VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

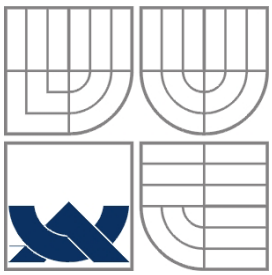
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE ŘEČNÍKA NA MOBILNÍM TELEFONU

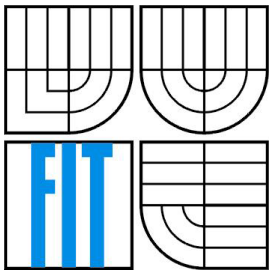
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE VOJTĚCH KALLAB
AUTHOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE ŘEČNÍKA NA MOBILNÍM TELEFONU

SPEAKER IDENTIFICATION FOR CELL PHONES

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

VOJTĚCH KALLAB

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR SCHWARZ, Ph.D.

BRNO 2013

Abstrakt

Tato bakalářská práce je rozdělena do devíti kapitol, které pojednávají o vývoji aplikace „IDENTIFIKACE ŘEČNÍKA NA MOBILNÍM TELEFONU“ pro operační systém Android. V úvodu práce je tento moderní operační systém, vzbuzující zájem široké veřejnosti, krátce představen. Jedna z částí práce je věnována vývojovému prostředí Eclipse. Hlavním bodem této práce je popis implementace aplikace „Speaker ID Watchdog“. Závěrečná část popisuje testování aplikace, marketingovou strategii a službu Google Play. Pomocí této služby bude nově vytvořená aplikace distribuována uživatelům.

Klíčová slova

Android, Elipse, BSAPI, vývoj aplikace, Gogole Play

Abstract

This bachelor thesis is devided into nine chapters which are about the development of an application based on the operating system Android. The introduction of this bachelor thesis is a presentation of this interesting operating system. One part of this thesis is about the Eclipse enviroment. One of the main points of this thesis describes an implementation of the application called „Speaker ID Watchdog“. The final phase is about testing the application, marketing strategy and about the Google Play service. This application is ment to be distribute to users by this service.

Keywords

Android, Elipse, BSAPI, applicationn development, Gogole Play

Citace

Kallab Vojtěch: Identifikace řečníka na mobilním telefonu. Brno, 2013, bakalářská práce, FIT VUT v Brně.

Identifikace řečníka na mobilním telefonu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Schwarze, Ph.D.
Další informace mi poskytli Ing. Milan Schwarz a Bc. Jiří Nytra.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Vojtěch Kallab
15.05.2013

Poděkování

Tímto bych rád poděkoval vedoucímu mé bakalářské práce Ing. Petru Schwarzovi Ph.D., za odborné vedení, náměty a pomoc při vypracování bakalářské práce.

Obsah

Úvod

Zadání této bakalářské práce jsem si vybral především proto, že se jedná o vývoj pro platformu Android. Nikdy před tím jsem neměl žádné zkušenosti s vývojem pro mobilní operační systém. Systém Android patří dnes k nejpopulárnějším operačním systémům vůbec. Chtěl jsem více proniknout do detailů tohoto systému a téma této bakalářské práce mi tuto možnost nabízí. Stejně jako systém Android pro mě bylo novinkou také seznámení s knihovnou BSAPI, vyvíjenou na Fakultě informačních technologií v Brně. Hlasová biometrie má své kouzlo zvláště v dnešní době zjednodušování. Není potřeba znát složitou kombinaci čísel, ale stačí jen mluvit a systém sám rozpozná, o jakého uživatele se jedná. Tato bakalářská práce je podkladem a teoretickým východiskem pro vyvíjenou aplikaci.

Cílem této bakalářské práce je použít dodanou knihovnu, sloužící k analýze hlasu a prezentovat tyto možnosti uživatelům přímo na jejich mobilních zařízeních. V práci se zaměříme na systém Android jako takový, seznámíme se s knihovnou použitou pro hlasovou analýzu a vývojovým prostředím Eclipse. Podrobně se budeme věnovat jednotlivým částem vývoje aplikace a představíme si, jak aplikace funguje. V dalších kapitolách je pozornost obrácena k zabezpečení aplikace a jejímu testování. V poslední řadě si představíme službu Google Play, která byla zvolena pro distribuci finální verze aplikace.

1 Android

1.1 Historie

Historie operačního systému Android, jak ji uvádí Marvan (2011) [1] začala již v roce 2003. Společnost byla založena čtyřmi zakadateli což byl Andy Rubin, Rich Miner, Nick Sears a Chris White. Cílem zakladatelů bylo přinést na trh nový pohled na mobilní zařízení. Tato společnost viděla potenciál v chytrých mobilních telefonem, které přinesou zcela revoluční pohled na využívání mobilních zařízení. Vlastnosti nového produktu, systému Android, měly splnit očekávání i náročných uživatelů. Společnost Android ve svých začátcích nepůsobila nikterak odlišně od konkurenčních vývojářských společností. Převrat přišel až v roce 2005, kdy společnost Android Inc. Byla koupena obrovskou společností Google. Stěžejní lidé pro vývoj nového systému na svých postech zůstali. Andy Rubin donedávna zastával pozici vice prezidenta mobilní divize Googlu. Společnost Google přinesla do vývoje projektu nový rozměr. Tempo vývoje systému Android se radikálně zrychlilo a čas od času bylo slyšet zprávu o připravovaném vstupu společnosti Google na trh mobilních telefonů. Společnost Google získala velkou spoustu patentů v oblasti mobilní komunikace a tím jen utvrzovala uživatele v naději na připravovaný průlom v oblastni mobilních zařízení.

Roku 2007 bylo společností Google vytvořeno konsorcium s názvem: „Open Handset Alliance“. Toto konsorcium představilo nový operační systém (dále OS) pro mobilní zařízení, Android. Logo tohoto systému Android je malý zelený robot. Hned od vydání prvních verzí se stal obrovským symbolem celého nově nastupujícího projektu Android. (viz. Obrázek č.1)

Obrázek č.1: Logo Android [24]

Společnost Google založila toto konsorcium společně s celou řadou dalších gigantických společností, stále má však hlavní slovo při vývoji tohoto operačního systému. Mezi další partnery patří společnosti jako je NVIDIA, Samsung (v roce 2012 Q3 pokrývá 46% trhu mobilních telefonů se systémem Android), HTC (držící se stejnou dobou na druhém místě s aktuálním propadem na 16%), LG, Motorola a Intel. Za zástupce mobilních operátorů můžeme zmínit například T-Mobile a

Telefonica. Cílem tohoto konsorcia bylo vytvoření nově představeného mobilního operačního systému na otevřených standardech.

Chytrý mobilní telefon obsahující systém Android ve verzi 1.0 se dostal na trh v roce 2008. Ohlas na nově představený systém byl pozitivní. Společnost HTC byla první, která systém Android nasadila do prodeje na svých telefonech prodávaných pod názvem HTC Dream neboli T-Mobile G1. Takto byla zahájena éra prozatím nejúspěšnějšího mobilního opračního systému. Momentálně jsou na prodej zařízení obsahující Android již ve verzi 4.2.1 a od vývoje tohoto systému se rozhodně neustupuje. V dohledné době má přijít na trh operační systém Android ve verzi 5.0 KeyLime.

Tento systém je založen na linuxovém jádře. Jádro systému je optimalizováno a konfigurováno přesně na míru procesoru moblních zařízení, jedná se především o procesory ARM. I když je jádro samo o sobě přizpůsobené pro potřeby mobilních zařízení, aplikace pro systém Android nekomunikují přímo s ním, ale s Android Application Programming Interface (dále jako API). Přes toto rozhraní komunikují vyvíjené aplikace k potřebným funkcím mobilního zařízení, jako je ovládání displeje, senzorů a možnostem operačního systému.

Samotný běh aplikace obstarává Dalvik Virtual Machine (dále jako VM). Jedná se o virtuální stroj podobný Java VM. Díky této podobnosti se společnosti Google podařilo spojit oblíbený programovací jazyk Java s vlasním operačním systémem Android. Vývoj pro platrofmu Android je programování v Javě, ale po následné kompilaci aplikace je zapotřebí VM určený pro mobilní aplikace.

Systém Android dnes patří mezi nejpopulárnější a zároveň nejrozšířenější mobilní platformu na světě. Díky velkému pokroku ve vývoji hardwarového vybavení mobilních telefonů, kde se největším lákadlem pro spousty uživatelů jistě stala dotyková obrazovka, došlo k téměř raketovému startu této nové platformy. Platforma Android pomocí svého virtuálního stroje Dalvik dokáže odstínit hardwarové odlišnosti mobilních telefonů či tabletů, a značně tak ulehčí práci vývojářům ladícím aplikace pro jednotlivá zařízení.

Jednou z mnoha výhod tohoto systému je existence velkého množství grafických nádstaveb. Právě tyto nádstavby systému umožňují odlišení konkurenčních značek i v případě stejného operačního systému. Jako příklad je možné uvést HTC Sence nebo Samsung TouchWiz. Díky této vymoženosti si může kdokoliv vybrat mobilní telefon podle svých vlastních preferencí, ať se jedná o cenu, značku nebo některou z hardwarových specifikací, a budeme mít skoro vždy stejné nebo alespoň podobné softwarové možnosti v nabízených aplikacích.

1.2 Smart aplikace

Největším úspěchem tohoto nového systému jsou ale zcela jistě dostupné aplikace skrze službu Google Play. Komfort procházení aplikací, kterých je v těchto dnech na Google Play již více jak 700 000, je zajištěn vestavěnými klienty v mobilních zařízeních, nebo výběrem aplikace přes

webovou verzi služby Google Play. Neznamená to však, že by bylo mobilní zařízení zcela odkázáno na aplikace distribuivané skrze tuto službu. Není problém instalovat aplikace ve formátu APK z libovolných dalších zrojů. Telefon se tak stává chytrým právě díky těmto aplikacím.

1.3 Multitasking

Jednou z klíčových vlastností popisovaného systému Android je plná podpora multitaskingu. V tomto systému je každé z aplikací přiřazen pro běh virtuální stroj, oddělený samostatným procesem. Jedná se tedy, na rozdíl od jiných operačních systémů pro mobilní zařízení, o plnohodnotný multitasking.

Pokud je aplikace určena pro běh na pozadí, ať již uživatelem nebo systémem, další řízení životního cyklu aplikace je pod správou operačního systému. Každá z aplikací se může nacházet v různých stavech. Konkrétně se jedná o tyto tři stavy: běžící, uspaná, zastavená. Mezi těmito stavy je přepínáno podle požadavků uživatele nebo podle stavu operační paměti zařízení. V případě, že je operační paměť zaplněná, například při spuštění další nové aplikace, systém vybere sám aplikaci, kterou ukončí. Správou procesů uživatel není ve většině případů obtěžován. Pokud má uživatel zájem dozvědět se více informací o probíhajících aktivitách, od verze systému Android 2.3 je k dispozici vestavěný nástroj na správu těchto probíhajících procesů. Pro náročnější uživatele existuje ale celá řada aplikací na správu procesů dostupných přes službu Google Play.

