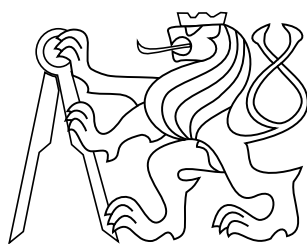


bakalářská práce

Integrace služby internetového/audio rádia

Lukáš Chvátal



Květen 2016

Vedoucí práce: Ing. Daniel Novák, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická, Katedra počítačové grafiky a interakce

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Lukáš Chvátal**

Studijní program: Softwarové technologie a management
Obor: Web a multimedia

Název tématu: **Integrace služby internetového/audio rádia**

Pokyny pro vypracování:


1. Seznamte se s problematikou rozhraní pro implementaci služby internetového/audio rádia.
2. Navrhněte řešení přístupu k vybrané knihovně pro zrakově nevidomé, použijte existující API, dále navrhněte GUI, které přístup bude realizovat.
3. Řešení naimplementujte v prostředí Android.
4. Řešení otestujte na 3 nevidomých uživatelích.

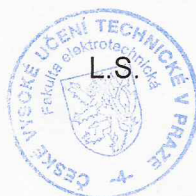
Seznam odborné literatury:

- [1] J. Lazar, Ensuring Digital Accessibility through Process and Policy, Elsevier, 2015
[2] K. Cunningham, Accessibility Handbook, O'Reilly Media, 2012

Vedoucí: Ing. Daniel Novák, Ph.D.

Platnost zadání: do konce zimního semestru 2017/2018


prof. Ing. Jiří Žára, CSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 29. 2. 2016

Poděkování

Rád bych poděkoval své rodině, přítelkyni a přátelům za podporu, vedoucímu práce Ing. Danielu Novákovi, Ph.D. za cenné připomínky, doc. Ing. Zdeňku Míkovcovi, Ph.D. za užitečné rady k testování a závěrem také samotným účastníkům testování vyvinuté aplikace.

Prohlášení

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

V Praze dne 24. května 2016

Podpis autora

Abstrakt

Práce se zabývá analýzou a implementací mobilní aplikace, která umožní zrakově postiženým osobám poslech internetových rádií vysílajících v jejich jazyce. Stanice jsou získávány ze serverů služeb poskytující databáze radiových stanic a v aplikaci jsou řazeny v kompletním seznamu nebo podle hudebních kategorií. Uživatel si také může vytvořit seznam oblíbených stanic. Aplikace je vyvinuta pro mobilní platformu Android.

Klíčová slova

Android; internetové rádio; nevidomí

Abstrakt

This bachelor thesis consist of analysis and implementation of mobile application that allows visually impaired people to access internet radios in their language. Stations are obtained from serveres providing databases of radio stations. In application, stations are sorted into a list of all stations and into categories by music genres. User is also allowed to manage list of favourite stations. The application is developed under Android mobile platform.

Keywords

Android; internet radio; visually impaired

Obsah

1 Úvod	1
1.1 Cíl práce	2
2 Operační systém Android	3
2.1 Android architektura a interface	3
2.1.1 Application framework	4
2.1.2 Binder IPC	4
2.1.3 System services (Systémové služby)	4
2.1.4 Hardware abstraction layer (HAL)	4
2.1.5 Linux kernel (Linuxové jádro)	4
2.2 Struktura Android aplikace	4
2.2.1 Activity (Aktivita)	4
2.2.2 Service (Služba)	5
2.2.3 Content providers (Poskytovatelé obsahu)	5
2.2.4 Broadcast receiver	5
2.2.5 Intent (záměr)	5
3 Analýza existujících aplikací	6
3.1 TuneIn	6
3.2 XiiaLive	7
3.3 Radio FM	8
3.4 RadioTunes	9
3.5 Poznatky pro návrh aplikace Internetové rádio	10
4 Návrh a analýza řešení	11
4.1 Získávání dat pro přehrávání stanic	11
4.1.1 Dirble	12
4.1.2 Play.cz	13
4.2 Funkce a klíčové vlastnosti aplikace	15
4.3 Struktura aplikace	16
4.4 Shrnutí návrhu	17
5 Implementace aplikace	18
5.1 Model - třída Station	18
5.2 Uchovávání dat - třída RadioData	19
5.3 API Service	19
5.3.1 Knihovna Volley	19
5.3.2 ApiListener	20
5.3.3 ServicePlayAPI	20
5.3.4 ServiceDirbleAPI	21
5.4 Ukládání nastavení uživatele - třída PreferencesService	22
5.5 Spuštění aplikace	23
5.5.1 Zobrazení stanic	23
5.5.2 Vyhledávání	24
5.5.3 Výběr a přehrávání stanice	24
5.6 Listování kategoriemi	26
5.7 Oblíbené stanice	26
5.8 Pokračování v přehrávání	27

6 Testování	28
6.1 Test použitelnosti	28
6.2 Usability lab	28
6.3 Uživatelé	29
6.3.1 Screener	29
6.3.2 Dotazník před testem	29
6.3.3 Dotazník po testu	29
6.4 Průběh testování	29
6.5 Poznatky z testování	30
6.6 Nálezy a návrhy na zlepšení	31
7 Závěr	32
Přílohy	
A Hodnocení funkce Rádio jako součást aplikace pro nevidomé	33
A.1 Úvod	33
A.2 Uživatelské prostředí	33
A.3 Problematická místa	33
A.4 Návrhy na úpravu	33
A.5 Shrnutí	34
B Dotazníky	35
B.1 Screener	35
B.2 Dotazník před testem	35
B.3 Dotazník po testu	35
C Participanti - získaná data	36
C.1 Participant 1	36
C.1.1 P1 - Screener	36
C.1.2 P1 - Dotazník před testem	36
C.1.3 P1 - Dotazník po testu	36
C.2 Participant 2	37
C.2.1 P2 - Screener	37
C.2.2 P2 - Dotazník před testem	37
C.2.3 P2 - Dotazník po testu	37
C.3 Participant 3	37
C.3.1 P3 - Screener	37
C.3.2 P3 - Dotazník před testem	38
C.3.3 P3 - Dotazník po testu	38
D Obsah příloženého CD	39
Literatura	40

