat jiný, zpravidla lepší, výkon než například při běžném rozhovoru. Vzdálené má význam pro monitorování řeči bez nutnosti osobní návštěvy lékaře nebo logopedického střediska. Je tak popsána možnost získávat od uživatelů této aplikace data z libovolného prostředí, kde je pro uživatele možné mít a používat mobilní telefon.

1.1 Parkinsonova nemoc

Parkinsonova nemoc, dále jen PN, je idiopatické onemocnění nervového systému, které se u jejích pacientů projevuje motorickými i nemotorickými symptomy. Jedná se o chronické neurodegenerativní onemocnění, které obvykle postihuje starší část populace, nejčastěji u pacientů ve věku od 60 let výše, ale objevují se i případy mladších pacientů. Je to druhé nejčastěji se vyskytující neurodegenerativní onemocnění. Mezi symptomy nemotorického rázu patří například změny v kognitivních schopnostech, změny v chování, narušený spánkový režim a mnohé další. Mezi motorické symptomy patří třes, bradykineze, což je snížená rychlost pohybu, ztuhlost těla, neschopnost udržet rovnováhu a další. Mezi právě zmíněné motorické symptomy patří i snížené komunikační schopnosti a různé vady řeči, jako poruchy ve fonaci, artikulaci či prozódii. [1]

1.2 Hlasové ústrojí

Cílem hlasového ústrojí je tvorba hlasu. Skládá se z několika částí. Jsou to respirační, fonační a rezonanční ústrojí. Každé z těchto ústrojí zprostředkovává jiný aspekt řeči, a všechna jsou pro zdravou řeč nezbytná.

Průzkumy ukázaly, že okolo 70 % pacientů zaznamenalo zhoršení v jejich řeči jako jeden z prvních symptomů PN. Jednou z hypotéz je, že symptomy pohybové rázu mají podobný vliv i na tělesné struktury zodpovědné za tvorbu hlasu. Řeč u PN bývá popisována jako monotónní, rozechvělá, udýchaná, tichá a chraptivá. [2]

1.3 Vady řeči u Parkinsonovy nemoci

Snížení komunikačních schopností má významný vliv na kvalitu života lidí trpících tímto onemocněním. Bylo zjištěno, že až 90 % pacientů s PN trpí sníženými komunikačními schopnostmi a závažnost těchto problémů zpravidla roste s časem. Snížené komunikační schopnosti a vady v řeči spojené s PN se nazývají hypokinetická dysartrie [3]. Řečové oblasti, které jsou postiženy u pacientů s Parkinsonovou nemocí nejčastěji, jsou fonace, artikulace, prozódie a kognitivně lingvistické aspekty řeči. [4]

1.3.1 Artikulace

U lidí trpících PN se projevují změny v artikulaci, které lze pozorovat u pacientů v různých stádiích nemoci, a které souvisejí s hypokinetickým omezením pohybu. Závažnost těchto omezení se liší mezi jednotlivými pacienty, přesto existují aspekty, které lze pozorovat častěji. Těmito projevy jsou uspěchaná nebo zrychlující se artikulace a zhoršená plynulost řeči. [4]

1.3.2 Prozódie

V kontextu prozódie jsou nejčastěji sledovány aspekty jako základní frekvence hlasu a hladina akustického tlaku řeči. Na téma prozódie u pacientů s PN mezi hlavní zjištění patří, že se u lidí trpících touto nemocí objevuje snížení hladiny akustického tlaku jejich řečového projevu. Průkazné jsou také výsledky v monitorování základního kmitočtu projevu, kde bylo zjištěno, že pacienti s PN mají snížený frekvenční rozsah jejich řeči. [5]

1.3.3 Fonace

Z oblasti fonace je nejčastěji analyzován zvukový signál prodloužené fonace samohlásky, nejčastěji samohlásky „a“, která musí být udržována po určitou dobu.

1.4 Existující práce na toto téma

Pro potřeby zpomalení postupu Parkinsonovy choroby do pokročilejších fází nemoci je nutná včasná diagnostika. Dnes již existují výzkumy a studie zabývající se využitím mobilních telefonů pro potřeby diagnostiky a analýzy PN.

Xu Y et al. provedli průzkum mezi lidmi trpícími PN, ve kterém bylo analyzováno 944 odpovědí z 1163 respondentů. Celkem 90,2 % respondentů odpovědělo, že jsou obeznámeni s tematikou Tele-zdravotnictví a 82,8 % z respondentů tyto možnosti aktivně využívá. [6]

Existují aplikace, které si dávají za cíl zmírnit specifické symptomy. Horin et al. hledali odpověď na otázku, zda by se při užívání mobilní aplikace zlepšil stav pacientů s PN, v oblastech chůze, řeči a zručnosti. Výsledky této práce nepotvrdily u testovaných výrazné zlepšení a aplikace využívaná pro tyto účely nebyla ve své stávající formě dostačující. [7]

1.4.1 Aplikace iPrognosis

Christos Laganas et al. studovali, zda je možné vyvinout metody využívající technologie, jež by byly pro uživatele snadno dostupné, pro dlouhodobé sledování symptomů PN v běžném životě, bez nutnosti návštěv lékařů či logopedických středisek. Jedním z přínosů této metodiky by mělo být zefektivnění diagnostiky PN a následující průběžné monitorování jejich stavu či případného zhoršení symptomů. Informace a data o řeči uživatelů byly získávány z pasivně nahraných telefonních hovorů.

Aplikace využívaná pro získání dat se prokázala jako užitečná. Její hlavní výhodou byla autenticita získaných dat. Jedná se o pasivně získané informace, u kterých se testující nesoustředí na kvalitu jejich řečového projevu, na rozdíl od logopedických a jiných řečových cvičení, kde jsou si testující vědomi toho, že jsou nahráváni a jejich řeč je analyzována. [8]

Aplikace iPrognosis je, že využívala určitých funkcí operačního systému Android v období, kdy byly běžně přístupné. Některé z těchto funkcí byly později zrušeny, nebo jejich využití znemožněno. Také nebyl nalezen podrobný popis aplikace z technického hlediska. Je nastíněno, čeho je schopná, ale ne jakým způsobem toho dosahuje.

2. MOBILNÍ APLIKACE

Mobilní aplikace je program, jinými slovy software, který je určen pro užívání v mobilních telefonech případně tabletech. V kontextu mobilních telefonů se myslí tzv. chytré telefony.[9]

Většina dnešních zařízení, z oblasti mobilních telefonů či tabletů, je již prodávána s předinstalovaným softwarem, jako například webový prohlížeč, kalendář, aplikace zprostředkující službu e-mail a mnohé další. Aplikace, které nebývají předinstalované výrobcem, a nejsou tedy pro koncového uživatele hned přístupné, je většinou nutné do mobilního telefonu stáhnout a nainstalovat. K tomuto účelu jsou využívány distribuční platformy, které jsou specifické pro svůj operační systém. Nejčastější dva operační systémy jsou iOS a Android, jejichž platformy jsou App Store, v telefonech značky Apple a Google Play v telefonech s operačním systémem Android spravovaným a oficiálně distribuovaným firmou Google.

Zatímco mobilní aplikace jsou podobné programům a aplikacím určeným pro stolní počítače a notebooky, co do jejich velikosti a možností jsou zpravidla menší, většinou jednoúčelové a fungují nezávisle na aplikacích ostatních. Je neobvyklé, aby mezi sebou jednotlivé aplikace komunikovaly a pracovaly jich více najednou. V případě že jich více najednou pracuje, fungují v systému priorit, kde některé aplikace mají vyšší prioritu než jiné. Například aplikace „Telefon“ má vyšší prioritu než aplikace „Messenger“. To znamená, že v případě příchozího hovoru přejde aplikace Telefon do popředí. V takovéto situaci se ostatní aplikace většinou nevypnou, ale dočasně utlumí. [10]

2.1 Nativní a webové aplikace

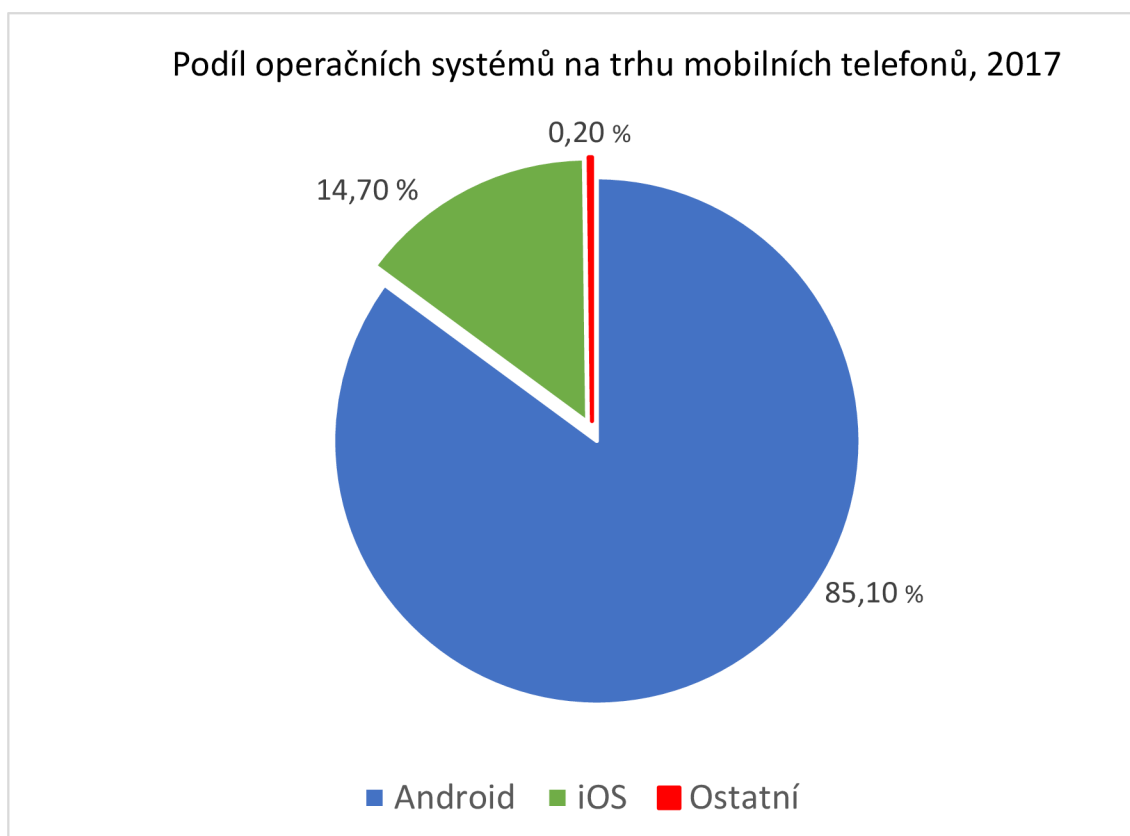
Mobilní aplikace je nutné rozlišovat, na dvě kategorie. Jsou to webové aplikace určené pro web a nativní aplikace, které jsou pouze pro mobilní telefony. Pro nativní aplikace se v naprosté většině případů jedná o aplikace pro Android OS nebo iOS. Mobilní aplikace podléhají více kritériím a požadavkům z hlediska možnosti jejich vývoje a bezpečnosti, mohou však využívat více možností mobilních telefonů.