1.4 Telefon, mediální centrum, PC

Chytré telefony, neboli smartphony spojují dříve neslučitelná zařízení jako je osobní počítač (videopřehrávač, prohlížeč internetu), mobilní telefon nebo herní konzoli.

Hardwarové možnosti jsou již v mobilních zařízeních dostačující a záleží jen na aplikacích, které bude uživatel potřebovat. GPS navigace v mobilním telefonu dnes již nikoho neuchvátí a je považována spíše za samozdřejmost. Platforma Android nezůstává vyhraněná jen pro mobilní telefony, vždyť původní záměr vývoje tohoto systému byl operační systém určený pro fotoaparáty, ale rozšiřuje své zastoupení i v tabletech a v poslední době i v mediálních centrech. Vize do budoucna je stále více směřována k jednomu či více televizorům s připojitelným mediálním centrem založeným na platformě Android, nebo přímo s mobilním telefonem. V tomto případě bude záležet na aplikacích vyvíjených a laděných přímo k tomuto účelu.

1.5 Služby Google

Neopomenutelnou součástí systému Android je také úzká provázanost se službami Google [2]. K tomu, aby mohlo být využíváno všech možností chytrého telefonu se systémem Android, je potřeba

mít založený účet u společnosti Google. Tento účet je následně použit pro přístup do aplikace Google Play, dříve známý jako Android Market. Propojení je zajištěno i s dalšími službami jako je Gmail nebo Google Talk. Tento účet je používán například také pro synchronizaci kontaktů a dalších nastavení telefonu. Systém Android a velké množství aplikací fungují optimálně, pokud je v zařízení k dispozici trvalý přístup na internet.

2 Phonexia Speaker ID BSAPI

2.1 Hlasová analýza

Vytváření hlasových otisků uživatelů je výpočetně optimalizované právě i pro mobilní telefony, které mají být především nosiči tohoto bezpečnostního mechanismu. Je použita upravená metoda hlasové analýzy i-vectors. Tuto metodu popisují O. Glembek, L. Burget, P. Matějka, M. Karafiát a P. Kenny, (2011) v článku „Simplification and optimalization of i-vector extraction“ [3]. Jedná se o metodu analýzy hlasu založenou na principu i-vectors s tím rozdílem, že tato metoda optimalizuje analýzu i pro běh na výkonově slabších zařízeních jako jsou například mobilní telefony. Původní metoda i-vectors je pro tyto zařízení nepřijatelná jak z pohledu množství zpracovávaných dat, tak z pohledu doby trvání analýzy. Tato upravená verze hlasové analýzy pracuje i s řádově menšími hlasovými nahrávkami a dosahuje výsledků v daleko kratším čase. Firma Phonexia s.r.o., která se zabývá vývojem společně s řečovou skupinou na Fakultě informačních technologií VUT v Brně o produktu ve své produktové prezentaci uvádí, že má tyto vlastnosti: Velikost vytvořeného hlasového otisku je pouhých 624 bajtů. Pro vytváření hlasových otisků je poměr 50 sekund zvukového záznamu roven výpočetnímu výkonu 1 sekundy strojového času CPU na jednom jádře. Pro porovnávání hlasových otisků se dostáváme na neuvěřitelných 1 000 000 porovnání dvou hlasových stop za 1 sekundu strojového času CPU na jednom výpočetním jádře.

Společnost Phonexia s.r.o. se zabývá zvukovou biometrií. Jedná se o metodu, která je založená na konkrétních charakteristikách zvuku analyzovaného objektu. Cíle společnosti jsou přivést hlasovou analýzu k takové dokonalosti, aby bylo možné použít svůj hlas místo hesel. Dalším přínosem by bylo případně zablokování přístroje pro volání osobou neověřitelnou. Důvěryhodnost komunikujících stran se stává v dnešním světě často kladenou otázkou, a proto se na tuto otázku snaží odpovědět i společnost Phonexia s.r.o. svou hlasovou analýzou lidské řeči. Tyto možnosti existují, pokud se jedná o státní orgány nebo důležité bezpečnostní složky, ale společnost Phonexia s.r.o. chce tyto možnosti nabídnout všem uživatelům, kteří o danou službu budou mít zájem i pro osobní použití, jako je zabazpečení osobních mobilních telefonů.

2.2 SID-android

Jeden ze zdrojů, který byl k dispozici této práci, byl vývojářský balíček firmy Phonexia s.r.o.. Tento balíček obsahoval knihovnu BSAPI s příslušnou adresářovou strukturou a všemi potřebnými soubory. Stejně tak byl k dispozici projekt, importovatelný přímo do vývojového prostředí Eclipse, který měl demonstrovat chování knihovny. Tento projekt nebylo možné dle dodaných manuálů bohužel ani po delším zkoumání zprovoznit, ale zdrojové kódy jasně vypovídaly o tom, jak má aplikace fungovat.

Tento projekt byl při vývoji aplikace „Speaker ID Watchdog“ v mnoha směrech oporou, jak s knihovnou nakládat. Nejzásadněji se informace z tohoto balíčku projevily v převodu zvuku do formátu WAV a při vytváření hlasových stop uživatele.

Dodaná aplikace měla za cíl pouze demonstrovat použití zmíněné knihovny. Uživatelská hodnota byla velice malá, protože přinesla jen identifikaci hlasových stop uživatelů, ale z hlediska reálného používání nebyla nijak marketingově zajímavá. Proto bylo potřeba demonstrovat použití na reálném příkladu, který by ocenili i uživatelé méně zapálení pro systém Android a výpočetní techniku obecně.

3 Vývojové prostředí

Společnost Google Inc. doporučuje a oficiálně podporuje pouze vývojové prostředí Elipse Integrated Development Environment (dále IDE) [4]. Jedná se o open source vývojovou platformu v jazyce Java. Jedná se o je plnohodnotné Java IDE prostředí, které existuje ve více standardních variantách s množstvím doplňků cílených přímo pro dané potřeby vývojáře. Díky těmto doplňkům se prostředí plně přizpůsobí jazyku, ve kterém bude probíhat vývoj aplikace.

3.1 Eclipse

Kterémukoli vývojáři nebo uživateli, který se zajímá o tuto platformu je k dispozici volně ke stažení plugin, tedy rozšíření, nazvaný Android Developer Tools, ve zkratece ADT, který nastaví vývojové prostředí právě pro vývoj dané platrofmy [5].

Systém Android je postavený na principech Javy a proto je k běhu vývojového prostředí nutné si stáhnout Java Development Kit, kterým je znám jako JDK. Další potřebnou součástí vývojového prostředí je Software Development Kit (dále SDK) pro Android, který je zmiňován a popisován níže. Dálším nutným doplňkem vývojového prostředí je Android Virtual Device (dále AVD) Manager.

3.2 Android SDK Manager

Android SDK Manager slouží ke správě SDK platforem a dalších vývojových doplňků.

Jedná se o doplněk vývojového prostředí Eclipse, který umožňuje stahování potřebných nástrojů, jako jsou například ladící nástroje, knihovny a především k aktualizaci a rozšiřování množství nových verzí platforem Androidu, pro které bude vyvíjená aplikace určena. Konfigurace SDK je rozdělena do více kategorií, pro potřebu této práce jsou zmíněny následující:

SDK Tools – nástroje určené pro ladění, testování aplikací a AVD. V této kategorii je zahrnut i emulátor mobilních zařízení a prostředí pro ladění grafického návrhu.

Android SDK platforms – jednotlivé platformy SDK jsou složeny z knihoven, příkladů zdrojových kódů. Nabízí různé druhy emulátorů. Pokud chceme kompilovat aplikaci nebo provést nastavení AVD, musí být stáhnutá alespoň jedna verze platformy Android přes SDK Managera.

USB Driver – pokud se jedná o testování a ladění aplikace na fyzickém zařízení, potřebujeme USB ovladače daného zařízení.

3.3 AVD Manager

AVD Manager slouží ke správě a vytváření virtuálních zařízení pro testování aplikací. Díky tomu je možné testovat aplikace bez fyzického zařízení.

Emulátor operačního zařízení systému Android je obsažen v Android SDK. Pomocí Android SDK a AVD Manageru je možné konfigurovat například:

- volbu síťového připojení
- SD karty
- spouštět jednotilivá virtuální zařízení
- další možnosti

Většina aplikací se chová v emulátoru stejně jako na fyzickém zažízení. Existují ovšem vyjímečné situace, které se virtualizovat nedají, nebo dají jen velmi obtížně, jako je zaznamenávání video a audio vstupu, úroveň nabití baterie, bluetooth a další.

Pro realizaci zadaného projektu vyvstal problém například s klíčovým přijímáním hovorů.

4 Vývoj aplikace Speaker ID Watchdog

4.1 Vytvoření nového projektu

Pro účely této bakalářkské práce byl vytvořen nový projekt Phonexia, Speaker ID Watchdog. Minimální verze SDK je nastavena na API úroveň 10 Android 2.3.3 s názvem Gingerbread. Cílová verze SDK je nastavena na API úroveň 15 Android 4.0.3, který nese název IceCreamSandwich.

Vytvořená základní adresářová struktura nově vytvořeného projektu se skládá především z adresáře zdrojových kódů aplikace `src` a adresáře zdojů `res`. Nejdůležitější podadresáře jsou především adresář `res/layout` pro grafické rozvržení aktivit a adresář `res/values` s pomocnými definičními soubory. Klíčovým a jedním z nejvýznamnějších položek je soubor v kořenovém adresáři. Tím je AndroidManifest.xml. Během dalšího vývoje aplikace jsou vytvořeny ještě další adresáře. Konkrténě se jedná o adresář `assets` pro další přídavné soubory přibalené k projektu a adresáře `libs` a `jar` pro další pomocné knihovny aplikace.

Soubor **AndroidManifest.xml,** jak uvádí příslušný článek [6], rámcově popisuje celou vyvíjenou aplikaci. Je zde definováno použití ikon, nastavení potřebného oprávnění pro běh celé aplikace a základní deklarace všech používaných komponent v projektu.

Při exportu aplikace do výsledného APK souboru je soubor AndroiManifest.xml automaticky přidán do balíčku spolu se zkompilovanou aplikací. Jedná se například o údaje, které jsou později přikládány k vytvořené aplikaci.