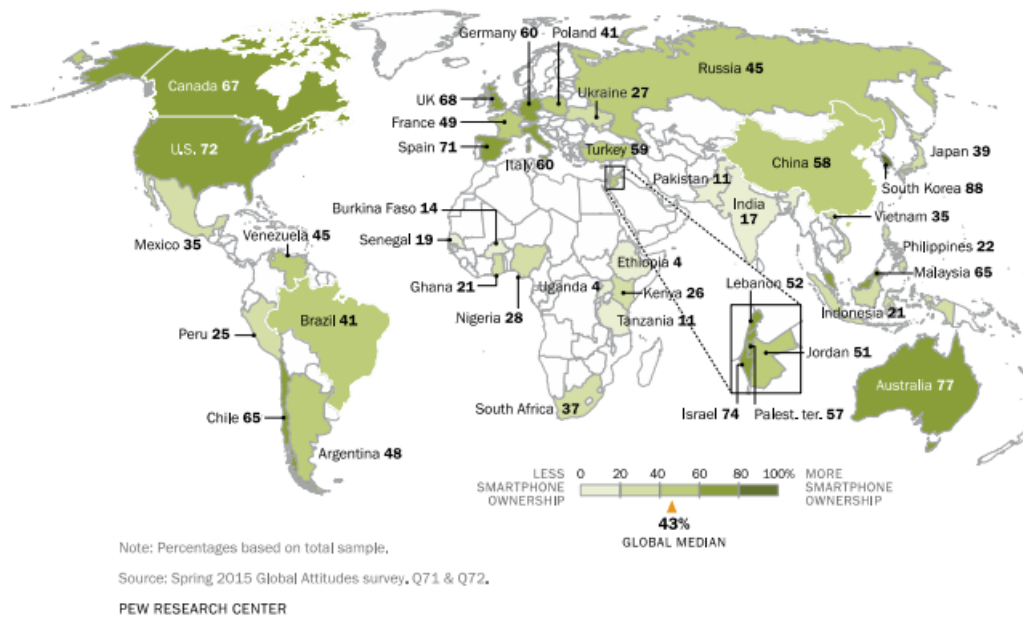
Zkratky

V textu se vyskytují tyto zkratky.

API	Application Programming Interface
REST	Representational State Transfer
JSON	JavaScript Object Notation

1 Úvod

Chytrý telefon (anglicky smartphone) je dnes již velmi rozšířené komunikační a multimediální zařízení. Ve vyspělých zemích ho používá více jak polovina dospělých lidí - například ve Spojených státech potvrzuje používání chytrého telefonu 72 % dospělých, v Kanadě 67 %, ve Velké Británii 68 % a v Austrálii dosahuje toto číslo 77 %. Globální medián je pak 43 %. Tyto čísla neustále rostou i v zemích s rozvíjející se ekonomikou.[1] Chytrý telefon využívá mobilní operační systém (např. Android, iOS, Windows Phone atd.) a aplikační rozhraní, které umožňuje instalaci programů do prostředí operačního systému. V kombinaci s přístupem k internetu tak tento telefon nabízí nepřeborné množství možností využití. Většina chytrých telefonů je ovládána pomocí dotykové obrazovky, a to i pro zadávání textu - od roku 2010 je na trhu jen několik modelů chytrých telefonů s hardwarovou klávesnicí.[2] Dotykové ovládání je pro běžné uživatele intuitivní a přehledné, avšak značný problém představuje pro uživatele se zrakovou indispozicí, pro které je hmat jedním z nejdůležitějších smyslů orientace a hladký dotykový displej pro ně tak může představovat zásadní bariéru. Na světě je přitom podle Světové zdravotnické organizace (angl. zkratka WHO) 285 miliónů zrakově postižených a z toho 39 miliónů je zcela nevidomých.[3] V současné době existuje několik telefonů s operačním systémem, které jsou přizpůsobeny potřebám zrakově postižených, ne vždy je však jejich používání pro uživatele příjemné nebo jejich používání vyžaduje pokročilé schopnosti uživatele. Velká část nevidomých tak využívá běžné tlačítkové telefony značky Nokia, které lze přizpůsobit pro ovládání zrakově postiženými. Telefony však nedisponují operačním systémem a jejich výroba je již ukončena - není u nich tedy vidina dalšího vývoje v budoucnosti, stejně tak jako podpora stávajícím uživatelům. Vývojem aplikace pro usnadnění používání chytrého telefonu s operačním systémem Android se ve své diplomové práci zabýval Petr Svobodník, který realizoval aplikaci pro zpřístupnění systému Android nevidomým.[4] Aplikace pokrývá běžné funkce telefonu a je navíc neustále rozšiřována o možnosti používat aplikace a funkce, které jsou na chytrých telefonech standardně využívány.



Obrázek 1 Procento dospělých lidí, kteří vlastní chytrý telefon [1]

1.1 Cíl práce

Cílem této práce je implementace aplikace „Internetové rádio“ umožňující poslech internetových rádií jako součást již vzniklé mobilní aplikace pro nevidomé, která je zmíněná výše (dále jako „kmenová aplikace“). Aplikace Internetové rádio by měla rozšířit pole funkcionalit, které mohou nevidomí uživatelé na chytrém telefonu využívat. V zásadě se tak jedná o doplnění funkce, na kterou jsou uživatelé zvyklí z běžných telefonů s tím rozdílem, že přijímání rádia přes internet přináší značné benefity v podobě možnosti kategorizace stanic či výběru z velkého množství stanic bez ohledu na jejich vysílací lokalitu.