Webové aplikace jsou určeny pro mobilní telefony ale také stolní počítače, tablety, notebooky a další. K přístupu na webové aplikace je nutné mít připojení k internetu, a libovolný webový prohlížeč. V mobilních telefonech s OS Android je většinou předinstalován prohlížeč Chrome. Tyto aplikace jsou výrazně méně náročné na výkon mobilního telefonu, protože všechny nebo téměř všechny soubory jsou uloženy na serveru a k uživateli zprostředkovány pomocí internetového připojení. Vývoj webových aplikací se s rozšířením chytrých telefonů mezi širokou veřejnost přizpůsobil takzvaným responsivním designem, ve kterém je webová aplikace navržena tak, aby se přizpůsobila velikosti displeje konkrétního zařízení.

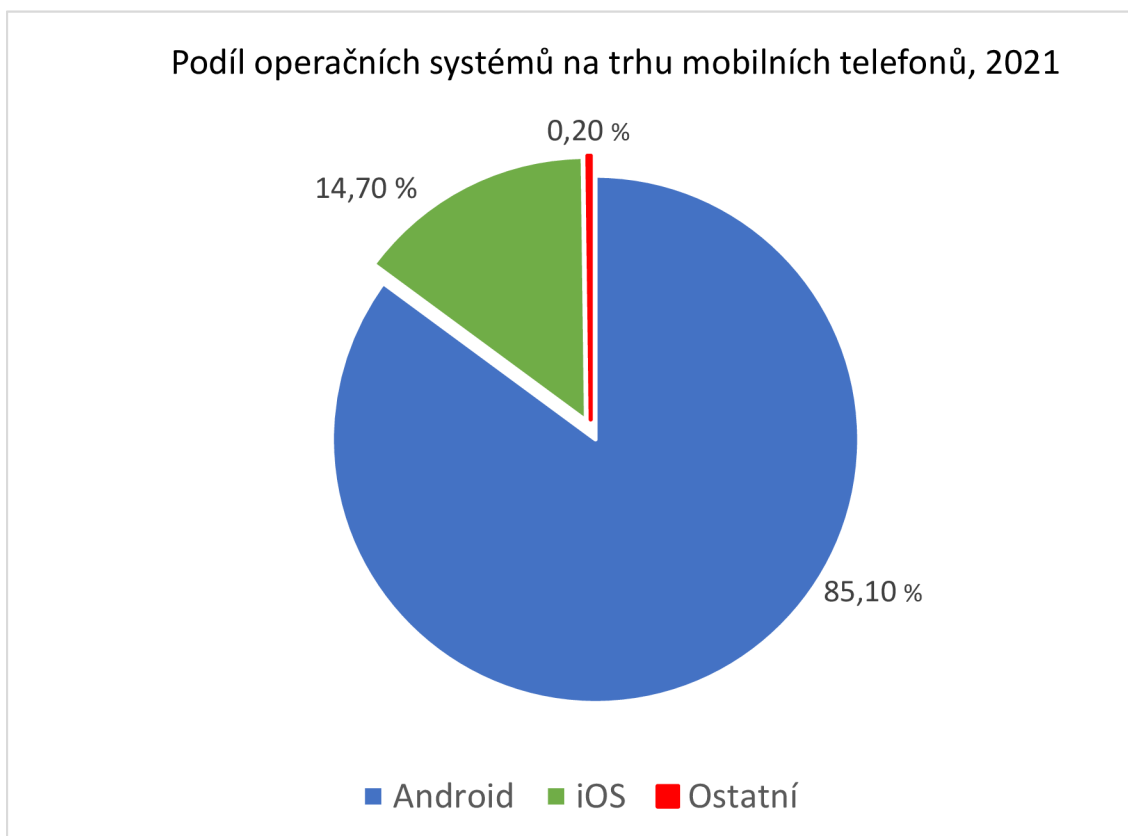
Pro potřeby této práce nebyla zvolena cesta webové aplikace, protože dle zadání je aplikace určena pouze pro mobilní telefony a nebyla by přizpůsobena jiným zařízením. Webové aplikace také nemusí mít přístup k mnohým hardwarovým možnostem mobilních telefonů. Hlavní výhodou webových aplikací je jejich přenositelnost na jiná zařízení. Tato vlastnost by tedy nebyla vůbec využita. Dále je nutné získat přístup k možnostem mobilního telefonu jako je mikrofón, je nutné nahrávat zvukové soubory, zanalyzovat data a ta uložit do paměti, což může u webových aplikací být obtížné.

2.2 Operační systémy mobilních telefonů

Dnešní trh s mobilními telefony je rozdělen do tří kategorií. Zařízení s Android OS, iOS a ostatní operační systémy. Naprostá většina zařízení využívá jeden ze dvou zmíněných operačních systémů, tedy Android OS nebo iOS. V grafech níže je možné vidět, že mezi lety 2017 až 2021 nedošlo k žádné výrazné změně tohoto stavu. Z tohoto důvodu není třetí kategorie vůbec zmíněna.



Obrázek 1: Podíl operačních systémů na trhu mobilních telefonů, 2017 [11]



Obrázek 2: Podíl operačních systémů na trhu mobilních telefonů, 2021 [11]

2.2.1 iOS

Operační systém iOS, dříve iPhone OS, je operační systém od společnosti Apple Inc. Původně byl vyvinut pouze pro telefony typu iPhone, později rozšířen i na další zařízení jako iPadOS, tvOS a watchOS. [12]

iOS je zde zmíněn převážně proto, že je to možný směr, kam by se aplikace mohla v budoucnu dále rozšířit. Postavení firmy Apple a operačního systému iOS je na trhu s mobilními telefony druhý nejvýznamnější konkurent pro OS Android, s potenciálními miliony nových uživatelů. Mohlo by se jednat o platformu, na kterou by bylo možné původní aplikaci rozšířit v případě, že by verze pro Android byla úspěšná.

Bude-li pro vývoj aplikace vybrána technologie určená pouze pro OS Android, nemůže být aplikace pouze „rozšířena“ na operační systém iOS. Bylo by nutné zvolit jinou technologii, programovací jazyk nebo způsob tvorby aplikace. Proto by se muselo jednat o aplikaci se stejnou koncepcí, ale vlastní implementací jednotlivých komponent a funkcí specifických pro iOS. Také s kódem v jazyce určeném pro vývoj aplikací na oba operační systémy, nebo specifickým pro vývoj právě iOS aplikací.

2.2.2 Operační systém Android

Android je operační systém určený pro mobilní telefony a také stejnojmenný projekt otevřeného softwaru společnosti Google LLC. Otevřený software je označení pro

software, jež byl zpřístupněn pro veřejnost, je pak možné nejen nahlédnout do části zdrojového kódu takového softwaru, ale mnohdy ho i upravovat a přispívat k jeho vývoji. [13]

2.3 Aplikace pro OS Android

2.3.1 Koncepce aplikací systému Android

Každá Android aplikace zaručuje větší bezpečnost díky tomu, že funguje samostatně a nezávisle na ostatních. Toho je docíleno tím, že operační systém Android funguje na principu jako systém Linux, ve kterém každá aplikace slouží jako jeden samostatný nezávislý uživatel. Každá aplikace dostane přidělené své vlastní unikátní uživatelské identifikační číslo – ID, které je známé pouze operačnímu systému. Žádná aplikace k tomuto ID nemá přístup, ani aplikace samotná, ke které je toto ID přiděleno. Všechny soubory v konkrétní aplikaci, které jsou uloženy v interní paměti jsou přístupné pouze pro aplikaci samotnou, žádná jiná aplikace k těmto souborům přístup nemá. [14]

2.3.2 Virtuální stroj

Každý proces má svůj vlastní virtuální stroj, někdy také nazýván virtuální počítač, jež se často označuje zkratkou VM (Virtual Machine). Od fyzických počítačů se liší pouze v tom, že nemají vlastní hardware. Existují pouze jako kód, je na ně nahlíženo jako na virtuální nebo softwarově definované. Mimo to fungují stejně jako fyzické počítače, které jsou fyzicky hmatatelné. Virtuální počítače mají vlastní procesor, paměť, mohou se připojit k internetu a mnohé další možnosti. [15]

V praxi to znamená, že jednotlivé aplikace fungují v kompletní izolaci od aplikací ostatních. OS Android spustí proces konkrétní aplikace, když některá z jejích komponent vyžaduje provedení. Obdobně pak ukončí proces v případě, že již není potřebný, nebo když je nutné uvolnit paměť pro běh ostatních aplikací.

2.3.3 Princip nejnižších privilegií

Systém Android využívá princip nejnižších privilegií. V informatice se jedná o princip, či metodu, při které jsou přidělována uživatelům, programům a procesům nejnižší možná oprávnění, která jsou dostatečná pro správnou funkci aplikace. Tato metoda se využívá, pro maximalizaci informační bezpečnosti. Někdy se tento princip nazývá princip nejnižší autority. [16]

Princip nejnižších privilegií tedy zaručuje, že aplikace má přístup pouze k těm částem systému, které jsou nezbytné pro její funkci. Aplikace nemá přístup částem systému, ke kterým nemá oprávnění. To v případě „Aplikace vzdáleného a pasivního monitorování řeči“ znamená, že bude vyžadován přístup k mikrofonu telefonu, protože je nezbytný pro funkčnost aplikace. Přístup například ke kameře mobilního telefonu není nutný, tudíž

nebude možné kameru telefonu v této aplikaci využít. Obdobně funguje práce s úložištěm. Vnitřní úložiště aplikace je přístupné pouze aplikaci samotné.