Zdrojový kód AndroidManifest má následující stukturu:

- Pojmenování balíčku – jedinečný identifikátor aplikace.

  ```
  package="com.phonexia.SpeakerIDWatchdog"
  ```

- Deklarace minimální API úrovně, kterou aplikace vyžaduje.

  ```
  android:minSdkVersion="10"
  android:targetSdkVersion="15"
  ```

- Popis všech komponent připravované aplikace - Deklaruje všechny aktivity, poskytovatele a intenty používané v celém projektu.

  ```
  <activity
          android:name="com.phonexia.SpeakerIDWatchdog.MainActivity"
          android:label="@string/title_activity_main" >
          <intent-filter>
              <action android:name="android.intent.action.MAIN" />
              <category android:name="android.intent.category.LAUNCHER" />
          </intent-filter>
  </activity>
  ```

- Definice oprávnění, které daná aplikace potřebuje ke svému chodu a interakci s jinými aplikacemi. Tyto informace jsou zobrazeny všem uživatelům před stažením nebo instalací aplikace a je nutné s tímto nastavením aplikace souhlasit.

  Takto vypadá nastavení oprávnění pro aplikaci Speaker ID Watchdog:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

4.2 Programování aplikace

4.2.1 MainActivity

Mezi základní komponenty patří tzv. Aktivity. Každá z nich reprezentuje zobrazení dat na obrazovce zařízení. Jedna aplikace, jako je tomu u aplikace „Speaker ID Watchdog“, je obvykle tvořena z většího množství aktivit, a mezi těmito aktivitami se pak může buď uživatel, nebo i aplikace sama přepínat. Na přepínání aktivit slouží „intenty“, které nejen že zajišťují přepínání, ale umožňují aktivitám předávat si potřebné parametry mezi sebou.

Pro správu aktivit v systému Android slouží Activity Manager, který pracuje se zásobníkem, ve kterém jsou všechny potřebné informace o právě prohíhajících aktivitách. Aktuálně zobrazená aktivita na obrazovce zařízení je vždy první v zásobníku a je zároveň jediná připravená na interakci s uživatelem.

Hlavní třída aplikace Speaker ID Watchdog, *MainActivity* dědí ze třídy *android.app.Activity*. V metodě této třídy *onCreate()* propojíme vzhled obrazovky s naší aktivitou pomocí metody *setContentView()*. Kdybychom nevyvíjeli na platformě Android, museli bychom hledat všechny soubory dle jejich umístění, teď například v *res/layout*. Android ale namísto toho automaticky vytváří odkazy a umožňuje tak vývojáři přistupovat k potřebným zdrojům přes generovaný soubor *R.java*, v našem případě se jedná konkrétně o *R.layout.activity_main*.

Při spuštění aplikace probíhá nahrání systémových knihoven *bsapi* a *bsapiwrapper* a aktivace používané knihovny *BSAPI* pro práci s hlasovými soubory za pomocí licenčního klíče. Dále je kontrolována dostupnost externích knihoven na SD kartě. Pokud není dohledatelný příslušný adresář aplikace, dojde k vytvoření nového vlákna a spuštění kopírování souborů z aplikačního adresáře *assets/* do paměti mobilního zařízení.

V případě, že se nejedná o projekt podporující APK Expansion Fines, jsou zdroje kopírovány do cílového adresáře pomocí zanořovacího se kopírování. Pokud se ve zdrojových souborech najde soubor určený ke kopírování na SD kartu, provede se kopírování [7]. Pokud je nalezen adresář, proběhne zanoření o úroveň níže. Na této úrovni jsou procházeny všechny soubory a opět probíhá kopírování při splnění základní podmínky, že se jedná o soubor. Pokud je použito rozšíření aplikace za pomoci APK Expansion Files, proběhne kontrola, případné kopírování souborů z připraveného balíčku.

Po úvodní inicializaci aplikace je k dispozici základní menu. Jako první položka menu je zobrazen stavu aplikace: „Watchdog is running“ nebo „Watchdog is not running“. Pokud aplikace běží, je aktivován příslušný *BroadcastReceiver*. Cílem bylo vytvořit uživatelsky příjemnou a jednoduchou aplikaci, proto je menu děleno do pěti základních kategorií. (viz. Obrázek č.2)

- „Start/Stop“

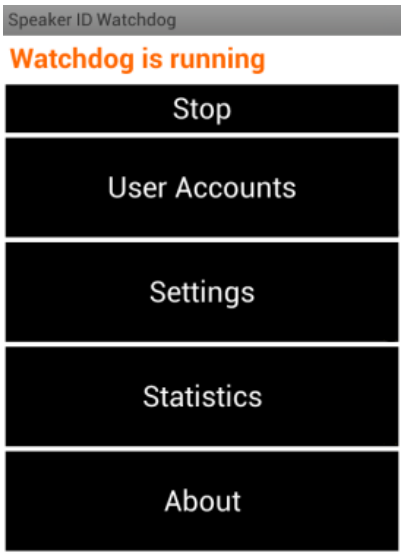
  Zapínání nebo vypínání aplikace pro evidenci telefonních hovorů.
- „Users Accounts“

  Správa uživatelských účtů.
- „Settings“

  Bezpečnostní nastavení aplikace v interakci s telefoním hovorem.
- „Statistics“

  Výpis statistik používání mobilního telefonu.
- „About“

  Údaje o aplikaci, kontakt na firmu Phonexia s.r.o, kontakt na vývojáře.



Obrázek č.2: Menu aplikace

4.2.2 Správa uživatelů – User Accounts

Správa uživatelů je aktivita, která se stará o evidenci uživatelů využívajích danou aplikaci „Speaker ID Watchdog“. Jsou zde implementovány základní nástroje pro práci s uživatelskými účty, jako je například jejich vytváření, editace či rušení.

Pro správu uživatelů byla vytvořena SQLite databáze. SQLite databáze je v Androidu součástí runtime prostředí, takže vlastní databázi může vytvářet každá aplikace. Jak uvádí M. L. Murphy v knize Průvodce programováním mobilních aplikací Android 2 (2011) [8], SQLite používá standardní rozhraní SQL a proto se práce s vytvářenou databází obejde bez dalšího studia nových rozhraní.

V aplikaci „Speaker ID Watchdog“ je vytvořená třída *AccountsDatabase*. Tato třída implementuje třídu *DatabaseHelper*, děděnou ze třídy *SQLiteOpenHelper*. Při volání metody *onCreate()* je vytvořena databáze nesoucí název *data* s tabulkou *users*.

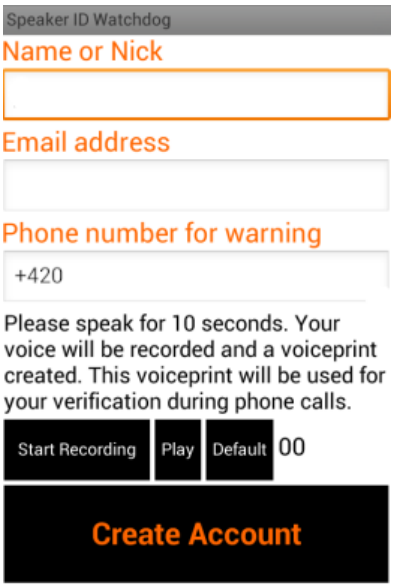
Samozřejmostí je ve třídě *AccountDatabase* implementace metody *open()* na vytvoření nebo otevření databáze a také metoda *close()* pro uzavření databáze k dalšímu použití. Další implementovaná metoda *createUser()* má na starosti vytváření nových uživatelů. U každého uživatele je evidováno jméno, email, telefonní číslo a jedinečný identifikátor *rowId*. Metoda *deleteUser()* maže uživatele z databáze právě na základě *rowId*. Další implementované metody řídí zobrazování seznamu všech uživatelů, či jejich detailních informací.

Pří otevření položky z hlavního menu aplikace *User Accounts* je zavolána aktivita *DisplayAccountsActivity*, která zobrazuje obsah databáze *data*. Při prvním otevření databáze je databáze prázdná a uživatel je informován textem na obrazovce „No Users Yet“. Pro založení profilu nového uživatele a zaznamenání těchto údajů do databáze, je v metodě *onMenuItemSelected()*, pod volbou „New User“, nebo také při zmáčknutí tlačítka „New User“, volán intent pro vytvoření aktivity *DisplayEditUser* dědící ze třídy *android.app.Activity*. Pro editaci údajů uživatele je implementována metoda *onListItemClick()*, v níž je volán opět intent pro vytvoření aktivity *DisplayEditUser* a jsou zobrazena data pro editaci příslušného uživatele. V případě že se jedná o editaci profilu uživatele jsou uložena nová data, a hlasová stopa uživatele v případě potřeby přejmenována. V metodě *onContextItemSelected()* je nabídnuta možnost pro odstranění uživatelského účtu. Tato možnost volá metodu *deleteUser()*, která zajistí, že je uživatel odstraněn z databáze.

4.2.2.1 Vkládání nového uživatele

Vytvoření profilu nového uživatele probíhá pomocí třídy *DisplayEditUser*. Zde je možné vkládat údaje o profilu, který chceme pro nového uživatele vytvořit. Při volání metody *onCreate()* je pomocí *setContentView()* nastaveno příslušné grafické rozhraní. Při vytvoření aktivity proběhne inicializace cest k potřebným knihovnám pro vytváření hlasových stop uživatelů. Připraví se vytvořená databáze

data. Všechna textová pole (viz. Obrázek č.3) jsou prázdná, až na pole s telefonním číslem, které je předdefinovano specifickou telefonní předvolbou České republiky, což je *„+420“*.



Obrázek č.3: Formulář uživatele

4.2.2.2 Nahrávání zvukové stopy uživatele

Nahrávání nové, případně aktualizace stávající zvukové stopy je zahájeno po stisknutí tlačítka „Start Recording“. Samotné nahrávání je realizováno pomocí objektu rekordér typu *AudioRecord*. *AudioRecord* je třída připravená pro nahrávání audio vstupu. Jako zdroj nahrávání je definován mikrofon v rozsahu 8000Hz, mono kanál a kódován na 16B s minimální velikostí bufferu.

Pro nahrávání je vytvořeno vlákno, které během nahrávání zvuk ve formátu 3GG převádí do formátu WAV. Nahrávání lze ukončit nejdříve po deseti sekundách, kdy je hlasová stopa považována za ověřitelnou. Nahrávání je ukončeno po stisknutí tlačítka „Stop“. Uživatel má možnost si nahrávku svého hlasu přehrát a ověřit si tak kvalitu pořízené nahrávky. Po stisknutí tlačítka „Play“ se vytvoří nový objekt, kterým je přehrávač ze třídy *MediaPlayer*. Zdroj přehrávání je nahraný zvukový soubor. Jakmile je přehrávač připraven, spustí se přehrávání zvuku.