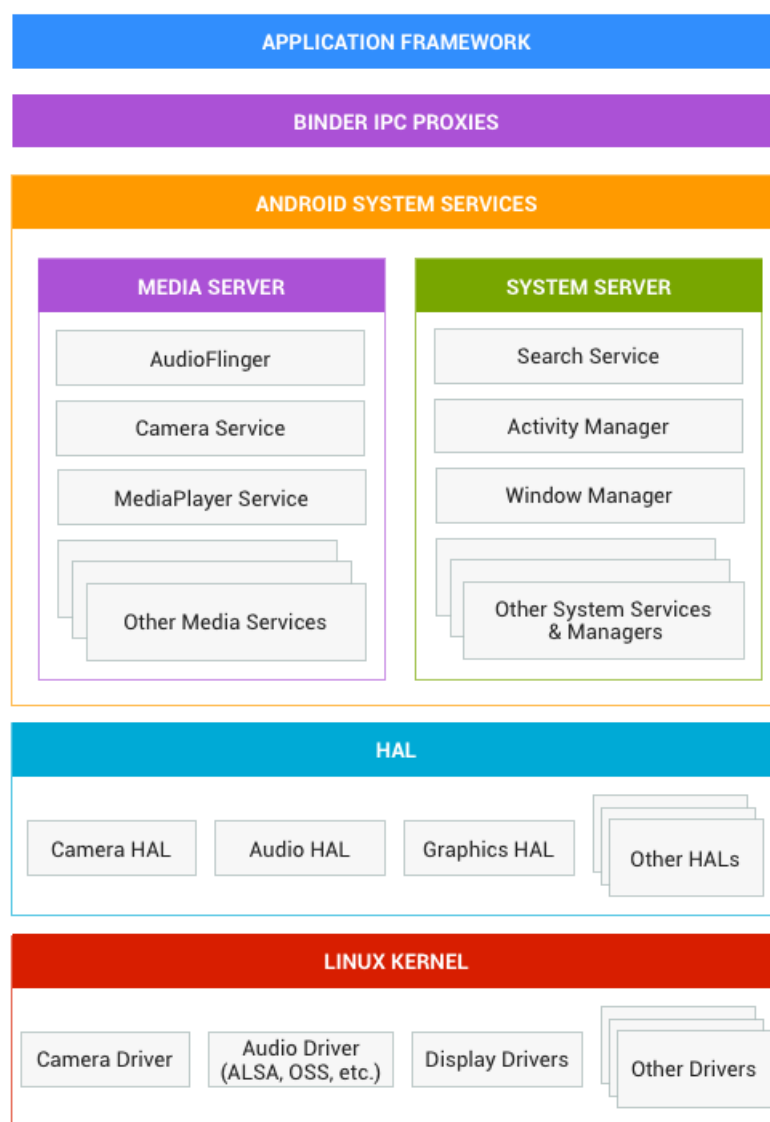
Aplikace Internetové rádio bude implementována v programovacím jazyce Java pro prostředí Android. Jelikož bude realizována jako součást již existující aplikace, její implementace zahrnuje především řešení rozložení uživatelského rozhraní a samotnou logiku fungování rádia, což obnáší získání a zpracovávání informací o rádiových stanicích společně se zpracováním jejich vysílacího streamu. Následně je také nutné vhodně prezentovat větší počet stanic a jejich výběr - to vše v uspořádání, které bude nevidomému uživateli umožňovat snadnou orientaci v aplikaci. Samotnou podobu uživatelského rozhraní pak řeší kmenová aplikace, která funguje na principu převodu textu na hlas (anglicky text-to-speech) a informuje tak uživatele, v jakém stavu se telefon nachází.

Závěrem práce by pak mělo být výsledné testování přímo nevidomými uživateli pomocí testu použitelnosti praktikovaném v tomu určeném prostředí, čímž je speciálně vybavená laboratoř použitelnosti (Usability lab), nacházející se v prostorách Fakulty elektrotechnické ČVUT na Karlově náměstí. To může odhalit drobné problémy a to, zdali je aplikace pro uživatele snadno použitelná.

2 Operační systém Android

Vzhledem k tomu, že aplikace Internetové rádio je vytvářena pro systém Android, je v této kapitole stručně popsána struktura a architektura tohoto operačního systému.[5]

2.1 Android architektura a interface



Obrázek 2 Architektura operačního systému Android [5]

2.1.1 Application framework

Vrstva Application framework je vývojáři nejvíce využívána, což je dáno především tím, že je pro ně nejdůležitější. Umožňuje přístup k mnoha službám, které mohou například zpřístupňovat data v jiných aplikacích, prvky uživatelského rozhraní či aplikace běžící na pozadí. [6]

2.1.2 Binder IPC

Binder Inter-Process Communication (IPC) je mechanismus pro předávání zpráv v komunikaci mezi procesy.

2.1.3 System services (Systémové služby)

System services jsou služby, které poskytují uživatelské aplikace s informacemi a schopnostmi nezbytnými pro fungování systému. Tyto služby mohou být také konkretizovány aplikacemi prostřednictvím Binderu. Android dělí tyto služby na dvě skupiny:

- system - sem spadají služby jako Windows Manager (správce oken), Notification Manager (správce upozornění)
- media - zahrnují přehrávání a zaznamenávání multimediálního obsahu

2.1.4 Hardware abstraction layer (HAL)

Hardware abstraction layer je vrstva, která definuje standardní interface pro dodavatele hardwaru a umožňuje tak Androidu být nezávislý na implementaci driverů nižších vrstev. To platí i naopak a HAL tímto zajišťuje funkcionalitu hardwaru bez nutnosti modifikace vyšších vrstev.