Existuje možnost, jak zařídit dvěma či více aplikacím, přístup k jednomu vnitřnímu úložišti, a to pomocí již dříve zmíněného ID aplikace. Pro dvě či více různých aplikací lze nastavit jedno konkrétní identifikační uživatelské číslo ID. Každá aplikace běží na svém vlastním virtuálním stroji, ale v případě více aplikací se stejným identifikačním číslem budou všechny tyto aplikace pracovat na jednom virtuálním stroji v rámci stejného procesu. [16]

2.3.4 Oprávnění a žádosti o přístup

U aplikací, které ke své správné funkci vyžadují přístup k některým funkcím telefonu, jako například kamera, fotoaparát, mikrofon, Bluetooth připojení apod., musí o oprávnění využívat tyto funkce požádat. Při tvorbě takové aplikace je nutné vzít tyto požadavky v úvahu, a zahrnout v aplikaci tyto žádosti o přístup k jednotlivým funkcím telefonu. Samotné oprávnění však potvrdí až uživatel, který konkrétní aplikaci chce použít. [16] Způsob, kterým o oprávnění lze požádat a konkrétní oprávnění vyžadované aplikací pro monitorování řeči, stejně jako konkrétní vyžadovaná oprávnění, jako přístup k mikrofonu telefonu, jsou popsány v kapitole „Vývoj Aplikace“.

2.3.5 Publikace aplikace

Při tvorbě aplikace pro zařízení s OS Android je využíváno Android SDK (Android Software Development Kit), vývojové rozhraní určené pro tvorbu softwaru na zařízení s Android OS. Toto vývojové rozhraní zkompiluje (sestaví) všechny kód aplikace včetně dalších zdrojových souborů, jako obrazové soubory, zvukové soubory, animace a další. Na výběr je poté ze dvou souborů: Android Package (APK) a Android App Bundle.

Android Package soubor má koncovku .apk, Android App Bundle má koncovku .aab. APK obsahuje všechny soubory nutné k instalaci a spuštění aplikace. Android App Bundle obsahuje vedle samotného APK souboru ještě další informace, tzv. metadata, která podrobněji specifikují, jak by měla aplikace fungovat na různých mobilních telefonech a v různých verzích operačního systému Android. Android App Bundle je nejčastěji využíván k nahrání aplikace do online distribuční platformy Google Play, ze které je poté možné, za pomoci internetového připojení, instalovat aplikaci do libovolných zařízení, jež běží na OS Android, a tedy mají distribuční platformu Google Play automaticky předinstalovanou. [17]

2.4 Programovací jazyky

Při volbě technologie vývoje a programovacího jazyka, je nutné zvážit, k čemu jsou jednotlivé jazyky určeny. Mezi rozhodující faktory patří rychlost a efektivita při vývoji, udržitelnost a čitelnost kódu, rozsah oficiální dokumentace a komunitní podpory, výkon,

přenositelnost na jiné platformy a zařízení, existující nástroje, knihovny a rozhraní, které lze při vývoji využít a mnohé další.

Aplikace pro OS Android mohou dle oficiální dokumentace být psány v jazyce Kotlin, Java a C++. Existuje však více možností, které jsou zpravidla pro mobilní aplikace méně časté. Tyto alternativy jsou diskutovány v kapitole „Alternativní způsoby vývoje“.

Dále existují tzv. „multiplatformní“ jazyky, tedy programovací jazyky, ve kterých lze vytvářet aplikace pro oba operační systémy, tedy Android i iOS, najednou. Jedná se o jazyk Dart, který je používán v rozhraní Flutter, jazyk Javascript v rozhraní React Native nebo například vývojové rozhraní Apache Cordova. Zde zmíněné programovací jazyky a rozhraní jsou ty nejčastěji používané, tedy zpravidla také s nejobsáhlejší oficiální dokumentací a největší komunitou vývojářů a v moderním vývoji mobilních aplikací nejvíce významné. [10]

2.4.1 Jazyk C/C++

Tvorba aplikací pomocí jazyka C, respektive C++, je možná. Dokonce je tato možnost zmíněna mezi oficiálně podporované a preferované jazyky pro vývoj aplikací pro platformu Android. Ale jedná se o velmi neobvyklý způsob vývoje mobilních aplikací. Mezi hlavní výhody tohoto jazyka patří lepší výkon a optimalizace, nebo možnost použití knihoven určených pro jazyky C či C++. Ačkoliv se jedná o dva odlišné jazyky, kvůli jejich podobnosti jsou v dokumentaci sdružovány do jedné kategorie. [18]

Mezi hlavní nevýhody tohoto přístupu patří výrazně složitější syntaxe. Jazyk C je nižší programovací jazyk, někdy také nazývaný nízkourovňový. V tomto jazyce není poskytována tak výrazná abstrakce mezi vývojáři psaným kódem a samotnými instrukcemi procesoru. Proto i instrukce odpovídají příkazům procesoru, kód je dost složitý a pro začátečníka nesrozumitelný. Obtížnost vývoje by se tak značně zvýšila. [19] Mezi další nevýhody patří, že oficiální dokumentace popisuje vývoj zpravidla pouze v jazycích Java a Kotlin, v některých případech dokonce pouze jazyce Java.

2.4.2 C#

V jazyce C# lze také vyvíjet aplikace pro OS Android. Jedná se o další alternativu, opět méně běžnou, a s méně rozsáhlou podporou i komunitou vývojářů. Oproti jazyku C a C++ je syntaxe C# čitelnější a přívětivější pro začátečníky.

Ačkoliv jazyk C# řeší část problémů, na které naráží jazyky C a C++, tak se stále jedná o méně preferovanou, suboptimální možnost. Není to preferovaný jazyk pro tvorbu Android aplikací, není mu uzpůsobena dokumentace a nemá takovou vývojářskou podporu. Jedná se tedy o možnost určenou spíše pro vývojáře zkušené s jazykem C#, kteří se nechtějí začít věnovat jazyku Java nebo Kotlin.

2.4.3 Java

Jazyk Java byl velmi dlouhou dobu oficiálním a preferovaným jazykem pro vývoj Android aplikací. Oficiálním je stále, ale není již preferovaným. Tím se stal jazyk Kotlin

zmíněný v následující kapitole. Pro značné množství vývojářů je Java stále vhodným jazykem pro tvorbu mobilních aplikací. Jedná se o jazyk, který se používá výrazně déle než Kotlin a má široký rozsah využití.

Mezi její výhody patří dlouhodobá podpora jednotlivých verzí Javy (konkrétně Java 8, 11 či 17). [20]

Jedná se také o poměrně čitelný jazyk. Mezi nevýhody patří velká podrobnost, pro napsání i jednoduché funkce je zapotřebí značné množství kódu. Také je méně výkonný než například jazyk C nebo C++.

2.4.4 Kotlin

Kotlin je staticky psaný programovací jazyk. Staticky psané jazyky vyžadují přiřazení typu proměnné, již při její deklaraci [21]. Kotlin patří mezi jazyky fungující nad JVM (Java Virtual Machine). JVM je virtuální stroj, ve kterém je možné spustit takzvaný *java bytecode*. Úkony, které JVM provádí jsou například načtení a verifikace kódu, spuštění kódu, deklarace místa v paměti a mnohé další. [22]

Kotlin podporuje jak funkční programování, tak objektově orientované programování, které je zvláště významné a využívané při tvorbě mobilních aplikací. Syntaxe jazyka je velmi podobná ostatní jazykům, zaměřeným na objektově orientované programování, jako jazyky Java, C#, Scala a další. Díky tomu, že je Kotlin oficiálně podporovaný jazyk pro vývoj aplikací na operační systém Android, je tomu uzpůsobena i oficiální dokumentace a další nástroje, jako programovací rozhraní a knihovny, jež jsou určeny pro softwarové vývojáře.

2.5 Alternativní možnosti vývoje

Aplikace má za cíl nahrávat a analyzovat řečový signál. Zpracování signálu, konkrétně řeči, je nejčastěji prováděno pomocí jazyků MatLab nebo Python. Jelikož ani jeden z těchto programovacích jazyků není určen ke tvorbě mobilních aplikací, není oficiálně podporovaný způsob jakým implementovat nativní komponenty tohoto jazyka pro mobilní telefony.

Úskalí interoperability mezi vícero programovacími jazyky, a otázka implementace kódu, jež není pro operační systémy Android nativní, tedy nemůže sám o sobě fungovat, bude diskutována a řešena v následujících kapitolách.

2.5.1 Python

Mezi nejzajímavější alternativu vývoje, k již zmíněným možnostem, je volba vytvořit aplikaci v jazyce Python, protože kód určený pro analýzu řeči by mohl být napsán právě v tomto jazyce. Při volbě jazyka Python, nejen pro analýzu řečového signálu, ale i tvorbu samotné aplikace, by došlo ke zjednodušení problematiky propojení funkcí dvou různých programovacích jazyků. Existuje několik softwarů či projektů třetích stran, které si daly

za úkol tuto problematiku vyřešit.

Jedním z hlavních problémů tohoto přístupu je, že i přes veškerou snahu vývojářů stále existují omezení a limitace při vývoji aplikací pomocí těchto nástrojů, rozhraní a knihoven. Mezi hlavní patří například to, že takto vyvíjené aplikace nemají všechny technické možnosti, jako aplikace v oficiálně podporovaných jazycích. Konkrétně jazyk Python má velké množství knihoven a modulů třetích stran, které lze většinou jednoduše využít na platformách, pro které je Python určen. Pokud má být jejich funkcionality přenesena na zařízení s OS Android, je možné že nejsou podporovány a nebudou mít možnost využít všechny své funkce. V některých případech nebude jejich přenositelnost na mobilní telefony vůbec možná. [23]

2.5.2 BeeWare

Snahou projektu BeeWare je přenést Python do mobilních a desktopových aplikací. Poskytuje pro to několik nástrojů a knihoven, které umožňují aplikace v tomto jazyce psát. Dále také tento kód zaobalí do stavu, ve kterém je možné ho na těchto zařízeních spustit. Poskytuje přístup k jednotlivým hardwarovým i softwarovým možnostem mobilních telefonů i stolních počítačů. Na závěr poskytuje potřebné nástroje pro ladění a analyzování kódu. [24]

2.5.3 Kivy Framework

Tento přístup řeší například programovací rozhraní Kivy. Toto rozhraní, jinak nazývané jako Projekt Kivy, je otevřený software, jehož smyslem je také spojit programovací jazyk Python a vývoj mobilních aplikací. Stejně jako zmíněný BeeWare je jeho cílem zajistit vývoj aplikací na stolní počítače a mobilní telefony. [25]

2.5.4 Jython

Jak název napovídá, jedná se o kombinaci jazyků Java a Python. Jedná se o projekt, který si dal za cíl spustit jazyk Python na JVM. Jednou z nevýhod projektu Jython je, že prozatím podporuje pouze verzi Python 2. Podpora verze Python 3 je pouze ve vývoji. [26]

2.5.5 Podpora knihoven a přídatných modulů

V této kapitole byly prezentovány různé jazyky a nástroje, pomocí kterých lze vytvářet aplikace pro OS Android. Dále bylo konkrétně pro jazyk Python uvedeno několik projektů a konkrétních řešení. Dokumentace těchto projektů většinou slibuje podporu standardních knihoven jazyka Python. V případě knihoven třetích stran je ale tato tematika výrazně složitější. Takových knihoven je velké množství. Velmi oblíbené a často používané knihovny, jako například SciPy, jsou těmito nástroji většinou podporované, ale značná část méně obvyklých knihoven a modulů tuto podporu již nemá.