4.2.2.3 Nastavení výchozího uživatele pro zabezpečení telefonu

Toto nastavení uživatelského účtu slouží pro evidování uživatele jako výchozí kontaktní osobu v případě bezpečnostního incidentu. V praxi tato možnost bude představovat použití uvedeného telefonního čísla a e-mailu, jako cíle pro směrování výstražných zpráv dle dalšího nastavení.

K tomuto nastavení výchozího uživatele je uživatel automaticky vyzván při potvrzení informací a vytváření svého hlasového otisku. Případně je možné nastavit jako výchozího kteréhokoliv již existujícího uživatele v databázi pomocí tlačítka „Default“. Tato data jsou následně uložena v konfiguračním souboru *myPrefs*. Tento konfigurační soubor je vytvořen ze třídy *SharedPreferences* s parametrem *„0“* kvůli dostupnosti i ze služeb běžících na pozadí.

4.2.2.4 Vytvoření hlasové stopy uživatele

Při vytváření záznamu o novém uživateli je vytvářena i nová hlasová stopa. Po zadání potřebných informací, což v tomto případě obnáší jen jméno uživatele, je možné potvrdit vytvoření nového záznamu stisknutím tlačítka „Create Account“. K tomuto účelu je vytvořeno nové vlákno. Během vytváření hlasové stopy je uživatel na proces vznikajícího hlasového otisku upozorněn.

Pro vytvoření hlasové stopy uživatele je z příslušného datového adresáře načtena zvuková stopa uživatele ve formátu WAV. Pro převod zvukové stopy ve formátu WAV do formátu VP je volána metoda *createVoiceprint()*. V této metodě je vytvořen nový objekt ze třídy *VoicePrintExtractor* s potřebnými parametry dle konfiguračního souboru *„extrakt_android.bs“*.

Metodě *processFile()* třídy *VoicePrintExtractor* jsou předány dva parametry. Jako první je název zvukové nahrávky uživatele a jako druhý je název vytvářené hlasové stopy. Při úspěšném vytvoření hlasové stopy je stopa uložena na zadané cestě.

Při splnění všech stanovených podmínek proběhne vytvoření nebo aktualizace účtu příslušného uživatele.

4.2.3 Nastavení úrovně zabezpečení - Settings

Nastavení úrovně zabezpečení je uživatelsky konfigurovatelná nabídka. Toto zabezpečení je aplikováno při reakci na neověřitelné použití mobilního zařízení. Uživatel si tímto způsobem nastaví, do jaké míry chce být informován o využívání svého mobilního zařízení.

V sekci nastavení má uživatel na výběr ze dvou typů upozornění. Veškeré provedené změny je potřeba potvrdit tlačítkem „Confirm“. Jinak se změny v nastavení neprojeví. Jako výchozí nastavení není použito žádné z těchto upozornění, aby nebyl uživatel zbytečně obtěžován příchozími e-maily či útratou za nevědomky odeslané SMS zprávy. (viz. Obrázek č.4)

- Prvním typem je zasílání e-mailu. Při nastavení této možnosti dochází k informování uživatele pomocí e-mailu. Viz. 4.2.3.1.
- Druhou možností zabezpečení je aktivace zasílání SMS zpráv. Viz 4.2.3.2

4.2.3.1 Uporozornění e-mailem

Android nabízí celou škálu možností jak odeslat e-mail. Problém však nastává v interakci s uživatelem. Aplikace „Speaker ID Watchdog“ obsahuje služby běžící na pozadí a proto není možné využít žádnou z nativně dostupných možností jak odeslat e-mail. Bylo by nežádoucí, aby uživatel, kterého se nepodaří ověřit vůči databázi, musel sám zasílat e-mail o neoprávněném použití telefonu, proto bylo použito alternativně volně dostupné JavaMail API navržené pro Android, díky kterému je možné zaslat e-mail bez potvrzení neověřitelným uživatelem.

V první řadě je nutné si stáhnout potřebné knihovny: additional.jar, mail.jar, activation.jar. Tyto knihovny jsou veřejně dostupné na stránkách Google Developers [9].

Podle vzoru [10], na který se odvolává i tato oficiální stránka, byla vytvořena třída Mail. Zde je základní šablona e-mailu: od koho je email posílán, komu je určen, jaký je předmět e-mailu a nakonec i samotné tělo zprávy. Definování smtp serveru i s komunikačním portem, jméno a heslo uživatele, přes které bude realizováno zasílání e-mailů si uživatel nastaví právě v této záložce „Settings“. Pokud je uživatelem nastavena možnost zasílání upozornění e-mailem, je potřeba tlačítkem „E-mail settings“ nastavit SMTP server, port a přihlašovací údaje k účtu ze kterého bude uživateli zaslán e-mail. Například pro službu Gmail se jedná konkrétně o smtp.gmail.com a port 465.

K vytvořené zprávě je připojována příloha díky metodě *addAttachment()* [11]. Jako příloha je použita hlasová stopa neověřitelného uživatele pro případnou dodatečnou hlasovou analýzu. Tato služba běží na pozadí díky komponentě „Service“ a proto není zaručené, že e-mail dojde výchozímu uživateli ihned, ale co nejdříve co bude realizace možná.

4.2.3.2 Upozorňování pomocí SMS

Zasílání SMS funguje díky funkcím mobilního telefonu. V této aplikaci je vytvořen objekt ze třídy *SmsManager*, který se stará o zasílání SMS zpráv. Tomuto objektu jsou nastaveny parametry potřebné pro žádané upozornění, což je telefonní číslo výchozího uživatele s textem zprávy „Your phone was used by unauthorized person!“ s případným doplněním o GPS souřadnice místa, odkud bylo telefonní zařízení použito, dle konfigurace nastavení viz. 4.2.3.3. Na nastavené telefonní číslo přijde SMS zpráva obratem.

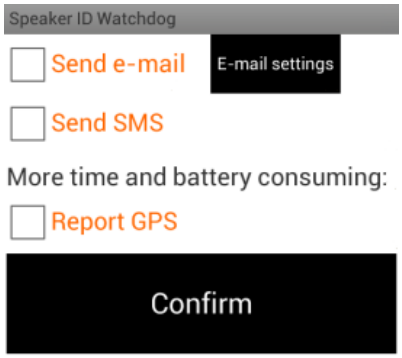
4.2.3.3 Využití údajů GPS

Pro obě varianty, tedy zasílání e-mailu i SMS zpráv je možné přidat výsledek zjišťování polohy uživatele pomocí GPS senzoru.

Tato geolokace je ale poměrně náročná. Vývojář nemůže skrytě zapnout GPS lokátor, kvůli ochraně osobních údajů uživatele, aby nebyli uživatelé sledováni a proto je uživatel vyzván k zapnutí senzoru. Pokud uživatel sám GPS modul nezapne, není možné získat informaci o jeho poloze. Pokud je zaškrtnutá volba používání GPS souřadnic a je aktivní GPS senzor v zařízení, probíhá zjišťování polohy automaticky [12]. V budovách a příliš uzavřených místech ale nemusí lokalizace vůbec proběhnout, protože GPS signál nemusí být dostatečně silný. Toto bezpečnostní nastavení rozšiřuje informační zprávy o GPS souřadnice. Při zjišťování souřadnic je ale více vytěžováno mobilní zařízení a logickým důsledkem je větší odběr baterie. Informace o zvýšených požadavcích na hardware zařízení je uvedena textem „More time and batterry consuming“ přímo nad volbou zapínání GPS. V případě, že uživatel sám nezapne GPS senzor, je k tomu ještě třikrát po jedné minutě vyzván. Pokud ani po tomto intervalu tak neudělá, je zaslána informace bez GPS souřadnic.

4.2.3.4 Služby běžící na pozadí

Pro implementaci služeb upozorňování uživatele byla použita komponenta „Service“. Jedná se o třídu bez grafického rozhraní. Je to proces, který běží na pozadí a stává se tak pro běžného uživatele neviditelým. Toto je jediný způsob jak reagovat na vzniklá bezpečnostní rizika, aby nebyl uživatel informován, že aplikace jeho neoprávněné počínání monitoruje. Tato komponenta je navržena tak, aby vykonávala úkony trvající delší dobu. V případě aplikace „Speaker ID Watchdog“ se jedná především o časově náročnější odesílání e-mailu. Komponentu je v případě potřeby aktivována metodou *startService()*.



Obrázek č.4: Nastavení

4.2.4 Statistiky - Statistics

V položce statistiky je evidován každý uskutečněný telefonní hovor.

Zapisování stavů a aktivit telefonního zařízení je ukládáno do souboru *Log.txt*. Tento soubor je v systému Android otevřen při vytvoření nového objektu *File* a jako parametr je použita cesta kde již soubor existuje či bude uložen. Pokud soubor neexistuje, je vytvořen při prvním zápisu zcela nový a prázdný dokument. V rámci kopírovaných dat, při inicializaci aplikace, je na patřičných místech v projektu umístěn prázdný logovací soubor, aby nedocházelo k chybám při otevírání prázdného souboru. Tento soubor je vytvořen z důvodu, aby uživatel měl možnost již po spuštění aplikace procházet všechny položky úvodního menu aplikace. Pro ošetření případné chyby je kontrolováno vytvoření logovacího souboru a v případě, že soubor neexistue, uživateli je oznámeno, že doposud nebyla pořízena žádná data.

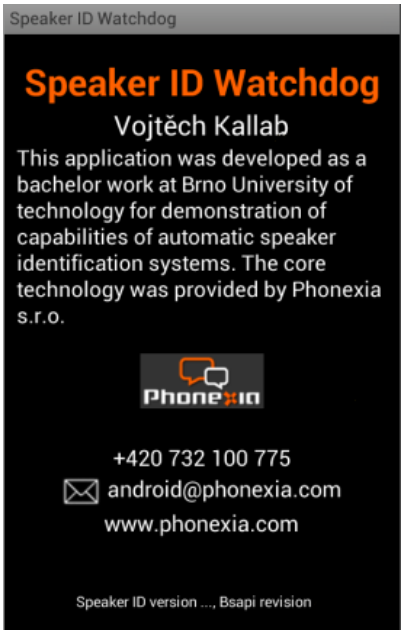
Pro zápis do textového souboru je použit objekt *BufferedWriter* s parametry k logovacímu souboru. První z údajů je název uživatele, který je pod daným jménem evidován v databázi, případně

je využit řetězec „Unknown“, pokud se jedná o neověřeného uživatele, dalšími položkami jsou datum a čas telefonátu, číslo, na které byl telefonní hovor uskutečněn a délka hovoru. Parametry výpisu do logovacího souboru mohou být vývojářem měněny dle zpětného ohlasu uživatelů, jestli jsou to všechna data, co potřebují znát, či ještě chtějí vědět nějaké další doplňující informace. Může se jednat například o ceny hovorů.