2.1.5 Linux kernel (Linuxové jádro)

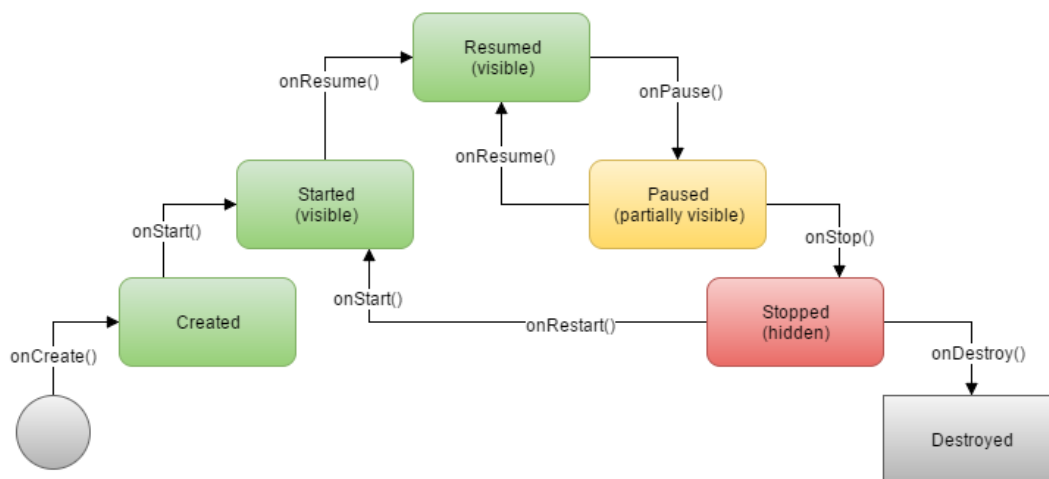
Nejnižší vrstvou je jádro operačního systému. To je v Androidu postaveno na jádře systému Linux, jenž obsahuje několik rozšíření, například právě Binder IPC driver.

2.2 Struktura Android aplikace

Základními částmi Android aplikace jsou komponenty *activity*, *service*, *content providers* a *broadcast receiver*. Všechny tyto komponenty (kromě *content providers*) spolu mohou komunikovat prostřednictvím systémových zpráv neboli záměrů - angl. *intent*. [6]

2.2.1 Activity (Aktivita)

Aktivita je entita systému Android, kterou lze přirovnat například k oknu klasické aplikace pro počítače. Z jedné aktivity je možné otevírat aktivity nové a rovněž se vracet do dříve otevřených aktivit. Je to entita s krátkým životním cyklem a lze ji kdykoliv ukončit.



Obrázek 3 Životní cyklus aktivity [6]

2.2.2 Service (Služba)

Služby na rozdíl od aktivit neposkytují uživatelské rozhraní a jsou navrženy k neustálému provozu, jsou tedy využívány k vykonávání dlouho trvajících úkolů, v případě potřeby nezávislé na aktivitách. Lze je využít například pro přehrávání hudby na pozadí, a to i bez nutnosti běhu řídicí aktivity. Kromě toho jsou také vhodné k provádění naplánovaných úkolů (cron jobs) a k zpřístupnění API jiným aplikacím nainstalovaným v zařízení.

2.2.3 Content providers (Poskytovatelé obsahu)

Content provideres je aplikační rozhraní sloužící pro sdílení dat mezi aplikacemi. Zajišťuje tak úroveň abstrakce pro jakákoliv data uložená v zařízení. Data mohou být uchovávána prostřednictvím webových kanálů, SQLite databází či v souborech.

2.2.4 Broadcast receiver

Tato komponenta slouží k naslouchání oznámení, na které pak reaguje (například spuštěním jiné komponenty). Oznámením může být například informace o stavu baterie.

2.2.5 Intent (záměr)

Záměry jsou systémové zprávy, které upozorňují aplikace na různé události. Těmi mohou být například přijetí SMS zprávy či spuštění nové aktivity. Na tyto záměry lze pak v aplikacích reagovat nějakou akcí.

3 Analýza existujících aplikací

Tato kapitola je věnována analýze již existujících aplikací poskytujících poslech internetových rádií. Aplikace zde nejsou řešeny z hlediska grafického uživatelského rozhraní - to totiž řeší již samotná kmenová aplikace, jehož bude vyvíjená aplikace Internetové rádio součástí. Analýza se tedy zaměřuje na funkcionalitu zkoumaných aplikací a na jejich logickou strukturu, která by mohla být prospěšná pro vyvození nejvhodnější struktury Internetového rádia pro nevidomé. Analyzované aplikace jsou volně dostupné prostřednictvím služby Google Play, která nabízí software pro operační systém Android.

3.1 TuneIn

Aplikaci TuneIn vytvořila společnost, kterou v roce 2002 založil pan Bill Moore. TuneIn nabízí k poslechu přes 100 000 rádiových stanic a patří tak mezi přední poskytovatele těchto služeb.[7] Používání TuneIn je možné přímo na webu nebo právě prostřednictvím aplikace pro chytré telefony, v tomto případě pro telefony se systémem Android.

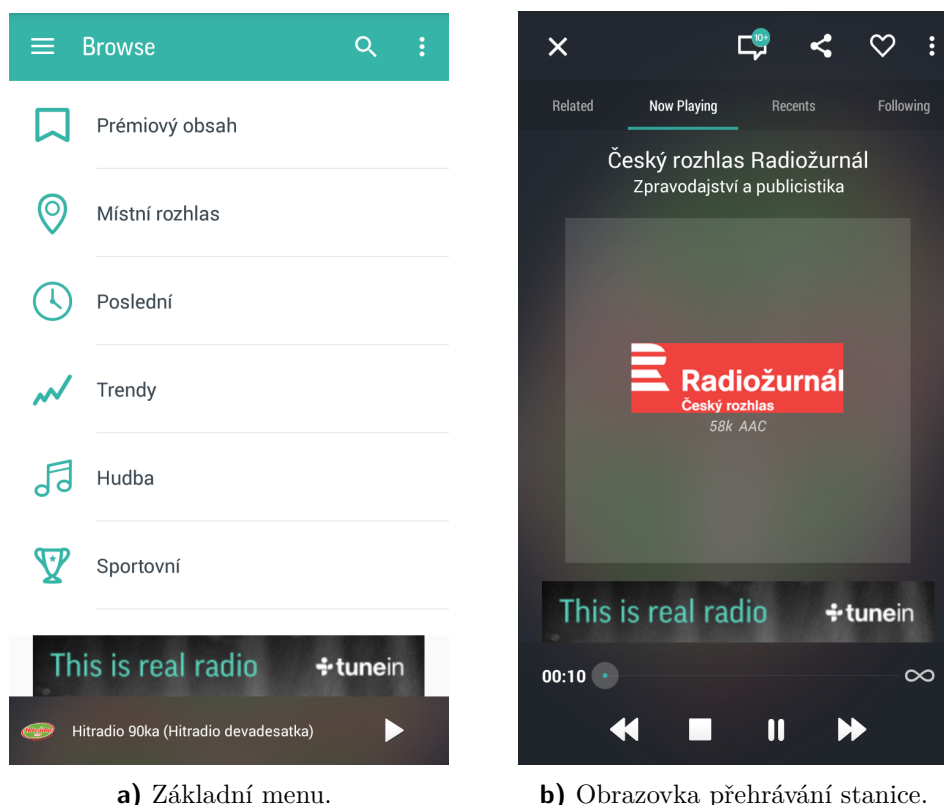
Po spuštění aplikace se zobrazí základní menu, které nabízí hlavní položky:

- Prémiový obsah
- Místní rozhlas
- Poslední
- Trendy
- Hudba
- Sportovní
- Novinky
- a další kategorie

Prémiový obsah dává uživatelům k dispozici např. živé vysílání ze sportovních událostí nebo vysílání stanic bez bloků reklam. Tato možnost je však dostupná pouze uživatelům s placeným účtem. Položka *Místní rozhlas* obsahuje stanice na základě určené lokality (země). Volba *Poslední*, jak už název napovídá, umožňuje uživateli vrátit se k posledním přehrávaným stanicím. *Trendy* zobrazuje populární stanice, ovšem pouze z aktuální lokalizace telefonu. *Hudba* nabízí seznamy stanic s hudebním obsahem. Tyto stanice jsou ještě dále rozděleny podle žánrů či hudebního období (např. 80. léta, blues, rock atd.). Další položky menu *Sportovní*, *Novinky* a další kategorie mají stejnou funkci, a to slučování stanic podobného vysílaného obsahu.

V seznamech stanic jsou zobrazeny nejhranější stanice daného seznamu, primárně na předních místech jsou umístěny lokální stanice. V seznamu je také možnost načíst více stanic, kdy uživatel dostane k dispozici nepřeborné množství dalších stanic. Volbou stanice vstoupí do okna přehrávání stanic, kde ovládá samotné přehrávání, může si zobrazit podobné stanice, uložit si stanici mezi *Oblíbené* stanice nebo sdílet stanici s

přáteli na sociálních sítích. K *Oblíbeným* stanicím se pak lze dostat přes záložku profilu uživatele v postranním menu aplikace.



Obrázek 4 Ukázka aplikace TuneIn.

3.2 XiiaLive

XiiaLive je aplikace, která se řadí mezi jedny z prvních aplikací, jež umožňovaly poslech internetových rádií.[8] Má podporu uberstations.com a shoutcast.com, což jsou webové platformy, které poskytují data pro streamování stanic. XiiaLive nabízí přes 50000 rádiových stanic produkujících hudební obsah i mluvené slovo. Nevýhodou je, že zde nejsou stanice, které běžně vysílají také na FM a AM (na rozdíl od TuneIn).

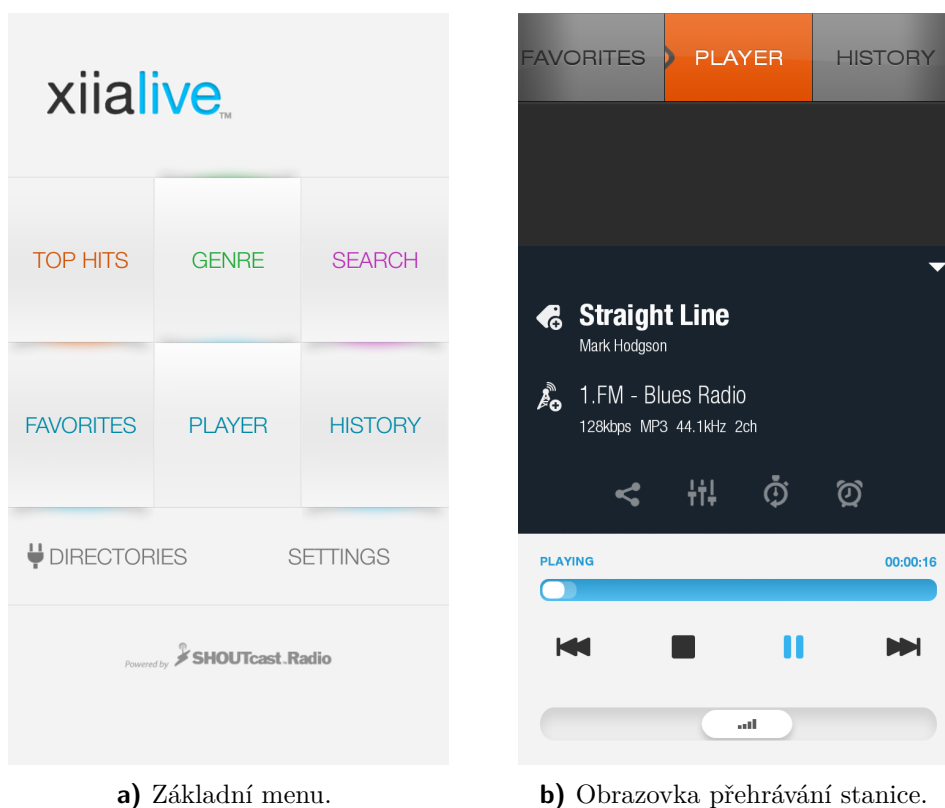
Základní obrazovka je tvořena dlaždicemi představujícími položky menu:

- Talk / Top hits (závisí, který zdroj stanic je vybrán)
- Music / Genre (závisí, který zdroj stanic je vybrán)
- Search
- Favorites
- Player
- History

dále je zde pak možnost výběru zdroje stanic (ShoutCast/Uberstations) a nastavení aplikace.

Význam položek v menu je obdobný, jako je tomu u TuneIn rádia, chybí zde však výběr lokálních stanic. Stanice jsou rozřazeny do kategorií, což umožňuje snadný výběr podle preferencí posluchače. Položka *Player* slouží k návratu na právě přehrávanou

(nebo naposledy přehrávanou) stanici a zobrazení informací o přehrávání. Tento přehrávač disponuje také jednoduchým ekvalizérem. V *History* se vytváří seznam stanic přehraných uživatelem. Vybrané stanice lze rovněž ukládat mezi „oblíbené“ (*Favorites*). Je zde také možnost nastavit si „buzení rádiem“, což v praxi znamená, že v uživatelem určený čas začne hrát vybraná stanice a aplikace tak zastoupí roli budíku, stejně tak jako tomu může být u běžných FM/AM přijímačů.