2.5.6 Chaquopy

Projekt Chaquopy má podobný cíl jako dříve zmíněné projekty, například Kivy, Jython nebo BeeWare, ale volí odlišný způsob, kterým se tohoto snaží dosáhnout. Místo přenesení celého vývoje mobilních aplikací do jazyka Python, se snaží přenést jazyk Python, do již existujících aplikací, napsaných v oficiálně podporovaných jazycích jako je Java nebo Kotlin.

Projekt Chaquopy má zajištěnou plnou integraci s nástrojem Gradle, který je používán pro automatizaci sestavování programů. Gradle je běžně používán při vývoji pro OS Android a s jeho pomocí jsou řešeny všechny další závislosti aplikace, které jsou vyžadovány pro její správnou funkčnost [27]. Chaquopy také slibuje podporu mnoha knihoven třetích stran. Mezi ně patří SciPy, OpenCV, TensorFlow a mnohé další [28]. Chaquopy také podporuje verzi 2 i 3 Pythonu.

Jednou z nevýhod tohoto přístupu je, že pro správnou funkci integrovaného Pythonu je nutné mít správný interpret pro jeho kód. Velikost takového softwaru narůstá až o desítky MB, což v případě mobilních aplikací může být velmi relevantní faktor. [29]

2.6 Výsledná volba programovacího jazyka

Jazyk Kotlin byl pro vývoj aplikace zvolen proto, že funguje na JVM a je zajištěna jeho přenositelnost na jiné platformy, které jsou schopné JVM implementovat. Je to relevantní faktor převážně proto, aby aplikace byla přenositelná na různé mobilní telefony, případně počítačové tablety, s operačním systémem Android.

Další důvody volby jazyka Kotlin jsou, že se jedná o oficiálně podporovaný jazyk pro vývoj aplikací na OS Android. Více než 60 % vývojářů pro OS Android využívá při tvorbě aplikací primárně jazyk Kotlin [30]. V oblasti vývoje mobilních aplikací má obsáhlou oficiální dokumentaci, velkou vývojářskou komunitu a je plně integrován do IDE Android Studio, což je integrované vývojové prostředí, určené pro tvorbu aplikací pro OS Android a vývoj těchto aplikací výrazně zjednodušuje.

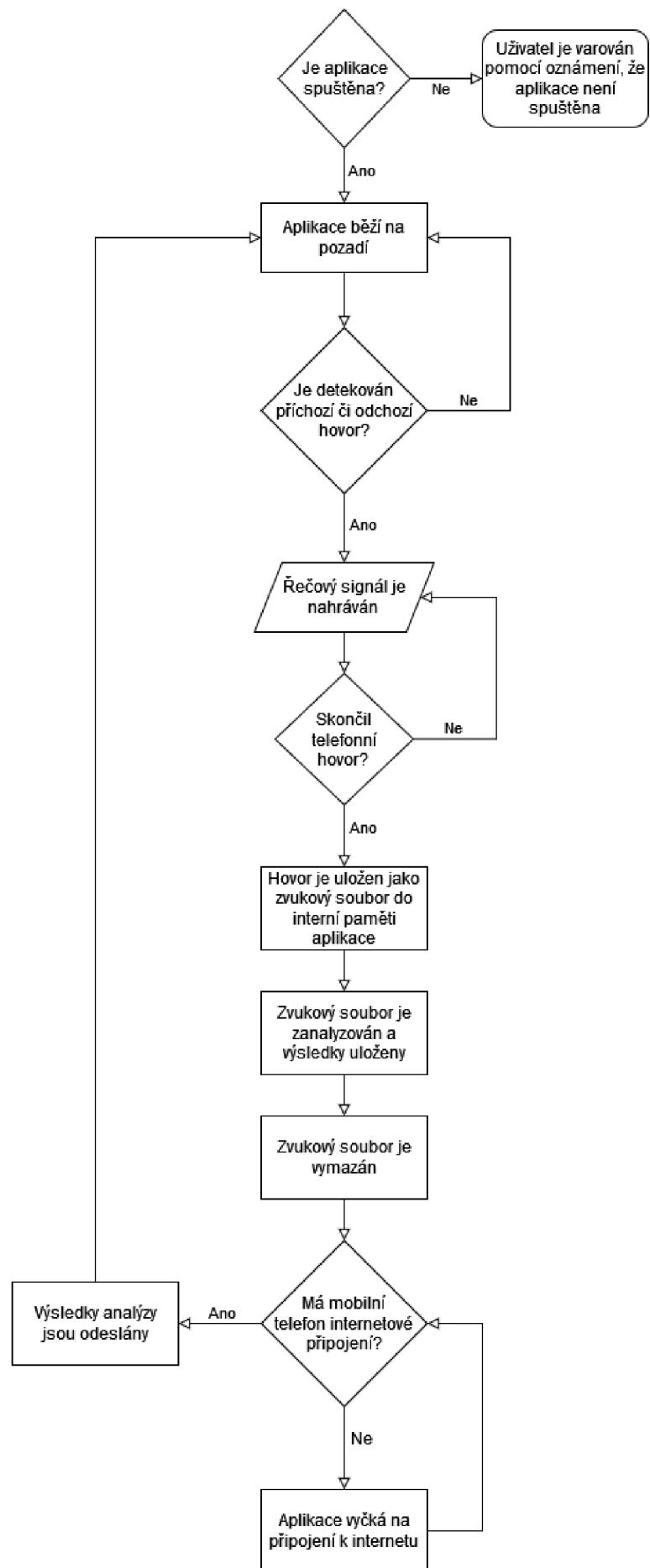
3. NÁVRH APLIKACE

3.1 Původní návrh aplikace

Aplikace má za cíl, pomáhat u pacientů s Parkinsonovou nemocí monitorovat změny v jejich řeči. Je velmi nepravděpodobné, že by u pacientů trpících touto nemocí došlo k náhlému zlepšení v jejich komunikačních schopnostech. V opačném případě tomu tak však může být. V případě náhlého zhoršení by tato aplikace mohla fungovat, jako indikátor neznámých změn, o nichž by měli být informováni lékaři, jež mají konkrétního pacienta ve své péči.

Dalším způsobem, kterým by tato aplikace mohla být prospěšná, je monitorování řeči mezi běžnou populací, jež Parkinsonovou nemocí netrpí. Fungovala by jako preventivní opatření, kdyby došlo ke zhoršení komunikačních schopností způsobem typickým pro Parkinsonovu nemoc. Onemocnění, jímž člověk trpí mnohdy několik let před tím, než se projeví pozorovatelné symptomy, by mohlo být identifikováno dříve, než je u pacientů běžné. Proto i návštěva lékaře a diagnóza nemoci včetně samotné léčby by mohla být nastavena včas, čímž by bylo možné zamezit dalšímu zhoršení nemoci.

Vzdálené monitorování řeči by umožňovalo získávat výsledky analýzy bez nutnosti navštěvovat lékaře. Alternativně by pacientům umožnilo provádět pravidelná cvičení, aniž by museli navštěvovat logopedická střediska, jež jinak bývají pro tento účel určena.



Obrázek 3: Vývojový diagram, původní návrh aplikace

3.2 Nahrávání telefonních hovorů

Bylo zjištěno, že nahrávání telefonních hovorů nemusí být v současné situaci možné. Důvody se liší pro jednotlivé verze operačního systému Android a jsou vysvětleny v následujících kapitolách.

3.2.1 Android 9 a méně

Dle původního zadání nemusí být možné aplikaci realizovat. Společnost Google, se záměrem chránit soukromí uživatelů mobilních telefonů s OS Android, znemožnila nahrávání telefonních hovorů. Pro verze operačního systému nižší než Android 10 není možné mít přístup ke zvukovému signálu z více aplikací najednou. Pouze jedna aplikace je tedy schopná audio signál zaznamenávat. V případě telefonního hovoru je to právě aplikace zprostředkující hovor. Žádná jiná aplikace nemůže za žádných okolností získat přístup ke zvukovému signálu, dokud aplikace, jež v danou chvíli přístup má, neukončí tuto aktivitu, tedy nepřestane nahrávat nebo alespoň zvukový signál snímat. V případě, že by se jakákoliv jiná aplikace pokusila začít zvuk nahrávat, nebude jí k tomu umožněn přístup. [31]

Existující aplikace, které dříve byly hovory schopné nahrávat, využívaly přístup k mikrofonu, přes který nahrávaly zvukový signál obou účastníků hovoru. Většina těchto aplikací je v momentální situaci částečně, nebo úplně nefunkčních.