Pokud je zapotřebí vymazat údaje o aktivitách telefonního zařízení, je zde připraveno tlačítko „Delete“. Toto tlačítko po stisknutí vybídne uživatele, zda chce opravdu smazat logovací soubor. Logovací soubor je opět vytvořen při prvním použití souboru pro zápis údajů.

4.2.5 O aplikaci - About

Poslední položka hlavního menu je pro uživatele čistě informativní. Obsahuje informační údaje o vyvíjené aplikaci, název, kontakt na firmu Phonexia s.r.o., kontakt na vývojáře a verze aktuálně používané aplikace a knihovny. (viz. Obrázek č.5)



Obrázek č.5: O aplikaci

4.3 Běh aplikace

Pro samotný běh aplikace byla zvolena komponenta *BroadcastReceiver*, která slouží právě k potřebám této aplikace. Tato komponenta naslouchá stavům a oznámením na mobilním zařízení a na jejich základě vykonává další operace. Z pohledu uživatele zde není příliš mnoho odlišností od komponenty „Service“. Žádná z těchto komponent nemá grafické uživatelské rozhraní.

Hlavní význam aplikace je založený na sledování telefonního zařízení. Při zahájení a ukončení telefonního hovoru provádí nahrávní zvuku z mikrofonu mobilního zařízení a následnou analýzu.

Celý tento běh aplikace na pozadí se zakládá na třídě *CallReceiver* děděné ze třídy *BroadcastReceiver*. V metodě *onReceive()*, kterou tato třída musí obsahovat je vytvořen objekt ze třídy *PhoneReceiver*[13] a objekt „manager“ třídy *TelephonyManager*[14]. Díky těmto objektům je možné zjistit, v jakém stavu se v daný okamžik nachází monitorované zařízení a ná základě tohoto stavu provést následné operace.

Vytvořený objekt *manager* metodou *getSystemService()* získává přístup k zařízení „TELEPHONY_SERVICE“ a naslouchá pomocí metody *listen()* stavu a změnám telefonního zařízení „LISTEN_CALL_STATE“. Při prvním spuštění aplikace je potřeba aktivovat komponentu typu *BroadcastReceiver* metodou *setComponentEnableSettings()* a parametrem COMPONENT_ENABLED_STATE_ENABLED.

Pro vyvíjenou aplikaci je potřeba rozlišit tři typy stavů zařízení.

- CALL_STATE_IDLE – stav kdy je telefonní zařízení v klidu, bez žádné aktivity
- CALL_STATE_RINGING – stav, který oznamuje příchod nového hovoru, vyzvánění, případně čekání před přijetím
- CALL_STATE_OFFHOOK – stav, který nastává v případě, že probíhá telefonní hovor, může být aktivní či pozastavený a žádná další volání nečekají na spojení

První stav zařízení, který vyžaduje aktivitu aplikace je vyzvánění CALL_STATE_RINGING. Během tohoto stavu je uloženo telefonní číslo protistrany, se kterou bude uskutečněn telefonní hovor.

Jakmile se stav telefonu změní na „CALL_STATE_OFFHOOK“, prvním krokem je zjištění času zahájení telefonního hovoru. Následuje příprava cest pro knihovny, nastavení velikosti bufferu pro nahrávání zvukového záznamu a začíná nahrávání zvuku.

4.3.1 Nahrávání telefonního hovoru

Nahrávání telefonního hovoru je realizováno stejným způsobem jako u zvukového záznamu při vytváření uživatelských profilů. Proběhne inicializace rekordéru z třídy *AudioRecord* s příslušnými parametry a je započato nahrávání. Jakmile je nahrávání zahájeno, je vytvořeno vlákno, které převádí nahrávaný zvuk do formátu potřebného pro následný převod zvukového záznamu na hlasovou sotpu. Tímto nahrávacím formátem je WAV.

Nahrávání je ukončeno při přechodu telefonního zařízení ze stavu *CALL_STATE_OFFHOOK* do stavu *CALL_STATE_IDLE*. Je zastaven a uvolněn rekordér a současně s tím je zastaveno i vlákno pro převod zvuku. Tato zvuková stopa je evidována pod názvem „unknown.wav“

4.3.2 Analýza hlasu

Poté co aplikace nahráváním získala hlasový záznam uživatele, je potřeba vytvořit jeho hlasovou stopu pro následnou analýzu obdobně jako u uživatelského účtu. Je vytvořena hlasová stopa „unknown.vp“.

Po splnění předpokladu, že je vytvořena hlasová stopa volajícího uživatele je vše připravené k následující analýze uživatelské hlasové stopy alespoň prozatím neznámého volajícího.

Při zahájení analýzy je vytvořen nový objekt typu *VoicePrintSet*. Tento objekt bude naplněn názvy všech hlasových otisků kromě otisku „unknown.vp“ a bude sloužit jako seznam položek k analýze. Pomocí nově vytvořeného objektu ze třídy *VoicePrintComparator* jsou procházeny a analyzovány všechny hlasové otisky. Do matice typu FloatMatrix jsou následně zapsány hodnoty pro každou hlasovou stopu. Výsledkem je číslo typu float, které je převedeno pomocí metody *rawScore2percent()* na hodnotu v procentech. Je zaznamenáno jméno uživatele s hlasovou stopou, která je nejvíce podobná prozatím neznámé hlasové stopě.

Hranice, zda byl hlas uživatele ověřen, byla stanovena na hodnotu 99%. Pokud je výsledek analýzy některého z hlasových otisků větší než tato hodnota, považuje se výsledek analýzy za pozitivní, kdy je hlas uživatele považován za ověřený. V případě pozitivního výsledku je do statistik zaznamenáno o jakého uživatele se jedná, jaký byl čas uskutečněného volání, délka hovoru v minutách a s jakým telefonním číslem byl hovor uskutečněn.

4.3.2.1 Neoprávněné použití telefonního zařízení

V případě, že uživatel nebyl ověřen, výsledek analýzy žádné z hlasových stop nepřesáhl hranici 99%, je proveden zápis do statistik stejně, jako tomu bylo v případě ověřeného telefonního hovoru, ale za název uživatele je dosazen řetězec „Unknown User“. Rozdíl oproti ověřenému uživateli nastává právě po zápisu do logu dle nastavení uživatele.

Uživatel má možnost zapnout si upozornění e-mailem nebo zprávou SMS. Právě v tento okamžik je zavolána metoda *startService()* a je vytvořena nová služba pro reakci na danou událost. V případě e-mailu je nastaven předmět zprávy na text: „Your phone was used by unauthorized person“, a jako tělo zprávy je nastaven tento stejný řetězec s případnými souřadnicemi GPS, podle uživatelského nastavení. Jako příloha e-mailu je přiložena hlasová stopa neověřeného uživatele. U SMS zprávy se jedná o nastavení těla zprávy na text: „Your phone was used by unauthorized person“, opět s volitelnými GPS souřadnicemi. Všechna tato oznámení jsou zasílána na předem určeného známého uživatele z SQLite databáze, kterým je výchozí uživatel.

Uživatel takto získá základní přehled o využívání svého mobilního telefonu ať už osobou evidovanou v aplikační databázi nebo i třeba osobou cizí, která se snaží zneužít možností telefonního zařízení. Hlasové stopy uživatelů používající mobilní telefon nejsou ukládány, je uložen vždy jen poslední hlasový záznam a nahrávky konkrétních zvukových stop nejsou zpětně dohledatelné. Pokud by uživatel měl zájem evidovat hlasové stopy neověřených uživatelů, může použít upozorňování e-mailem, ve kterém bude mít vždy přiloženou hlasovou stopu.

5 Zabezpečení aplikace

Na zabezpečení zdrojového kódu existuje více metod. Před uvolněním aplikace na Google Play je dobré některou ze zmíněných metod aplikovat. Zabezpečením aplikace se sníží riziko, že bude vyvýjená aplikace odcizena a zneužita pro účely cizích subjektů.

5.1 Export z Eclipse

Při exportu svého projektu ve vývojovém prostředí Eclipse je vývojář vybídnut, aby použil klíč, kterým bude aplikace uzamčena. Pokud vývojář tento klíč nemá dosud vytvořený, je zapotřebí vytvořit nový klíč. Vývojové prostředí Eclipse nabízí možnost vytváření mnohých dalších klíčů nejen pro tuto, ale i pro další aplikace.

První krok k vytvoření nového klíče obnáší zadání názvu souboru klíče a heslo. Název klíče i heslo je možné si stanovit libovolně dle vlastních preferencí. V dalším kroku je nastavována životnost klíče a doplní se další údaje pro podepisování aplikací. Mezi povinné položky při vytváření klíče patří jméno a příjmení. Další informace jsou volitelné a patří mezi ně název organizace, lokalita (stát a město) a kód země (CZ/CZE). S takto vytvořeným klíčem podepíšeme aplikaci a vytvoříme soubor typu APK připravený k nahrání na Google Play.

5.1.1 Ověřování na serveru

Jednou z možností, jak lépe zabezpečit aplikaci proti nekontrolovatelnému šíření je ověřování vůči serveru. Jedná se o ověřování klíče k aplikaci vůči vzdálenému serveru namísto ověřování přímo aplikace u klienta.

5.1.2 Obfuskace

Další z metod jak ztížit podmínky při prolomení bezpečnosti vyvíjené aplikace je obfuskace, neboli zatemnění zdrojového kódu [15].

Zatemnění zdrojového kódu znamená transformaci zdrojového kódu do podoby nečitelné, nebo jen velice obtížně čitelné pro člověka. Dle příslušného nastavení míry obfuskace se může jednat například o vypouštění bílých znaků, změnu názvů proměnných či názvů metod. Do zdrojového kódu v některých případech můžou být přidány i části kódu, které nijak nezdržují běh aplikace, ale činí zdrojový kód ještě méně čitelným.

Pro Android existuje nástroj zvaný ProGuard [16]. Konfigurační soubory k tomuto nástroji, zabudovaném v systému Android, jsou součástí každého nově vytvořeného projektu (proguard.cfg) [17]. Předtím, než bude zapnut tento nástroj na nejen zatemnění, ale i optimalizaci kódu, je potřeba povolit tento nástroj při exportování projektu v souboru project.properties.

Zapnutí nástroje ProGuard se provádí aktivací následujícího konfiguračního řetězce:

```
proguard.config=${sdk.dir}/tools/proguard/proguard-android.txt:proguard.cfg
```

V této vyvíjené aplikaci bohužel není možné použít výše uvedený nástroj, kvůli velkému množství rozšiřujících knihoven, zejména doplňkovým souborům pro knihovnu BSAPI. Tyto knihovny se nedají úspěšně propojit s vyvíjenou aplikací, aby nedošlo k chybám po optimalizaci a zatemnění kódu.