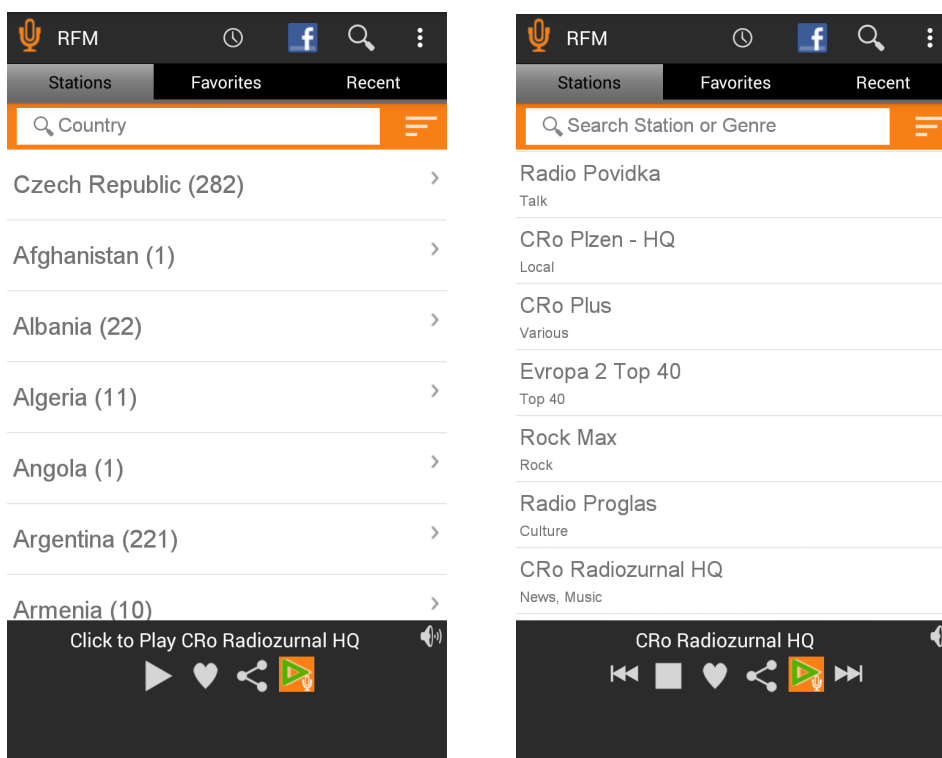
Obrázek 5 Ukázka aplikace XiiLive.

3.3 Radio FM

Aplikace Radio FM připomíná více klasické rádio - jsou zde pouze stanice, které vysílají prostřednictvím FM technologie, i tak jich je k dispozici přes 10 000 z různých zemí.[9] Oproti předchozím analyzovaným aplikacím zde chybí zobrazení stanic stejného žánru. Stanice jsou primárně rozděleny podle zemí, ve kterých vysílají. Po zvolení země se vypíší všechny dostupné stanice, což může při větším počtu uživateli znesnadnit jejich procházení. Ovládat přehrávání lze pouze v liště ve spodní části aplikace (není zde samostatné okno přehrávače), kromě informací o přehrávané stanici tento panel poskytuje také možnost uložení stanice mezi „oblíbené“ či její sdílení na sociálních sítích. Navíc je také možné přepínat stanice šipkami doleva a doprava, jak jsou tomu uživatelé zvyklí například z hudebních přehrávačů. Celkově je struktura aplikace i uživatelské rozhraní velmi jednoduché. Stěžejní jsou tři záložky:

- Stations
- Favorites
- Recent

Dále je zde také možnost vyhledávání ve stanicích nebo státech. Položka *Recent* obsahuje seznam posledních přehrávaných stanic.



a) Obrazovka pro výběr země.

b) Seznam stanic dané země a přehrávání stanice.

Obrázek 6 Ukázka aplikace Radio FM.

3.4 RadioTunes

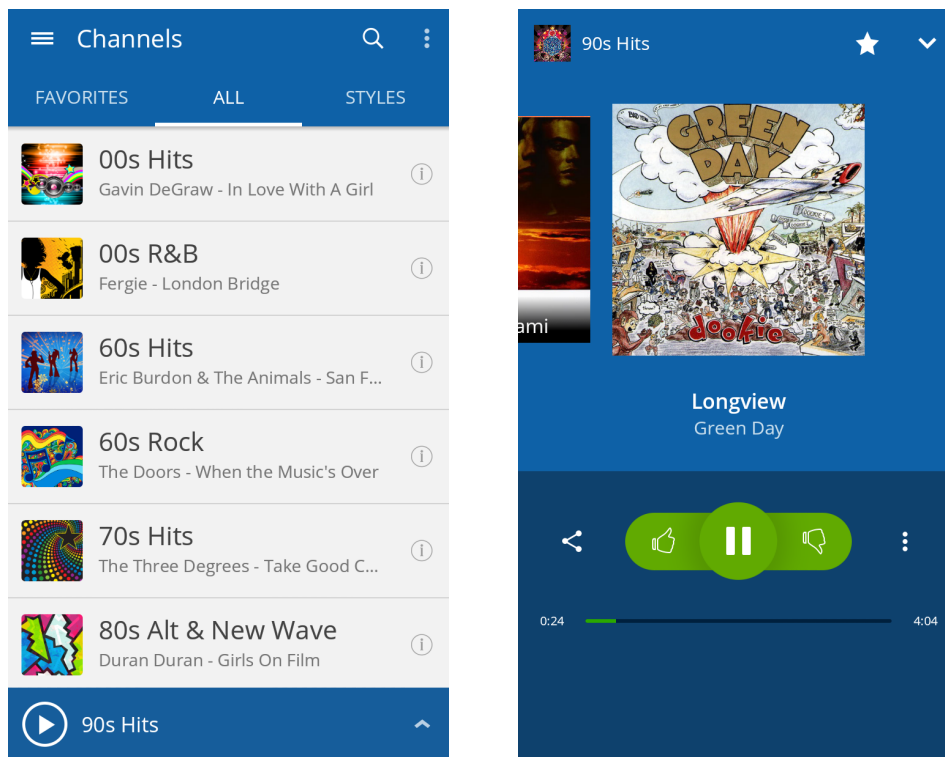
RadioTunes je trochu jiný typ aplikace oproti všem předchozím, znatelně zaostává v počtu nabízených stanic, kterých je jen přes 80. Velkou protiváhou tomu je však jejich kvalita. Nejedná se totiž o běžné stanice, jež vysílají také prostřednictvím FM, ale o stanice vytvářené jednotlivými správci kanálů. Ti, jak tvrdí poskytovatel, by měli být experty v daném hudebním žánru, a tak by skladby přehrávané na dané stanici měly splňovat kritéria žánru, do kterého je stanice zařazena.[9] Ovládání aplikace je velmi snadné, základem jsou tři záložky:

- Favorites
- All
- Styles

Jak už názvy napovídají, *Favorites* zobrazí seznam oblíbených stanic, *All* slouží k výpisu všech dostupných stanic a *Styles* otevře nabídku žánrů, do kterých jsou stanice rozřazeny. Výhodou je, že už samotné názvy stanic určují, jakého hudebního žánru stanice je, což vyplývá ze samotné filosofie aplikace. Po spuštění stanice se otevře obrazovka přehrávače, na které lze kromě standardních ovládacích prvků přehrávání nalézt také informace o aktuálně přehrávané skladbě včetně přebalu alba, na kterém vyšla. Tyto

3 Analýza existujících aplikací

informace jsou rovněž dostupné o již odehraných skladbách rádia. Kromě možnosti přidat stanici do oblíbených má uživatel k dispozici možnost přidat si právě přehrávanou skladbu do seznamu oblíbených skladeb. Samotnou obrazovku přehrávání lze minimalizovat do formy panelu ve spodní části obrazovky - během procházení aplikace je tak snadné ovládat přehrávání nebo se rychle vrátit na obrazovku přehrávače.



a) Výběr ze všech stanic.

b) Obrazovka přehrávání.

Obrázek 7 Ukázka aplikace RadioTunes.

3.5 Poznatky pro návrh aplikace Internetové rádio

Z analýzy populárních aplikací pro přehrávání internetových rádií je k dispozici několik poznatků týkající se struktury a funkcí, které by neměly být opomenuty při vývoji aplikace Internetové rádio. Důležitou funkcí je možnost ukládat si oblíbené stanice. Mít seznam oblíbených věcí je běžnou praxí v mnoha jiných typech aplikací a uživatelům tato funkce usnadňuje a zpříjemňuje používání aplikace. Podstatné je také nezahltit uživatele stanicemi, kterými by musel dlouho listovat. Pro nevidomého člověka je to nepřijemné i z důvodu, že musí čekat na zvukovou odezvu systému. Užitečné je rovněž rozřazení stanic podle kategorií, které může vyhledávání značně urychlit a navíc dává uživateli prostor výběru stanic podle nálady či hudebního vkusu. Neměl by chybět také návrat k poslední přehrávané stanici.

Aplikace bude moci poskytovat pouze stanice, které vysílají prostřednictvím internetu. Použití technologie FM pomocí hardwaru telefonu není možné z důvodu absence oficiálního API. Řešení tzv. „hrubou silou“ bez použití API by bylo možné pouze na některých typech telefonů, a to kvůli rozdílným FM čipům. Tím by však aplikace ztratila svoji univerzálnost.

4 Návrh a analýza řešení

Následující kapitola je věnována návrhu řešení aplikace Internetového rádia pro nevidomé, která bude součástí již fungující kmenové aplikace. Návrh je zaměřen na logiku fungování aplikace, její strukturu a funkce, které by měla mít. Klíčovým bodem je získávání dat, prostřednictvím nichž lze stanice přehrávat. To je možné pomocí url adresy, na které stanice streamují své vysílání. Metoda streamování medií umožňuje doručit obsah uživateli v reálném čase.[10] Zjednodušeně řečeno se jedná o podobnou technologii jako při zobrazování webových stránek s tím rozdílem, že webové servery *stahují* soubory, zatímco media servery soubory *streamují*. V praxi to znamená, že v prvním případě jsou stažené soubory zobrazeny až po doručení všech souborů, kdežto v případě *streamování* jsou soubory přehrány hned po jejich doručení, což je vhodné pro živé vysílání rádií.

4.1 Získávání dat pro přehrávání stanic

Jak již bylo řečeno, důležitou roli při návrhu fungování aplikace hraje zdroj, ze kterého budou čerpány informace a data nutná k přehrávání stanic. Těmi jsou údaje o jazykové lokaci stanice, jméno stanice a především url adresa, na které stanice provozuje stream svého vysílání. Na internetu lze nalézt webové aplikace, které mají databáze stanic, avšak pro účely vyvíjené aplikace je nejpodstatnější, aby měly volně dostupné API (Application Programming Interface), prostřednictvím něhož by bylo možné získat z databáze informace o stanicích. Mít tato data pevně zanesená v aplikaci by totiž bylo značně nepraktické a neefektivní, a to vzhledem k množství stanic i tomu, že streamovací url adresy se mohou s časem měnit.

Velmi populární je služba Shoutcast, kterou využívají některé z analyzovaných aplikací z kapitoly 3. Ta na svém webu umožňuje přehrávat nepřeborné množství stanic rozdělených podle žánrů. Zároveň poskytuje vývojářské API, jehož prostřednictvím lze přistupovat k této rozsáhlé databázi, jež čítá přes 59 000 stanic.[11] K používání tohoto API je však zapotřebí klíč (identifikační číslo), který je možné získat po zažádání o partnerství se společností Shoutcast. Toto partnerství se bohužel nepodařilo navázat, a tak není možné toto API využít.