3.2.2 Android 10 a více

Pro operační systém Android verze 10 došlo ke změně dosavadní koncepce, jak systém se zvukem pracuje. Stále je umožněn přístup pouze jedné aplikaci k audio signálu, ale žádné jiné aplikaci není přístup k tomuto signálu umožněn. V této verzi operačního systému je však možné v průběhu snímání zvukového signálu změnit aplikaci, která zmíněný signál zaznamenává. Nastane-li situace, že dojde ke změně aplikace, která má v daný moment přístup ke zvukovému signálu, aplikace, jež zvukový signál zaznamenávala původně, bude běžet dále, ale místo zvukového signálu nedostane žádný vstupní signál, bude tedy zaznamenávat ticho. Tato funkcionality však také neposkytuje řešení, pro potřeby zadání této práce, tedy aplikaci určenou k pasivnímu monitorování řeči u telefonních hovorů. [31]

3.2.3 Speciální případy

Existují speciální scénáře, ve kterých je možné zvukový signál snímat ze dvou aplikací současně. Ty platí právě pro Android verzi 10 a více. Pro monitorování zvuku jsou aplikace v OS Android rozděleny do dvou hlavních kategorií – privilegované aplikace a běžné aplikace. Privilegované aplikace jsou předinstalované v zařízení, zatímco běžné aplikace jsou nainstalované samotným uživatelem. [31]

Pro funkci těchto aplikací platí určitá pravidla. Níže jsou popsána ta, jež jsou relevantní pro tematiku nahrávání telefonního hovoru. Privilegované aplikace mají vyšší

prioritu než běžné aplikace. Aplikace na popředí, tedy ta, co je právě zobrazována displejem, má vyšší prioritu než ostatní aplikace běžící na pozadí. Dvě běžné aplikace nikdy nemohou mít ke vstupnímu zvukovému signálu přístup najednou. V některých přesně definovaných situacích může privilegovaná aplikace sdílet zvukový signál s běžnou aplikací.

Privilegovaná aplikace může sdílet zvukový signál s běžnou aplikací v těchto případech:

- Google Asistent a běžná aplikace,
- Accessibility service a běžná aplikace,
- dvě běžné aplikace,
- telefonní hovor a běžná aplikace.

Google asistent je předinstalovaná aplikace, která umožňuje uživateli ovládní mobilního telefonu pomocí jeho hlasu, vydáváním úkolů, které má Google Asistent splnit.

Accessibility Service je speciální verze aplikace, která má za úkol zlepšit či zjednodušit používání mobilního telefonu pro zrakově, sluchově nebo jinak zdravotně postižené uživatele. Dvě běžné aplikace, jak bylo vysvětleno dříve, nemohou navzájem sdílet zvukový signál, avšak telefonní hovor může běžnou aplikací sdílet zvukový signál, ale pouze v případě, že je tato běžná aplikace vytvořena právě jako Accessibility service. [31]

3.2.4 Shrnutí

Způsoby, kterými bylo dříve možné telefonní hovory nahrávat, byly částečně, anebo úplně zakázány či znemožněny. Cesty, kterými vývojáři dříve vytvářeli aplikace, které telefonní hovor dokázaly zaznamenávat jsou nyní výrazně omezeny.

V případě telefonního hovoru využívá většina mobilních telefonů předinstalovanou aplikaci „Telefon“. Během telefonního hovoru má také aplikace zprostředkující telefonní hovor vyšší prioritu, ale na konkrétní aplikaci zprostředkující telefonní hovor nezáleží. Jakákoliv jiná aplikace, která je ve stejný moment spuštěna, může běžet pouze na pozadí. V případě, že by se jakákoliv aplikace běžící na pozadí pokusila začít zvuk nahrávat, nebude jí k tomu umožněn přístup. V případě že nahrávání se podaří začít, bude s chybou (errorem) ukončena. I v krajním případě, že by se nahrávání podařilo spustit, operační systém telefonu neumožní snímat mikrofon z více zdrojů najednou. Možná řešení této problematiky jsou popsána v následující kapitole.

3.3 Řešení problematiky nahrávání hovorů

3.3.1 Aplikace zprostředkující telefonní hovor

Jednou z možností, jak problematiku nahrávání hovoru vyřešit, je vytvořit aplikaci, která by zprostředkovala samotný hovor. V takovém případě by mělo být možné, získat přístup ke zvukovému signálu vstupujícímu do mikrofonu telefonu, a tento signál nahrávat. To se ovšem potýká s dvěma problémy. V oficiální vývojářské dokumentaci nebylo nalezeno, zda je toto řešení možné. Ostatní zdroje, které naznačují, že by toto provedení mělo být proveditelné, nedodávají dostatečné informace, jež by tuto hypotézu potvrdily, ani funkční příklad tohoto řešení. Nejedná se o zpravidla věrohodné zdroje, ale spíše hypotézy, které byly nalezeny na různých fórech a v komentářových sekcích pod články a videi, nebo na webové stránce „Stack Overflow“, což je webová stránka sloužící k psaní dotazovacích příspěvků pro vývojáře. Bylo by tedy nutné, zabývat se touto tematikou podrobněji a nalézt relevantní informace, které by jednoznačně potvrdily, nebo vyvrátily tuto možnost. Alternativně přímo realizovat toto řešení, aby bylo zjištěno, zda to možné je či nikoliv. Dalším problémem takového provedení je, že velikost a obsáhlost aplikace by byla značně rozšířena. Obtížnost, a doba nutná k jejímu vývoji, by výrazně vzrostla, a přesto by nebyla zaručena proveditelnost tohoto řešení.

3.3.2 Zabudovaná funkce nahrávání hovorů

Druhým možným řešením je možnost nahrávat telefonní hovory, jež je zabudována přímo v mobilních telefonech [32]. Však i toto řešení se potýká s problémy. Zda je možné nahrávat telefonní hovory je v tomto případě určeno tím, zda výrobce telefonu zpřístupnil tuto funkci. Nelze tedy žádným způsobem zaručit, aby tato funkce byla přístupná právě pro uživatele, kteří by ji mohli využít. Funkce nahrávat hovory také nahrává celý hovor, tedy obě osoby, jež se hovorů účastní. Proto by bylo pro potřeby aplikace pasivního monitorování řeči nutné v nahrávce hovoru od sebe oddělit tyto signály, a pracovat pouze se vstupním zvukovým signálem od uživatele jehož řeč má být monitorována. Tato funkce také není přístupná ve většině zemí Evropy včetně České republiky. Je tedy pravděpodobné, že i přesto, že je přístupnost této funkce určena výrobcem, nelze zaručit, že ji lze v České republice vůbec využít. [33]

3.3.3 Accessibility API

Třetím možným řešením je využití rozhraní Accessibility API, což je sada funkcí, metod a dalších specifikací, které jsou určeny pro lidi s postižením. Oficiálním stanoviskem firmy Google je, že nesmí být používáno k nahrávání telefonních hovorů, protože smyslem tohoto rozhraní je pouze zjednodušení používání mobilního telefonu a jeho funkcí pro lidi s postižením. Využívání Accessibility API je dle jejich „Google Play Developer Policy Updates – April 2022“ zneužíváním této funkcionality. Vývojáři Android aplikací ji dříve využívali právě pro to, aby jejich aplikace byly schopné hovory nahrávat, protože ostatní způsoby nahrávání hovory byly znemožněny. Možnost

nahrávání hovoru by mohla být údajně zneužívána a jakákoliv aplikace, která by tuto funkcionalitu slibovala, by byla dle dostupných informací od firmy Google zakázána a smazána z distribuční platformy Google Play. [34]

Účelem aplikace pro pasivní a vzdálené monitorování řeči je zlepšit diagnostiku v oblasti Parkinsonovy nemoci. Jedná se například o možnost monitorovat komunikační schopnosti pacientů sledováním parametrů jako například základní frekvence hlasu. Teoreticky by taková aplikace splňovala požadavky pro to, aby mohla Accessibility API využívat, ale jedná se velmi o nejasnou problematiku. Na jedné straně je stanovisko společnosti Google jasné a netoleruje zneužívání Accessibility API a všech aplikací definovaných jako Accessibility Service. Nahrávání hovorů je znemožňováno jak na úrovni operačního systému, tak na distribuční platformě Google Play. Na druhou stranu by taková aplikace měla za cíl přesně to, k čemu je Accessibility Service určena.

Navíc, není dostupné dostatečné množství informací, které by prokázalo či vyvrátilo, že lze funkci zaručit a bylo by tedy nutné se touto možností zabývat podrobněji a nalézt průkazné informace o využitelnosti Accessibility Service pro potřeby podobné aplikace. V případě, že by bylo potvrzeno, zda je možné využít tuto funkcionalitu k nahrávání telefonních hovorů, bylo by dále nutné zkoumat, jakým způsobem jsou takové aplikace vyvíjeny, a v čem se liší od běžných aplikací, které toto rozhraní nevyužívají.

3.4 Nový návrh aplikace

Alternativním řešením je nový návrh aplikace, ve kterém je nahráván a analyzován řečový signál pacientů léčených s Parkinsonovou nemocí při konkrétních cvičeních, u kterých jsou sledovány konkrétní parametry řeči, za účelem monitorování jejich zdravotního stavu a komunikačních schopností. Tímto způsobem by mohla být nahrazena logopedická cvičení, která pacienti jinak podstupují přímo u svého lékaře.

V této práci jsou využita konkrétně dvě, která se právě pro tyto účely používají. Jedním z nich je popis obrázku, u kterého je sledován parametr základní frekvence řečového signálu. Druhým cvičením je prodloužená fonace hlásky „A“. V případě že by taková aplikace měla být publikována a používána, bylo by možné aplikaci rozšířit o další cvičení.

3.5 Analýza řeči

3.5.1 Dostupné modely strojového učení pro Android

Ve vývojářské dokumentaci pro Android OS je možné najít modely strojového učení určené právě pro mobilní telefony s tímto systémem. Tyto modely jsou optimalizované a je možné je bez větších obtíží zabudovat přímo do našich aplikací. Dle této dokumentace jsou schopné s nízkou latencí zpracovávat text, audio i video v reálném čase. Umožňují také udržovat data přímo v zařízení bez nutnosti implementace cloudu a může fungovat offline, tedy bez nutnosti mít přístup k internetu.

Mezi tyto modely patří ty určené pro obraz a video, například rozpoznávání obličeje, rozpoznávání a extrakce textu z fotek, segmentace „selfie“ fotek, kde lze oddělit uživatele od pozadí fotky a další. Mezi modely určené pro jazyk patří identifikace jazyka textu z několika slov nebo překlad obsahu mezi až 58 jazyky. [35]

Dále je v dokumentaci zmíněna knihovna TensorFlow, respektive TensorFlow Lite, která je určena pro mobilní telefony. V ní je možné si vybrat úplně nový model, nebo již existující a přetrénovat jej dle konkrétních potřeb. Pak jej konvertovat do podoby méně náročné na hardware. Nasadit tento model do mobilního zařízení a případně jej dále optimalizovat pro lepší výkon. [36]

Model pracující se zvukem je například „automatic speech recognition“, který lze naučit hlasové příkazy. Krátká slova nebo instrukce, jako „doprava“, „doleva“, „nahoru“ a podobně. Případně je možné tento model naučit na ruchy, jako tlesknutí rukou či lusknutí prsty. [37]

Žádný z příkladů uvedených v dokumentaci strojového učení pro OS Android neodpovídá požadavkům tohoto zadání. Časem se bude počet modelů s největší pravděpodobností rozšiřovat, takže je možné, že tyto požadavky mohou být v budoucnu naplněny.