5.1.3 Další metody zabezpečení

Každý vývojář může pozměnit svůj kód, doplnit části, které mají cíleně útočníka zmást a nedávají smysl. Je třeba dávat pozor na zbytečné zatěžování procesoru přílišným množstvím podmínek a ošetření, což by aplikaci zpomalovalo a nadbytečně vytěžovalo zdroje mobilního zařízení.

Kromě těchto metod existují i další metody zabezpečení obsahu aplikace, jako například služba přenášející informace v reálném čase. Data jsou v tomto případě vždy aktuální a neukládají se do paměti zařízení, nebo jsou data ověřování proti vzdálenému serveru.

Všechny změny, týkající se zabezpečení aplikace, je dobré provádět až těsně před uvolněním její finální verze. V opačném případě by se mohlo stát, že se vývojář sám nedokáže pořádně zorientovat ve vlastním zdrojovém kódu.

5.2 Použité zabezpečení

Pro uvolnění vyvíjené aplikace skrze službu Google Play bylo použito zabezpečení knihovny aktivačním klíčem. Součástí knihovny BSAPI je i třída *LicenceManager*. Pří volání metody *getActivationKey()* je získán klíč. Po modifikaci klíče „hashovaní“ metodou *hashMix()* se klíč použije k aktivaci metodou *activate()*. Toto zabezpečení je dle konzultací vývojářů knihovny dostačující. Použití knihovny je vázáno se zdrojovými kódy aplikace a není tak možné její samostatné použití. Pokud dojde k použití neaktivované knihovny je zahlášena chyba, že byl použit neplatný klíč. Životnost aktivačního klíče prozatím nebyla nijak omezena. Aplikace „Speaker ID Watchdog“ je přes tuto knihovnu podepsaná ještě vlastním vývojářským klíčem a tak by se mělo zcela zamezit nekontrolovatelnému šíření aplikace a knihovny pro analýzu hlasu.

6 Marketing

Hlavním cílem vyvíjené aplikace „Speaker ID Watchdog“ je předvést široké veřejnosti knihovnu BSAPI, konkrétně její verzi pro Android a ukázat její funkce na praktickém použití na mobilních zařízeních.

V těchto dnech je na Google Play dostupná celá řada aplikací, které nabízí zabezpečení telefonu. Žádná z nich ale nevyužívá data získaná během telefonního hovoru pro analýzu volajícího a ověření jeho totožnosti. S tímto jedinečným modelem by se aplikaci „Speaker ID Watchdog“ mohlo podařit zaplnit sice nepatrnou, ale znatelnou mezeru na trhu.

6.1 Prezentace knihovny BSAPI

Primární snahou vývojářů aplikace je také v praxi předvést přednosti hlasové analýzy v rámci knihovny vyvíjené na Fakultě informačních technologií v Brně.

Každý chytrý telefon má svůj primární účel, tím je aby se z tohoto zařízení dalo telefonovat. Aplikace musí být uživatelsky zajímavá a měla by přinášet určité obohacení ve smyslu funkcí chytrých telefonů a to nejen cíleně, ale také při jejich rutinním používání. Tento projekt nabízí uživateli více možností evidence a analýzy telefonních hovorů. Každý uživatel svého mobilního zařízení si nastaví míru zabezpečení dle svých vlastních preferencí. Buďto jen zápis do logovacího souboru nebo upozorňování pomocí definovaných služeb. Tyto služby, jako je odesílání SMS zpráv nebo e-mailů a s tím související aktivace mobilního internetu, jsou ve většině případech zpoplatněny, proto záleží jen na uživateli, jakou míru upozorňování si zvolí.

6.2 Nahrávání hovorů

Projekt analýzy hlasu je založen na nahrávání hlasových stop, kde je jako zdroj zvuku použit mikrofon mobilního zařízení. Nahrávání druhé strany není v systému Android dovoleno. Kdyby tato možnost byla povolena, musel by být uživatel protistrany patřičně informován a musel by souhlasit, že bude hovor nahráván. Tato možnost by ze strany platformy Android existovala, pokud by se jednalo o „root“ zařízení. Tato myšlenka naplňuje původní vize firmy Phonexia s.r.o., ale záměr aktuálně vyvíjené aplikace je alespoň prozatím jiný. Další vývoj aplikace bude do určité míry záviset na zpětné vazbě, informacích a zkušenostech od uživatelů.

Nahrávání telefonních hovorů [18] je obecně realizováno na základě vyslovení senátu Ústavního soudu ČR. Ten ve svém nálezu ze dne 13. září 2006, sp. Zn. I. ÚS 191/05 uvádí: *„Každý má právo zaznamenávat své telefonické hovory“*.

Jak uvádí Mgr. Petr Kubačka (2009) [18], použití telefonní nahrávky je s ohledem na ochranu osobnosti možné se svolením zaznamenaného, či toho, koho se záznam týká. Aby bylo zabráněno zneužití nahrávání telefonního hovoru, po nahrávání je již dále používána výhradně hlasová stopa, která nese pouze identitu uživatele, který hovor uskutečnil, nikoliv obsah hovoru či předmět uskutečněného jednání. Touto identitou je pouze hlas nahrávaného. V tomto případě se jedná o záznam nezachycující projevy osobnostní povahy, mezi které patří například právo na soukromí a rodiný život (Pecina (2012)) [19]. Takovéto položky by byly chráněny právem na ochranu osobnostní povahy. V českém právním řádu je ochrana osobnosti zakotvena v čl. 10 Listiny základních práv a svobod. *„Provedení důkazu soukromě pořízeným audiozáznamem nevylučuje ust. § 125 o.s.ř.; to tím spíše byl-li takový záznam učiněn v prostoru určeném pro veřejnost a netýká se soukromí zúčastněných osob.“* (Kubačka,M., 2009). Proto je v tomto případě možné použít hlasovou stopu uživatele jako důkaz o použití telefonního zařízení.

6.3 Další možnosti hlasové analýzy pro Android

V budoucnu bude pravděpodobně možné využívat hlasovou analýzu i například na odemykání telefonu, nebo svůj hlasový otisk použít jako klíč pro přístup k aplikacím. V ideálním případě pak bude možné používat svůj hlas jako alternativu k elektronickým podpisům či ověřování důvěryhodnosti komunikačního zdroje a tak omezit bezpečnostní rizika s tím spojená.

6.4 Vzhled aplikace

V rámci předlohy dodané vzorové aplikace SID-Phonexia pro Android a materiálů firmy Phonexia s.r.o. byly použity barvy: černá, bílá a oranžová.

Prostředí aplikace je nastavené tak, aby působilo jednoduchým čistým dojmem, aby uživatel mohl aplikaci používat co nejpohodlněji. Přehlednost chodu aplikace je reprezentována vyskakujícími okny běžících vláken a případně upozorněním ve formě doplňujících dotazů, aby byly kroky při používání aplikace co nejvíce automatizované. Uživatel je v mnoha případech informován o běhu aplikace, aby neustále věděl co se v běžící aplikaci děje a nebyl zmatený z neurčitého chování aplikace. Pokud například uživatel zadá nekorektní informace, nebo z různých důvodů neproběhne nahrávání zvukové stopy, je upozorněn díky widgetu *Toast* a metodě *makeText()* o konkrétní chybové či výstražné hlášce.

6.5 Distribuce aplikace

Pro rozšíření aplikace byl zvolen distribuční kanál Google Play, který jako jediný zaručí distribuci pro nejvíce mobilních zařízení.

Cena aplikace byla stanovena na 0,- Kč. Aplikace bude šířena jako free verze. V případě, že bude o aplikaci zájem, bude potřeba stanovit cenu na základě poptávky. Předpokládá se, že tato free verze může obsahovat řadu nedostatků, které se nepodařilo objevit během testování. O ziscích z distribuce aplikace bude pojednáváno v kapitole 8 Google Play.

6.5.1 Rozšíření funkcí aplikace v budoucnosti

Na aplikaci „Speaker ID Watchdog“ se může vázat sada určitých doplňků a možností. Například již dříve zmíněný modul na odemykání telefonu pro daného uživatele ve víceuživatelském prostředí. Nově vyvíjené verze Androidu „KeyLime“ již toto protředí má obsahovat. Další doplňkové moduly pro telefony s oprávněním „root“ by tak mohly umožnit uživatelům nahrávat obě komunikující strany.

Další z možných rozšíření, které připadá v úvahu, je i evidence všech neověřených zvukových stop, jejich zpětné dohledání a následná analýza, která by tyto hlasové stopy zpětně propojila s konkrétními uživateli. Import porovnávaných zvukových stop by mohl probíhat jak prostřednictvím mikrofonu, tak i vložením WAV souboru přímo do aplikace.

S neustálým zdokonalováním použité hlasové biometrie se jistě najde ještě mnoho dalších využití, které by mohly způsobit na tomto poli průlom.

7 Testování

Testování aplikace „Speaker ID Watchdog“ se vyvíjelo a modifikovalo spolu s vyvíjenou aplikací. Od prvních testů na emulátoru systému Android až po testování finální aplikace na fyzických zařízeních.

7.1 Android Virtual Device Manager

V počátečním období vývoje aplikace pro platformu Android bylo vytvořeno virtuální zařízení zastupující fyzické zařízení Nexus S. API level byl nastaven na 10, tudíž se jednalo o Android 2.3.3.

Po prvních pokusech jednoduchých aktivit bylo potřeba zajistit rychlejší odezvu a již při práci s databází se emulátor jevil jako poměrně nešikovný a pomalý.

Pro testování odesílání SMS zpráv bylo vytvořeno další, druhé virtuální zařízení s obdobným hardwarovým vybavením na API levelu 16, Android 4.1.2. Prostředí Eclipse nabízí možnosti pohodlné správy emulátorů jak přes AVD, tak i přes pohled „Emulator Control“, ve kterém je jasně zobrazen přehled možností jak interagovat se zařízeními. Každý emulátor má definované číslo pro odesílání SMS zpráv. Prvnímu zařízení bylo přiděleno číslo 5554 a druhému bylo přiděleno číslo 5556. Po úspěšném otestování této verze na emulátoru bylo možné přejít na fyzická zařízení, která nabízí i reálné možnosti sledování telefonních hovorů.

Další vývojové testování bylo prováděno na později zmiňovaných fyzických zařízeních, protože odezva při práci s aplikací, nahrávání nebo ladění aplikace v tomto příapadě byla několikanásobně rychlejší.

7.2 HW zařízení

7.2.1 API level 17 Android 4.2 Jelly Bean

Vývoj aplikace „Speaker ID Watchdog“ došel do stádia, kdy bylo potřeba začít testovat aplikaci na reálném fyzickém zařízení. Pro tyto účely byl použit tablet ASUS Transformer Pad TF300T. Díky svižnému 5-ti jádrovému procesoru Tegra 3 od firmy NVIDIA byl vývoj značně efektivnější než při práci s emulátorem. Zařízení bylo používáno do doby, než bylo zapotřebí otestovat funkce aplikace přímo na mobilním telefonu. Jednalo se o odesílání SMS zpráv a práci se správou telefonních hovorů.