3.5.2 Praat

Existuje celá řada programů a softwarů, které mají za cíl zpracovávat zvukový signál. Daly by se rozdělit na dvě základní kategorie, a to ty, jenž jsou určeny pro analýzu hudby, nebo pro analýzu řeči. Pro případ této práce je relevantní analýza řeči. Proto je možné použít software Praat. Ten umožňuje široké rozpětí práce s řečovým signálem. Vedle zmíněné analýzy je možné jej využít pro syntézu řeči, její úpravu, jako například změnu intenzity, délky nebo její výšky, dále je možné provádět například statistickou analýzu. [38]

3.5.3 Parselmouth

Kód softwaru Praat je napsán v jazycích C a C++. Ty je sice možné v Android aplikacích použít, ale jedná se o poměrně složitý proces integrace. Existuje však knihovna používaná pro tyto potřeby se nazývá Parselmouth. Jejím cílem je poskytnout kompletní funkcionalitu softwaru Praat v jazyce Python. Parselmouth má přístup k samotnému kódu

softwaru Praat, tedy jazyku C++, ale zároveň se chová stejně jako libovolná jiná Python knihovna třetích stran. [39]

4. VÝVOJ APLIKACE

4.1 Android Studio

Android Studio je oficiální integrované vývojové prostředí pro tvorbu aplikací pro OS Android. Kromě editoru kódu a vývojářských nástrojů, jako je nástroj pro debugování kódu, zabudovaný příkazový řádek nebo nástroj pro automatizaci sestavování programu Gradle, má Android Studio řadu dalších funkcí, určených pro zvýšení produktivity a efektivity při vývoji mobilních aplikací. Jedním z nich je emulátor, jež je speciální typ softwaru, který umožňuje spuštění a testování mobilních aplikací i na jiných zařízeních, nejčastěji na stolních počítačích, na kterých jsou mobilní aplikace zpravidla vyvíjeny. [40]

4.1.1 Typy souborů

Při tvorbě aplikací pro Android jsou nejčastější dva typy souborů. Soubory s koncovkou .kt, při volbě jazyka Kotlin, případně .java, pro jazyk Java a soubory s koncovkou .xml jsou soubory tzv. Extensible Markup Language (XML).

V této práci je samotná programovací logika řešena v jazyce Kotlin, jazyk XML je využíván při tvorbě uživatelského rozhraní. Dále je používán v souboru AndroidManifest.xml, který je nedílnou součástí každé aplikace pro OS Android a je nezbytný pro jejich funkci. V tomto souboru jsou definovány všechny komponenty aplikace a jejich parametry. Jsou v něm také definována oprávnění nutná pro správnou funkci aplikace.

4.2 Uživatelské rozhraní

4.2.1 Rozhraní napsané v XML

Uživatelské rozhraní (User Interface, dále jen UI) je vytvářeno v jazyce XML. Pro Android aplikace platí přesně definovaná pravidla, jak má kód vypadat a jaké parametry může či musí definovat. Některé jsou totiž povinné a aplikace bez nich není schopná fungovat. Příklad povinných parametrů textového pole je `layout_width` a `layout_height`.

```
<TextView
    android:id="@+id/tvTimer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="00:00.00"
    android:textSize="56sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Příklad kódu: tvorba textového pole

Poznámka: Jedná se pouze o ukázky a příklady kódu aplikace, celý zdrojový kód je přiložen k dokumentu jako příloha ZdrojovyKod.zip

4.2.2 Jetpack Compose

Alternativním způsobem, jak vytvářet UI pomocí moderní sady nástroje Jetpack Compose. Ten pro tyto potřeby odebrává nutnost jazyka XML a umožňuje vytvořit celé UI pouze v jazyce Kotlin. Díky tomu je možné téměř celou aplikaci vytvořit pomocí jednoho programovacího jazyka. Jetpack Compose volí deklarativní přístup tvorby UI, je proto stručnější, lépe čitelný, v mnoha případech má lepší výkon. Složitě UI napsané pomocí XML mohou být náročné na renderování.

4.2.3 XML vs Jetpack Compose

V případě této aplikace začalo UI být tvořeno v jazyce XML a byla v něm vytvořená značná část práce. Na možnosti a výhody nástroje Jetpack Compose se přišlo v situaci kdy už nebylo žádoucí aplikaci přepisovat. Samotná logika na pozadí aplikace funguje beze změny, pouze UI by bylo vytvářeno a renderováno pro uživatele jinak.

4.3 Oprávnění aplikace

4.3.1 Požadovaná oprávnění

Pro funkci aplikace je nutné mít patřičná oprávnění. Jedním z nich je oprávnění nahrávat zvuk. Nestačí získat toto oprávnění jednou, které bude platit na trvalo, dokud je aplikace nainstalována, ale pokaždé když má být nahráván zvukový vstup, je nutné znovu

zkontrolovat, že toto oprávnění přetrvává. Uživatelé mají tři možnosti, povolit aplikaci nahrávání zvuku na dobu neurčitou, povolit aplikaci nahrávání zvuku pouze tentokrát, nebo při příštím užívání.

Jako první je nutné uvědomit aplikaci, že budete oprávnění vyžadovat. K tomu je určen soubor `AndroidManifest.xml`, ve kterém je nutné všechna oprávnění definovat.

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
```

4.4 Nahrávání zvuku

Nahrávání zvuku je poměrně jednoduchý úkol a OS Android má proto dedikované některé třídy. Stačí pak vytvořit instanci některé těchto tříd, jinými slovy objekt, pomocí kterého je možné zaznamenávat mikrofon zařízení.

4.4.1 Objekt `Media Recorder`

Nejjednodušší je použití objektu `Media Recorder`. Stačí jej definovat na začátku programu jako proměnnou, která bude inicializována později za běhu programu.

```
private lateinit var recorder: MediaRecorder
```

Samotný objekt je tedy inicializován až když je započato nahrávání. Před nahráváním je nejprve nutné zkontrolovat, zda uživatel zpřístupnil aplikaci požadovaná oprávnění, jinak nahrávání nemůže probíhat.

```
private fun startRecording() {
    if (!hasPermissions()) {
        requestPermissions()
    }
    ...
}
```

Problémem třídy či objektu `Media Recorder` je, že není schopen nahrávat soubory ve formátu `Wave`. Je tak možné nahrávat například soubory `MP3`, `3GPP` nebo `OGG`, ale pro nahrání souboru ve formátu `Wave` je nutné zvolit jiný způsob nahrávání. [41]

4.4.2 Objekt `AudioRecord`

Pro OS Android je přístupná druhá třída, respektive objekt, který je schopný nahrávat zvukové soubory. Tento objekt je tzv. „blíže k hardwaru“ a je pomocí něj možné nahrávat nekomprimovaná data a ta poté ukládat. Je tedy možné je dále upravit do formátu `Wave`.

Tento přístup má větší možnosti v oblasti konfigurace takového nahrávání, ale jeho implementace je výrazně složitější. [42]

4.4.3 Externí knihovna Android Wave Recorder

Android Wave Recorder je knihovna napsaná v jazyce Kotlin pro potřeby nahrávání souborů ve formátu Wave. Je určena pro implementaci do aplikací pro Android OS, má jednoduchý zápis a je také možné ji nakonfigurovat pro potřeby aplikace.

```
val waveRecorder = WaveRecorder("$dirPath$fileName.wav")
waveRecorder.waveConfig.sampleRate = 44100
waveRecorder.waveConfig.channels = AudioFormat.CHANNEL_IN_MONO
waveRecorder.waveConfig.audioEncoding = AudioFormat.ENCODING_PCM_16BIT
waveRecorder.startRecording()
```

V této aplikaci byl tento objekt nakonfigurován dle příkladu výše. Pro jeho správnou funkci musí být také naimportován v systému Gradle do souboru „build.gradle“ na úrovni aplikace (v projektu se nachází ještě druhý build.gradle na úrovni projektu). [43]

```
dependencies{
    implementation 'com.github.squti:Android-Wave-Recorder:1.7.0'
}
```

4.5 Připojení k internetu

Pro monitorování internetového připojení je možné použít objekt ConnectivityManager, pomocí kterého je možné sledovat stav a druh připojení. Je tedy možné určit kdy je zařízení připojené k internetu a zda se jedná o připojení pomocí wifi, mobilních dat, nebo přes síťový kabel (ethernet).

Pro posílání dat byl vybrán http protokol. Ten umožňuje provést příkaz „Post Request“, ve kterém jsou odeslány požadované informace na server, na kterém jsou dále zpracovány. Pomocí příkazu typu Post je možné poslat parametry, například samostatné číslo, nebo pole čísel, ale je možné posílat také celé soubory.

4.6 Databáze Room

Aplikace, které pracují s větším množstvím dat mohou využít pro zjednodušení své práce integrovanou lokální databázi Room. Jedná se o SQL databázi, tedy využívající jazyk „Structured Query Language“. Pro používání databáze je také nutné ji definovat v souboru build.gradle.

```
implementation "androidx.room:room-runtime:2.5.1"
kapt "androidx.room:room-compiler:2.5.1"
implementation "androidx.room:room-ktx:2.5.1"
```

V případě této aplikace je diskutabilní, zda je implementace databáze nutná. V principu by nemělo docházet k práci s velkým množstvím dat, ale je potřeba je efektivně ukládat a zařídit, aby bylo možné je ve správný čas pomocí internetového připojení odeslat. Po odeslání je za cíl tato data vymazat. V případech že připojení k internetu není zrovna možné, zůstanou výsledky v databázi a může se s nimi na připojení vyčkat. [44]

4.7 Plánování práce pomocí objektu Work Manager

OS Android má několik způsobů, jak zařídit, aby požadované úkony proběhly úspěšně a ve správném pořadí. Jedním z nich je objekt WorkManager. S jeho pomocí je možné pravidelně provádět předem definovanou práci v konkrétním časovém intervalu. V našem případě je cílem zařídit vymazání nahraných audio souborů, po tom, co již není nutné, aby byly v zařízení uloženy. Zamezí se tak zbytečnému využívání paměti zařízení.