7.2.2 API level 10 Android 2.3.3 Gingerbread

Další zařízení v řadě, na kterých probíhalo finální testování a ladění aplikace, byla zařízení Samsung Galaxy W a Samsung Galaxy Ace. Tato zařízení byla použita na testování finální podoby aplikace a

to především reakce na telefonní hovory. Tento typ mobilních telefonů dokáže dát poměrně přesné informace o práci aplikace i na jiných zařízeních se systémem Android. Tato zařízení se řadí do střední třídy mobilních zařízení, a tudíž patří i mezi jedny z nejrozšířenějších typů chytrých telefonů.

7.3 Finální testování před uvolněním aplikace

Jako každou aplikaci bylo nutné před uvolněním řádně otestovat i aplikaci „Speaker ID Watchdog“. Po průběžném testování vývojářem nastupuje beta testování. Toto testování slouží především k eliminaci nespokojenosti budoucích uživatelů této aplikace. Samotné testování probíhalo u vedoucího této bakalářské práce Ing. Petra Schwarze, Ph.D. a u vybraných zaměstnanců firmy Phonexia s.r.o..

8 Google Play

Distribučním kanálem pro zařízení s operačním systémem Android je Google Play. Jedná se o službu, dříve nazývanou Android Market, která slouží na stahování her, aplikací a různých doplňků pro daný systém. Tuto službu zaštiťuje přímo společnost Google.

Služba Google Play je podporována u valné většiny mobilních zařízení. Není potřeba se bát, že nepodporovaných zařízení s tímto systémem je mnoho. Existují ale také různá speciální zařízení, jako jsou například průmyslové tablety a mobilní telefony méně známých a dostupných značek, kterých je na našem trhu ale velmi málo. V případě že pro dané mobilní zařízení není k dispozici služba Google Play, není ještě boj prohraný. Vývojáři mobilních aplikací pro platformu Android mohou šířit své aplikace i jinými distribučními kanály. Stačí jen uvolnit k distribuci potřebný APK soubor, nahrát soubor do zařízení a provést instalaci. Další možností je stahování aplikací přes počítač. Od roku 2011 mohou uživatelé procházet a stahovat aplikace do svého osobního počítače a následně provést instalaci na svém mobilním zařízení. Některé aplikace jsou vyvíjeny pouze pro speciální typy mobilních zařízení. Jedná se především o aplikace určené pro běžně dostupné chytré telefony a aplikace optimalizované pro tablety. Mezi těmito aplikacemi jsou značné rozdíly. Mohou to být jak rozdíly v rozlišení obrazovek, uživatelsky přívětivé ovládání pro dané typy zařízení nebo i optimalizace pro daný typ použitého procesoru v zařízení.

Některé konkrétní aplikace jsou vázané nejen k daným typům mobilních zařízení, nebo verzím Androidu, ale mohou být vázané i s daným státem. Toto donedávna platilo pro vývojáře z České Republiky, kteří mohli uvolňovat aplikace jen přes zahraniční.

8.1 Programové zásady

Pro distribuci aplikací i zobrazované reklami přes službu Google Play platí určené Obsahové zásady [20].

Pomocí služby Google play je zakázáno šířit obsah se sexuálně explicitnímy materiály, násilím a šikanováním, s projevy nenávisti vůči určitým skupinám osob, například z etnických, rasových důvodů, nebo kvůli pohlaví, či sexuální orientaci. Je zakázáno předstírat cizí identitu nebo klamavě jednat a podvrhovat uživatelům klamné informace. Platí zákaz zveřejňování osobních a důvěrných informací. Není dovoleno porušovat práva k duševnímu vlastnictví jiných uživatelů a autorských práv. Zákaz podpory nezákonné činnosti, hazardních her a nebezpečných produktů jako jsou viry, červi, trojské koně a cokoliv, co by mohlo poškodit uživatele nebo jejich zařízení.

Nákupy aplikací jsou realizovány pomocí platebních karet evidovaných ve službě Google Play.

Jsou zde uvedeny zásady pro reklamy a šíření aplikací. Reklama musí být jasně přiřazená a identifikovatelná ke konkrétní aplikaci a nesmí ovlivňovat fungování zařízení. Uživatel musí mít

možnost reklamu odstranit, případně i s odinstalováním aplikace. Uživatel musí mít možnost reklamu bez postihu odmítnout. Reklamy, které blokují chod aplikace dříve, než uživatel provede určité kroky, jsou přísně zakázané. Žádné hlášení aplikace či reklamy nesmí mít podobu systémového ohlášení nebo chyby.

8.2 Distribuční smlouva

Distribuční smlouva [22] služby pro vývojáře vymezuje základní pojmy používané v celém distribučním prostředí.

„Pokud za své produkty chce vývojář účtovat poplatky, musí také získat a mít platný Platební účet od autorizovaného Zpracovatele plateb“. Pro distribuci aplikací přes Google Play musí každý vývojář souhlasit s podmínkami distribuční smlouvy.

V distribuční smlouvě jsou uvedeny pokyny jak postupovat při zpoplatňování aplikací. Služba Google Play může uživatelům nabízet převedenou cenu do místní měny uživatele, neručí ale za správnost kurzu. Je zakázáno vybírat transakční poplatek u aplikací zdarma a není dovoleno vybírat dodatečné poplatky za kopie produktů, které si uživatelé mohli stáhnout zadarmo. Všechny poplatky, které by vývojář přijal za daný produkt pomocí Google Play musí projít přes Zpracovatele plateb.

Společnost Google umožňuje uživatelům, kteří neměli možnost vyzkoušet si aplikaci před jejím zakoupením, vrácení peněz, pokud o to uživatel požádá do 48 hodin. Pokud se jedná o aplikaci, kterou si uživatel může prohlédnout, je zakázáno vracet uživatelům peníze.

Podpora, stížnosti a dotazy jsou všechny delegovány na vývojáře aplikace a záleží jen na něm, jak se s nimi vypořádá. Nedostatečná podpora vlastních produktů zřejmě dopadne na hodnocení produktu a tím pádem spadne v žebříčku umístění. Uživatelé mohou bez omezení aplikace reinstalovat.

Google Play na vystavené aplikaci nemá žádný podíl ani nárok, včetně duševního vlastnictví.

Každý vývojář nese zodpovědnost za zachování důvěryhodnosti vývojáře společnosti Google a za vývoj aplikace v rámci svých pověření.

Data zjískaná z službou Google Play slouží pro vytváření statistik o používání služeb a tyto data mohou být vývojářům za účelem zlepšování produktů na základě zádosti zpřístupněna.

V případě ukončení vývojářského poměru se společností Google je výpovědní lhůta stanovena na třicet dní. Společnost Google tuto smlouvu může kdykoliv ukončit při porušení smlouvy o distribuci aplikací v případě ztráty statusu „autorizovaný vývojář“ nebo v případě zrušení služby Market.

Společnost Google jasně odmítá záruky jakéhokoliv druhu. Společnost Google anebo žádná její dceřiná společnost nenesou žádnou zodpovědnost.

8.3 APK Expansion Files

Aktuální nastavení služby Google Play je připraveno pro APK soubory do velikosti 50MB. Pro velké možství aplikací je to dostačující prostor, ale tento limit byl během vývoje aplikace „Speaker ID Watchdog“ překážkou. V případě, že je aplikace větší než 50MB, je možné dodatečný obsah aplikace stáhnout díky právě těmto vytvořeným balíčkům nazývaných APK Expansion Fines [22]. V případě služby Google Play toto již zmíněné rozšíření velikostní náročnosti aplikace není nijak zpoplatněno. Stanovená hranice 50MB je nastavena z důvodu, aby se vývojíři snažili optimalizovat své aplikace a šetřili tak i místo na uživatelských zařízeních.

Rozšiřující balíčky jsou službou Google Play stahovány společně se samotnou aplikací. Ukládání rozšiřujících souborů probíhá na „external storage“, což může být buď SD karta, nebo přímo vnitřní pamět zařízení. Služba Google Play může být ve chvíli kdy probíhá stahování natolik vytížená, že zamítne stahování rozšiřujících balíčků. V tomto případě při spouštění aplikace probíhá kontrola potřebných rozšíření, a pokud je zjištěna jejich nepřítomnost, je v danou chvíli zahájeno stahování. V případě, že stahování již proběhlo, aplikace se spustí běžným způspbem.

Každý projekt, který byl nově přidán skrze vývojářskou konzoli „Google Play Developer Console“ může přidat jeden nebo dva takto rozšiřující balíčky. Celková velikost těchto balíčků se pohybuje až do velikosti 2GB pro každý z nich. Rozšiřující balíček může být téměř v libovolném formátu, jako jsou například ZIP, PDF, MP4. Vývojářem je zvolen typ balíčku a je určeno, zda se jedná o jádro *main* nebo záplatu *patch.* V poslední řadě je definována cesta pro uložení potřebných souborů.

8.4 Zpoplatnění služeb

8.4.1 Registrace vývojářů

Pokud se vývojář rozhodne, že chce svoji aplikaci distribuovat pomocí služby Google Play, musí se zaregistrovat jako Google Android vývojář. Za tuto registraci se v případě Androidu platí jednorázová částka ve výši 25$. Ve srovnání s ostatními mobilními operačními systémy je to částka přijatelná. Pro srovnání s OS Windows pro mobilní zařízení je nutné platit poplatek za každý další rok a to v hodnotě 99$, tak stejně je tomu i u iOS. Jen OS Symbian si drží cenu vemi nízko a to pouze 1€ a to jednorázově.

8.4.2 Ceny aplikací a poplatky

Aplikace, které jsou ke stažení, se dají rozdělit do několika základních typů dle zpoplatnění. Dělí se na placené a freeware, tedy neplacené aplikace, která mají díky své bezplatnosti častokrát svá specifika.

Placené – tento typ aplikací je potřeba zaplatit při stažení aplikace. Výše částky je vždy uvedena pro Českou Republiku v korunách českých (Kč). Může se jednat i o trial verze (zkušební verze), kdy je potřeba po určitém časovém období aplikaci zaplatit, jinak dojde k zablokování přístupu do aplikace.

Freeware – tyto aplikace se ještě dají rozdělit na další tři podkategorie:

Aplikace s omezenou funkcionalitou – tyto aplikace jsou poskytovány zdarma. Jejich nevýhodou zůstává fakt, že velká část některých především uživatelsky zajímavých funkcí je blokována a odblokování daného rozšíření proběhne až po zaplacení poplatku.