S pomocí tohoto objektu je také možné patřičné výsledky odesílat za pomoci internetového připojení. Je totiž možné definovat omezení nebo požadavky pro to, aby byly tyto úkony provedeny. Například – Work Manager vykoná, co mu bylo zadáno pouze v případě, že má zařízení přístup k internetu nebo má dostatečně nabitou baterii. [45]

4.8 Řečová cvičení

V aplikaci jsou implementována dvě cvičení. Při stisknutí tlačítka je jedno náhodně vybráno. Uživatel je v aplikaci přesměrován do nahrávacího rozhraní, kde je popsáno zadání dané úlohy. Jedním je popis obrázku, u něhož je sledována základní frekvence řeči, s níž je možné kvantifikovat její monotónnost. Druhým cvičením je prodloužená fonace hlásky „A“. U tohoto cvičení je sledován stejný parametr, který v tomto případě kvantifikuje nestabilitu fonace. Poté zmáčknutím tlačítka pro nahrávání může začít úlohu vypracovávat. Nahrávky je možné zobrazit a nepovedené nahrávky vymazat. S nahranými zvukovými soubory z úspěšně dokončených cvičení nemusí nijak dále interagovat. Ty jsou pravidelně zpracovány, odeslány ve vhodné formě na vzdálený server a poté vymazány.

4.9 Analýza řečového signálu

V ideálním případě by bylo možné provádět analýzu přímo na zařízení uživatele. Tím by bylo zajištěno, že se hlasové nahrávky zpracují, výsledky uloží a původní zvukový soubor vymaže. Tyto výsledky by se nadále mohly odeslat na vzdálený server. V případě že by se pro tyto účely použila některá z populárních a široce rozšířených knihoven jazyka

Python, by toto řešení bylo proveditelné.

Avšak knihovna Parselmouth je dle uvedených informací v její dokumentaci stále ve vývoji. Dále se jedná o velmi specifickou knihovnu a bohužel se ji pro tyto účely nepodařilo do aplikace úspěšně implementovat. Ačkoliv existují způsoby, kterými lze jazyk Python na OS Android využít, tak pro tento konkrétní případ je proveditelnost těchto řešení přinejmenším nejistá. V oficiální dokumentaci jednotlivých projektů není dohledatelné, zda je s jejich pomocí možné knihovnu Parselmouth do mobilních telefonů integrovat. Pro ověření či vyvrácení této hypotézy, by bylo nutné pro jednotlivé technologie, programovací jazyky a nástroje, zmíněné v kapitole „2.4 Programovací jazyky“ a „2.5 Alternativní možnosti vývoje“, takovou aplikaci vytvořit.

4.9.1 Chaquopy

Sada nástrojů Chaquopy, jejímž cílem je umožnit integraci kódu v jazyce Python umožňuje importovat širokou řadu knihoven a modulů, jako například SciPy, OpenCV, TensorFlow a další. Dle dokumentace takto podporuje implementaci až 90 % z 1000 nejpoužívanějších Python knihoven [46]. Knihovna Parselmouth pomocí Chaquopy podporu prozatím nemá. Nástroj používaný k adaptování Python knihoven do podoby použitelné pro Chaquopy patří od 2. března 2023 k otevřenému softwaru, je tedy pro uživatele dostupný a do budoucna by bylo možné knihovnu Parselmouth tímto způsobem upravit [47].

4.9.2 Analýza na straně serveru

Jednou z možných variant je provádět analýzu na straně serveru. Klient poskytne potřebná data, v našem případě nahrávku řeči, a s pomocí internetového připojení a za použití vhodného protokolu bude tato nahrávka odeslána. Toto provedení je také uvedeno mezi příklady použití v dokumentaci knihovny Parselmouth. [49]

Pravděpodobně největší výhodou tohoto provedení je nezávislost na platformě a programovacím jazyce. Je pak nepodstatné, v jakém jazyce je aplikace napsána a na jaké platformě implementována. Stačí aby na ni bylo možné provádět dostatečně kvalitní audio nahrávky a připojit se k internetu. Další výhodou je, že je možné upravit značnou část aplikační logiky, včetně samotné analýzy řečového signálu, na serverové straně bez výrazných změn v aplikaci na straně klienta (uživatele). Potřebná data se takto dostanou rovnou na server. Tato funkcionality by musela být implementována bez ohledu na to kde by probíhala analýza.

Toto provedení má ale také řadu nevýhod. Mezi první patří vyšší nároky na implementovaný server. Aplikace je v tomto provedení také schopná nahrávky pouze ukládat, ale software Praat, respektive knihovnu Parselmouth, přenášející jeho funkcionality do jazyka Python není v tomto stavu možné použít přímo v mobilním zařízení.

Další nevýhodou tohoto provedení je nutnost, aby uživatel měl připojení k internetu. To by bylo potřeba zajistit v případě i kdyby analýza proběhla na straně uživatele, aby se

získaná data dostala na server. Ale v případě, že by se místo výsledků analýzy posílal celý soubor s nahrávkou, bylo by zapotřebí stabilní internetové připojení, po kterém by bylo nutné posílat značné množství dat. V případě mobilních telefonů se jedná nejčastěji o mobilní data nebo připojení k wifi. Mobilní data, pokud je uživatel vůbec má zpřístupněna, jsou omezena mobilním operátorem, a není žádoucí za jejich pomoci posílat velké soubory. Wifi připojení by bylo pro tyto potřeby dostačující, ale opět se jedná o případ, kdy musí uživatel mít možnost se takto na internet připojit.

Jedná se také o osobní citlivá data uživatelů a bylo by potřeba zařídit, že takový server bude dostatečně zabezpečený a nebude možné jej napadnout a tato data zneužít. Uživatelé by také museli důvěřovat, že jsou nahrávky ze serveru po analýze vymazány.

4.9.3 Webové rozhraní Flask

Flask je webové rozhraní napsané v jazyce Python. Proto je možné využít tento jazyk i veškeré knihovny pro něj napsané, jak ty standardní, tak knihovny třetích stran jako například právě Parselmouth.

Pro implementaci aplikace při použití tohoto rozhraní stačí pouze jediný soubor. Nastavit ho proto nevyžaduje mnoho práce. Základní serverová aplikace, jenž zpracovává řečový signál je v tomto případě použita, proto aby přijímala data odeslaná z mobilní aplikace příkazem „post“ z protokolu http. Na serveru se tato data zpracují, potřebné hodnoty jsou uloženy a původní soubor je vymazán.

4.10 Lokalizace

Veškerý text v aplikaci je napsán v anglickém jazyce, tedy je lokalizován pro anglicky mluvící cílovou skupinu. Většina textu je takzvaně „pevně zakódována“. Alternativně by mohla být stejným způsobem napsána v českém jazyce.

Nejlepším řešením by bylo upravit aplikaci tak, aby byl překlad automatizován. Lokalizace do jednotlivých jazyků by probíhala automaticky podle nastavení zařízení. Aplikace by detekovala jazyk, na který je mobilní telefon nastaven. Poté by stačilo doplnit potřebné zdrojové soubory pro překlad.

4.11 Responzivita

Aplikace v tomto stavu je uzpůsobena pouze pro chytré telefony. Její uživatelské rozhraní (UI) není upravené pro menší zařízení, jako chytré hodinky, či větší zařízení, jako jsou například tablety. Pro tyto účely by bylo nutné jednotlivé parametry komponenty UI upravit tak, aby se aplikace přizpůsobila velikosti displeje.

4.12 Struktura přiloženého zdrojového kódu

Všechen zdrojový kód je přiložen ve ZdrojovyKod.zip. Ten je rozdělen do několika složek. Ty jsou pojmenovány tak, aby reflektovaly jejich funkci i název v IDE Android Studio. Jejich popis je možné nalézt v sekci „Seznam přiložených souborů“.

ZÁVĚR

Tato práce se zabývá vývojem mobilních aplikací a jakým způsobem mohou být prospěšné u pacientů s Parkinsonovou nemocí. Popisuje, jak může být pasivní a vzdálené monitorování řeči užitečné nejen pro diagnostické účely, ale také pro průběžné sledování zdravotního stavu pacienta. Dále se věnuje návrhu a vývoji takové aplikace.

V první části popisuje problematiku vad řeči, se kterými se pacienti s PN potýkají. Jedná se zejména o problémy v artikulaci, fonaci a prozódii, které jsou pro tuto nemoc typické. Věnuje se tomu, jak se změny v jejich komunikačních schopnostech projevují. Dále prezentuje existující práce na podobné téma, konkrétněji aplikaci iPrognosis, která se právě touto problematikou zabývá.

Ve druhé kapitole je obecný popis OS Android a jakým způsobem na něm fungují mobilní aplikace. Je detailně popsáno, jakými technologiemi a programovacími jazyky lze mobilní aplikace vyvíjet. Jedna tato technologie je dále vybrána, konkrétněji programovací jazyk Kotlin, a tato volba odůvodněna.

V další části práce je vytvořen návrh aplikace a pro OS Android vysvětlen její vývoj. Je podrobně popsáno, za jakých okolností je možné nahrávat telefonní hovory, aby byla splněna podmínka pasivně získaných dat. Je vysvětleno proč toto řešení nemusí být proveditelné a je proveden nový návrh aplikace.

V poslední části je popsán konkrétní vývoj aplikace, tvorba uživatelského rozhraní, využití principů objektově orientovaného programování a práce s oprávněními. Jsou prezentovány možnosti nahrávání zvuku na OS Android a popsána další funkcionality, jako připojení k internetu nebo implementace lokální databáze. Je vysvětleno úskalí implementace jazyka Python a jeho knihoven na systému, který jej nepodporuje, a jsou navržena řešení této problematiky pro potřeby analýzy řeči. Jedno řešení je zvoleno, konkrétně využití serveru, a jednoduchý server je pro demonstrování funkce implementován.

LITERATURA

- [1] BEITZ, Janice M. Parkinson's disease: a review. *Frontiers in Bioscience-Scholar*, School of Nursing-Camden, Rutgers University, 2014, 6.1: 65–74.
- [2] Douglas A. Rahn, Maggie Chou, Jack J. Jiang, Yu Zhang, Phonatory Impairment in Parkinson's Disease: Evidence from Nonlinear Dynamic Analysis and Perturbation Analysis, *Journal of Voice*, Volume 21, Issue 1, 2007, Pages 64–71, ISSN 0892-1997,
- [3] THEODOROS, Deborah a Lorraine RAMIG. *Communication and Swallowing in Parkinson Disease*. San Diego: Plural Publishing, 2011. ISBN 9781597567039.
- [4] LAUREANO, Moro-Velazquez a Dehak NAJIM. *A Review of the Use of Prosodic Aspects of Speech for the Automatic Detection and Assessment of Parkinson's Disease*. Cambridge, Massachussets: Springer, 2019. ISBN 978-3-030-65654-6.
- [5] WHITFIELD, Jason A. *Sources of Intraspeaker Variation in Parkinsonian Speech Related to Speaking Style*. Cambridge, Massachussets: Springer, 2019. ISBN 978-3-030-65654-6.
- [6] Xu Y, Feeney MP, Surface M, Novak D, Troche MS, Beck JC, Alcalay RN. Attitudes Toward Telehealth Services Among People Living With Parkinson's Disease: A Survey Study. *Mov Disord*. 2022 Jun;37(6):1289-1294. doi: 10.1002/mds.28990. Epub 2022 Mar 26. PMID: 35338664; PMCID: PMC9314606.
- [7] Zhan A, Mohan S, Tarolli C, Schneider RB, Adams JL, Sharma S, Elson MJ, Spear KL, Glidden AM, Little MA, Terzis A, Dorsey ER, Saria S. Using Smartphones and Machine Learning to Quantify Parkinson Disease Severity: The Mobile Parkinson Disease Score. *JAMA Neurol*. 2018 Jul 1;75(7):876-880. doi: 10.1001/jamaneurol.2018.0809. PMID: 29582075; PMCID: PMC5885192.
- [8] Laganas, C.; Iakovakis, D.; Hadjidimitriou, S. K.; Charisis, V.; Dias, S. B.; aj.: Parkinson's Disease Detection Based on Running Speech Data From Phone
- [9] What is a Mobile Application? - Definition from Techopedia. Techopedia: Educating IT Professionals To Make Smarter Decisions [online]. Copyright © 2022 [cit. 29.11.2022]. Dostupné z: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [10] Kompletní průvodce pro vývoj mobilních aplikací (Postup, Ceny & Časté chyby). Pixelfield | Vývojáři a designéři z Prahy [online]. Dostupné z: <https://pixelfield.cz/vyvoj-aplikaci/>
- [11] Chart: Android and iOS Dominate the Smartphone Market | Statista. Statista – The Statistics Portal for Market Data, Market Research and Market Studies [online]. [cit. 27.11.2022]. Dostupné z: <https://www.statista.com/chart/9628/smartphone-platform-market-share-forecast/>
- [12] iOS | Apple Wiki | Fandom. [online] [cit. 05.12.2022]. Dostupné z: <https://apple.fandom.com/wiki/IOS>

- [13] Co je Open source. Jan Štráfelda: průvodce online projektem [online] [cit. 03.12.2022]. Dostupné z: <https://www.strafelda.cz/open-source>
- [14] Application Fundamentals | Android Developers. Android Mobile App Developer Tools – Android Developers [online] [cit. 22.11.2022]. Dostupné z: <https://developer.android.com/guide/components/fundamentals>
- [15] Co je virtuální počítač a jak funguje | Microsoft Azure. [online]. Copyright © Microsoft 2022 [cit. 24.11.2022]. Dostupné z: <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-a-virtual-machine/>
- [16] Princip nejnižších oprávnění. Proč je důležitý a jak ho prosadit? - CIO Business World. CIO Business World [online]. Copyright © 2020 [cit. 25.11.2022]. Dostupné z: <https://www.cio.cz/clanky/princip-nejnizsich-opravneni-proc-je-dulezity-a-jak-ho-prosadit/>
- [17] Android App Bundle frequently asked questions | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/guide/app-bundle/faq>
- [18] [online]. Dostupné z: <https://learn.microsoft.com/en-us/visualstudio/cross-platform/cross-platform-mobile-development-in-visual-studio>
- [19] Výuka Informatiky – Programování – Programovací jazyky. *Výuka Informatiky – Informatika – Informace a informatika - Cesta ke hvězdám ...* [online]. Dostupné z: <http://info.spsnome.cz/Programovani/Zaklady-Programovaci-jazyky>
- [20] <https://blogs.oracle.com/javamagazine/post/java-long-term-support-lts>
- [21] 1.2.4. Dynamické vs. statické typování proměnných. *Gymnázium Matyáše Lercha* [online]. Dostupné z: <http://www.gml.cz/projekty/objektove-programovani/programy/output/ch01s02s04.html>
- [22] JVM | Java Virtual Machine – Javatpoint. Tutorials List – Javatpoint [online]. Copyright © Copyright 2011 [cit. 05.12.2022]. Dostupné z: <https://www.javatpoint.com/jvm-java-virtual-machine>
- [23] What is the difference between standard libraries and third-party packages in Python? - Quora. *Quora – A place to share knowledge and better understand the world* [online]. Dostupné z: <https://www.quora.com/What-is-the-difference-between-standard-libraries-and-third-party-packages-in-Python>
- [24] What is BeeWare?— BeeWare. *Write once. Deploy everywhere.— BeeWare* [online]. Copyright © Russell Keith [cit. 24.05.2023]. Dostupné z: <https://beeware.org/project/overview/>
- [25] Welcome to Kivy — Kivy 2.1.0 documentation. Kivy: Cross-platform Python Framework for GUI apps Development [online]. Copyright © Copyright 2010, The Kivy Authors. [cit. 05.12.2022]. Dostupné z: <https://kivy.org/doc/stable/>
- [26] Home | Jython. *Home | Jython* [online]. Dostupné z: <https://www.jython.org/>
- [27] Configure your build | Android Studio | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/build>

- [28] Chaquopy – Python SDK for Android. [online]. Dostupné z: <https://chaquo.com/chaquopy/>
- [29] Gradle plugin – Chaquopy 14.0. [online]. Dostupné z: <https://chaquo.com/chaquopy/doc/current/android.html>
- [30] Kotlin and Android | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/kotlin>
- [31] Sharing audio input | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/guide/topics/media/sharing-audio-input>
- [32] Use the Phone app to record calls – Phone app Help. Google Help [online]. Copyright ©2022 Google [cit. 05.12.2022]. Dostupné z: <https://support.google.com/phoneapp/answer/9803950?hl=en>
- [33] List of Countries that allow Google Phone’s App Call Recording. YTECHB – Android Updates, Wallpapers, News, GCam & More [online]. Copyright © 2022 [cit. 05.12.2022]. Dostupné z: <https://www.ytechb.com/countries-that-support-google-phones-app-call-recording/>
- [34] Developer Program Policy – Play Console Help. Google Help [online]. Copyright ©2022 Google [cit. 06.12.2022]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/12971372>
- [35] Machine Learning | Platform | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/mlhttps://www.tensorflow.org/lite>
- [36] TensorFlow Lite | ML for Mobile and Edge Devices. *TensorFlow* [online]. Dostupné z: <https://www.tensorflow.org/lite>
- [37] Simple audio recognition: Recognizing keywords | TensorFlow Core. *TensorFlow* [online]. Dostupné z: https://www.tensorflow.org/tutorials/audio/simple_audio
- [38] GitHub - praat/praat: Praat: Doing Phonetics By Computer. *GitHub: Let’s build from here · GitHub* [online]. Copyright © 2023 GitHub, Inc. [cit. 24.05.2023]. Dostupné z: <https://github.com/praat/praat>
- [39] Parselmouth – Praat in Python, the Pythonic way — Parselmouth 0.4.3 documentation. [online]. Copyright © Copyright 2017 [cit. 24.05.2023]. Dostupné z: <https://parselmouth.readthedocs.io/en/stable/>
- [40] Meet Android Studio | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/studio/intro>
- [41] MediaRecorder.OutputFormat | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/reference/android/media/MediaRecorder.OutputFormat>

- [42] AudioRecord | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/reference/android/media/AudioRecord>
- [43] GitHub – squti/Android-Wave-Recorder: A powerful and efficient library to record WAVE form audio files (WAV) in Android. *GitHub: Let's build from here · GitHub* [online]. Copyright © 2023 GitHub, Inc. [cit. 24.05.2023]. Dostupné z: <https://github.com/squti/Android-Wave-Recorder>
- [44] Save data in a local database using Room | Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/training/data-storage/room>
- [45] App Architecture: Data Layer - Schedule Task with WorkManager - Android Developers. *Android Mobile App Developer Tools – Android Developers* [online]. Dostupné z: <https://developer.android.com/topic/libraries/architecture/workmanager>
- [46] Gradle plugin - Chaquopy 14.0. [online]. Dostupné z: <https://chaquo.com/chaquopy/doc/current/android.html#android-requirements>
- [47] Failed to install praat-parselmouth · Issue #508 · chaquo/chaquopy · GitHub. *GitHub: Let's build from here · GitHub* [online]. Copyright © 2023 GitHub, Inc. [cit. 24.05.2023]. Dostupné z: <https://github.com/chaquo/chaquopy/issues/508>
- [48] Supported libraries and packages /pypi-7.0. Dostupné z: <https://chaquo.com/pypi-7.0/>
- [49] Web service — Parselmouth 0.4.3 documentation. [online]. Copyright © Copyright 2017 [cit. 24.05.2023]. Dostupné z: https://parselmouth.readthedocs.io/en/stable/examples/web_service.html

SEZNAM PŘILOŽENÝCH SOUBORŮ

A.1 Manifest

V této složce je možné nalézt soubor „AndroidManifest.xml“.

A.2 Java

Všechn kód napsán v jazyce Kotlin je ve složce „java“. Pojmenování složky následuje názvy souborů a složek podle IDE Android Studio.

A.3 Drawable

V příloze ve složce „drawable“ lze nalézt soubory k ikonám, tlačítkům či obrázkům použitým při tvorbě aplikace.

A.4 Layout

V příloze ve složce „layout“ je možné najít soubory uživatelského rozhraní.

A.5 Gradle

V příloze ve složce „gradle“ je možné nalézt soubory softwaru Gradle pro automatizaci sestavení aplikace.

A.6 Server Flask

Ve složce „flask“ jsou uloženy soubory pro implementaci serveru.