Aplikace obsahující reklamy – tento typ aplikací je poskytován také zadarmo a uživatel není ve většině případů omezován deaktivovanými nebo zablokovanými prvky, které je potřeba aktivovat. Aby byl z aplikací generován nějaký zisk, především pokud se jedná o aplikace hojně rozšířené, jsou v nich zobrazované reklamy. Dle úvahy distributora nebo vývojáře je umístěna reklama na vybrané místo. Zisk je generován prostředníctvím reklamy.

Aplikace zadarmo – ano, na Google Play je i obrovské, téměř nepřeberné množství aplikací, které jsou zcela zdarma bez jakéhokoli omezení.

Poslední z marketingových tahů, jak vydělat na distribuci aplikací, je transfer peněz na kreditový systém využívaný aplikací. Tento systém funguje tak, že pokud si uživatel doplatí určitou částku, je mu tato částka převedena na předem známou kreditovou hodnotu do aplikace. V praxi se tento systém podobá žetonům v kasinu. V určené aplikaci si za získané kredity může uživatel obstarat určité bonusy. Například v kancelářských aplikacích se s tímto přístupem moc často uživatel nesetká, ale naopak velice často se tento kreditový systém objevuje v hrách. Soutěživost potom stojí některé hráče nemalé částky.

Podíl ze zisku na dané aplikaci, takzvaný „Transakční poplatek“ tvoří 30% pro společnost Google. Stejně tak je tomu ale i u všech ostatních dříve zmíněných mobilních operačních systémů.

8.5 Developer Console

V případě že se vývojář pod svým účtem Google zaregistruje jako vývojář pro Android aplikace, distribuované přes službu Google Play, je mu zpřístupněna Vývojářská konzole *Developer Console*.

Pomocí tohoto prostředí [23] jsou aplikace schvalovány a uvolňovány na Google Play. Tato konzole obsahuje správu všech distribuovaných aplikací podle jména aplikace, ceny, počtu stažení a počtu aktuálně používaných instalací. V záložce „Financial reports“ jsou dostupné informace o zisku z distribuovaných aplikací.

Pro každou aplikaci, která bude díky této službě Google Play distribuovaná, je zde formulář, do kterého se vyplňují informace, zobrazené před stažením a instalací aplikace. Pro vývojáře toto prostředí přináší ale i další užitečné informace jako je přehled hodnocení a komentářů k uvoněné aplikaci, přehled statistik instalací a další.

9 Závěr

Během této práce byla vyvinuta aplikace pro mobilní telefon se systémem Android, která informuje majitele telefonu o jeho neautorizovaném použití. Ověření autorizovaných osob se provádí pomocí technologie identifikace řečníka. K vlastní identifikaci byla použita knihovna BSAPI vyvíjená firmou Phonexia s.r.o..

Před zahájením vývoje aplikace bylo potřeba se seznámit se systémem Android obecně. Jak tato moderní platforma funguje a jak bude vývoj probíhat. Jako vývojové prostředí bylo zvoleno Eclipse IDE, protože se jedná o jediné oficiálně podporované vývojové prostředí pro Android. Po seznámení se základními principy Androidu se přistoupilo k seznámení s používanou metodou hlasové analýzy i-vectors. Jednalo se především o studium dodané knihovny a vzorové aplikace, která s knihovnou pracuje. Na základě prostudování dodaných matriálů byl vytvořen návrh a zahájen vývoj aplikace. Po implementaci a testování jednotlivých částí aplikace, jako je například databáze uživatelů nebo sledování telefonních hovorů, byla sestavena první verze aplikace. Jako další krok byla pozornost věnována přehlednosti a uživatelské přívětivosti aplikace. Po čas celého vývoje byla použita knihovna BSAPI verze 0.0.6. Testování probíhalo během implementace i po dokončení aplikace. Na testování musí být kladem velký důraz, z důvodu uvolnění aplikace službou Google Play. Aplikace by měla být pro uživatele zajímavá a stabilní, aby dokázala zaujmout uživatele hned od uvolnění. Úpravy nebo rozšíření aplikace jsou dále závislé na požadavcích uživatelů.

Ve finální verzi aplikace byla použita knihovna BSAPI ve verzi 1.2.0, která musí být před použitím v rámci aplikace digitálně podepsaná, aby se zcela zabránilo nekontrolovatelnému šíření používané knihovny. Než bude aplikace uvolněna službou Google Play, proběhne ještě redukce velikosti používané knihovny, která v současnosti zabírá skoro 50MB, aby nedošlo k přetěžování mobilních zařízení.

Aplikace „Speaker ID Watchdog“ nyní prochází finálním testováním. Po testování a doladění detailů bude aplikace uvolněna na Google Play.

Literatura

[1] MARVAN, Filip. Mobilní operační systém Android. *Mobilní operační systém Android* [online]. 2011 [cit. 2013-05-02]. Dostupné z: http://diit.cz/clanek/mobilni-operacni-system-android

[2] Android: About. *Android* [online]. [cit. 2013-05-02]. Dostupné z: http://www.android.com/about/

[3] GLEMBEK, Ondřej, BURGET, Pavel MATĚJKA, Martin KARAFIÁT a Patrick KENNY. Simplification and Optimazation of I-VECTOR extraction. Simplification and Optimazation of I-VECTOR extraction. 2011, č. 1, s. 4. Dostupné z: http://www.fit.vutbr.cz/research/groups/speech/publi/2011/glembek_icassp2011_4516.pdf

[4] Developer Tools. In: *Developer Tools* [online]. 2012 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/tools/index.html

[5] HAVRYLUK, Michal. Naučíme vás programovat aplikace pro android. In: *Naučíme vás programovat aplikace pro android.*[online]. 2012 [cit. 2013-05-02]. Dostupné z: http://mobil.idnes.cz/naucime-vas-programovat-aplikace-pro-android-zaciname-prave-dnes-phe-/aplikace.aspx?c=A120410_125436_aplikace_ham

[6] Soubor AndroidManifest.xml. In: *Soubor AndroidManifest.xml* [online]. [cit. 2013-05-02]. Dostupné z: https://sites.google.com/site/androiddevguidecz/tema-frameworku/2/soubor-androidmanifest-xml

[7] Android: How to copy files in 'assets' to sdcard?. In: *Android: How to copy files in 'assets' to sdcard?* [online]. 2013 [cit. 2013-05-02]. Dostupné z: http://stackoverflow.com/questions/4447477/android-how-to-copy-files-in-assets-to-sdcard

[8] L. MURPHY, Mark. *Android 2: Průvodce programováním mobilních aplikací*. Brno: Computer Pres, a.s., 2011. ISBN 978-80-251-3194-7.

[9] *Javamail-android* [online]. 2009 [cit. 2013-05-02]. Dostupné z: https://code.google.com/p/javamail-android/downloads/list

[10] Sending Emails without User Intervention (no Intents) in Android [online]. 2010 [cit. 2013-05-02]. Dostupné z: http://www.jondev.net/articles/Sending_Emails_without_User_Intervention_(no_Intents)_in_Android

[11] FARLEY, Matt. Sending Emails without User Intervention (no Intents) in Android. In: *Sending Emails without User Intervention (no Intents) in Android* [online]. 2010 [cit. 2013-05-02]. Dostupné z: http://www.jondev.net/articles/Sending_Emails_without_User_Intervention_(no_Intents)_in_Android

[12] MEIER, Reto. *Professional Android 4 application development*. Updated for Android 4. Indianapolis: John Wiley, 2012, xlii, 817 p. ISBN 978-111-8262-153.

[13] PhoneStateListener. In: *PhoneStateListener* [online]. 2013 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/reference/android/telephony/PhoneStateListener.html

[14] TelephonyManager. In: *TelephonyManager* [online]. 2013 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/reference/android/telephony/TelephonyManager.html

[15] Security and Design. In: *Security and Design* [online]. 2012 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/google/play/billing/billing_best_practices.html

[16] LAFORTUNE, Eric. ProGuard. *ProGuard* [online]. 2013 [cit. 2013-05-02]. Dostupné z: http://proguard.sourceforge.net/

[17] ProGuard. In: *ProGuard* [online]. 2012 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/tools/help/proguard.html

[18] KUBAČKA, Petr. Záznam telefonního rozhovoru pořízený jeho účastníkem jako důkaz v občanskoprávním řízení. In: *Záznam telefonního rozhovoru pořízený jeho účastníkem jako důkaz v občanskoprávním řízení* [online]. 2009 [cit. 2013-05-02]. Dostupné z: http://www.ipravnik.cz/cz/clanky/obcanske-pravo/art_6200/zaznam-telefonniho-rozhovoru-porizeny-jeho-ucastnikem-jako-dukaz-v-obcanskopravnim-rizeni.aspx

[19] Právo na ochranu osobnosti. In: PECINA, Tomáš. *Právo na ochranu osobnosti* [online]. 2012 [cit. 2013-05-02]. Dostupné z: http://iuridictum.pecina.cz/w/Právo_na_ochranu_osobnosti

[20] Programové zásady služby Google Play pro vývojáře. In: *Programové zásady služby Google Play pro vývojáře* [online]. 2013 [cit. 2013-05-02]. Dostupné z: https://play.google.com/about/developer-content-policy.html

[21] Distribuční smlouva služby pro vývojáře. In: *Distribuční smlouva služby pro vývojáře* [online]. 2013 [cit. 2013-05-02]. Dostupné z: http://play.google.com/intl/ALL_cz/about/developer-distribution-agreement.html

[22] APK Expansion Files. In: *APK Expansion Files* [online]. 2012 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/google/play/expansion-files.html

[23] *Developer Console* [online]. 2012 [cit. 2013-05-02]. Dostupné z: http://developer.android.com/distribute/googleplay/publish/console.html

[24] LOGO PICTURES, *Android Logo* [online]. 2013 [cit. 2013-05-10]. Obrázek ve formátu PNG. Dostupné z http://www.ranklogos.com/websites-logos/android-logo/

Seznam příloh

- Příloha 1. – CD se zdrojovými kódy aplikace

Příloha 1. – CD se zdrojovými kódy aplikace

Přiložené CD obsahuje následující adresáře a soubory:

`src`	adresář obsahující zdrojové kódy aplikace
`libs`	adresář obsahující knihovny pro běh aplikace, bez knihovny BSAPI
`res`	adresář obsahující nejen grafické zdroje aplikace
`readme.txt`	soubor s pokyny pro kompilaci aplikace
`manual.txt`	soubor popisující instalaci a ovládání aplikace
`AndroidManifest.xml`	soubor popisující rámcově celou aplikaci
`bachelor_thesis.pdf`	elektronická verze textu bakalářské práce
`bachelor_thesis.doc`	elektronická, upravitelná verze textu bakalářské práce
`SpeakerIDWatchdog.apk`	výsledná aplikace pro službu Google Play