Vhodnou alternativou je obdobná služba s názvem Dirble.[11] Ta se od Shoutcast liší v podstatě jen množstvím stanic v databázi (přibližně 11 500), jinak je fungování Dirble podobné, což znamená, že stanice lze přehrávat přes webové stránky a zároveň je poskytováno vývojářské API. Klíč (v tomto případě nazýván *token*) k jeho používání je možné získat při registraci a založení účtu. Výhodou této služby je, že stanice jsou rozděleny podle zemí a rovněž do hudebních kategorií. Nevýhodou je malý počet českých stanic, což znamená, že pro českou lokalizaci není tato varianta vhodná, nicméně pro ostatní jazykové mutace aplikace je dobře použitelná.

Podobnou službou v České republice je Play.cz - také lze prostřednictvím jejích webových stránek poslouchat stanice internetových rádií a navíc je zde velký výběr českých

stanic, ať už těch čistě internetových nebo i těch, které vysílají prostřednictvím FM technologie. Stanice jsou rovněž rozděleny podle kategorií. Důležitým bodem je, že služba poskytuje vývojářské API, které je možné po domluvě používat. Tato domluva byla provedena prostřednictvím emailové komunikace se společností.

Pro získání dat pro přehrávání stanic budou tedy použity služby Dirble a Play.cz. Play.cz pro získání informací a přehrávacích dat českých stanic, Dirble pak pro další jazykové mutace, ve kterých bude aplikace používána (aktuálně německá a anglická verze).

4.1.1 Dirble

API služby Dirble poskytuje informace o stanicích v podobě REST webových služeb. REST navrhl a popsal v roce 2000 Roy Fielding v rámci disertační práce *Architectural Styles and the Design of Network-based Software Architectures*. [12] Jedná se o architekturu rozhraní, která slouží pro jednotný a snadný přístup ke zdrojovým datům. Přístup REST využívá metody protokolu HTTP jako POST, GET atd., pomocí nichž předává data službě. Zdrojová data pak mohou mít různou reprezentaci (např. XML, HTML, JSON), se kterou se pracuje. Requesty (dotazy) na Dirble API vracejí zdrojová data reprezentována ve formátu JSON (JavaScript Object Notation). Takto reprezentovaná data mohou být organizována v polích nebo agregována v objektech.

Přehledná dokumentace [11] k Dirble API na stránkách služby popisuje, jak API používat - tedy seznam HTTP requestů, které lze použít pro získání zdrojových dat. Pro účely této aplikace je nejdůležitější HTTP požadavek pro získání stanic z dané země (*your token* zastupuje číslo tokenu - klíče, *<code>* je kód požadované země):

```
http://api.dirble.com/v2/countries/<code>/stations?  
token={your token}
```

Tento request vrátí JSON pole objektů (vizte Obrázek 8) se všemi důležitými informacemi o stanici jako jsou:

- id
- jméno stanice
- země
- pole objektů kategorií, do kterých rádio spadá
- pole objektů streamů obsahující streamovací url adresy rádií

Nutno podotknout, že získané JSON pole objektů obsahuje maximálně 30 stanic. Jak se uvádí v dokumentaci, je to z důvodu snížení rizika zatížení serverů v případě, že by měly poskytovat mnoho dat najednou. Pro získání více stanic je tedy nutné použít *stránkování*. To znamená dotazovat se na data requestem s parametry *číslem stránky (page)* a *počtem položek na stránku (per_page)*. Například:

```
http://api.dirble.com/v2/stations?page=1&per_page=10&  
token={your token}
```

zobrazí 10.-20. položku z databáze.

```
[
  {
    "id":36589,
    "name":"STAR FM",
    "country":"DE",
    "categories":[
      {
        "id":2,
        "title":"Rock",
        "description":"simple rock. from elvis to metallica
          and like hardrock as iron maiden.",
        "slug":"rock",
        "ancestry":null
      }
    ],
    "streams":[
      {
        "stream":"http://85.25.209.150:8400",
        "bitrate":64,
        "content_type":"audio/aacp",
        "status":1
      },
      {
        "stream":"http://85.25.209.150:8400",
        "bitrate":64,
        "content_type":"audio/aacp",
        "status":1
      }
    ],
    ...
  },
  ...
]
```

Obrázek 8 Část odpovědi API ve formátu JSON na HTTP request (všechny německé stanice).

4.1.2 Play.cz

API Play.cz poskytuje rovněž informace o stanicích v podobě REST webových služeb a stejně tak na requesty vrací zdrojová data reprezentována ve formátu JSON. Jak je popsáno v dokumentaci služby Play.cz [13], dotaz do API se provádí pomocí HTTP GET requestu na adresu:

```
http://api.play.cz/<FORMAT>/<FUNKCE>/<...params...>
```

kde *<FORMAT>* je požadovaný formát odpovědi (lze si vybrat mezi XML a JSON), *<FUNKCE>* je název volané funkce a *<...params...>* jsou parametry volané funkce.

Funkce API jsou uvedeny v dokumentaci s detailním popisem parametrů a návratových hodnot. Pro účely implementace této aplikace budou zapotřebí funkce *getRadios* a *getStreamMobile*.

Metoda *getRadios* vrací seznam všech rádií v databázi jako JSON objekt s dalšími objekty. Upravena může být parametry *styl* a *kraj*, které vyfiltrují stanice daného stylu nebo stanice z určitého kraje. Odpověď obsahuje informace jako název rádia, popis rádia, pole stylů, do kterých rádio spadá, a také řetězec *shortcut*, který slouží jako jednoznačný identifikátor rádia. V odpovědi však není streamovací url adresa rádia, tu je třeba zjistit na základě řetězce *shortcut* pomocí funkce *getStreamMobile*.

```
{
  "data": {
    "playuk40": {
      "title": "PLAY.CZ UK TOP 40",
      "description": "Top 40 singl",
      "shortcut": "playuk40",
      "listeners": "23",
      "cooluri": "http://www.play.cz/playuk40",
      "onair": "true",
      "ordering": "9999",
      "style": [
        "pop",
        "hotac"
      ],
      "style_title": [
        "Pop",
        "Hot AC"
      ],
      "region": [
        "internet"
      ],
      "region_title": [
        "Internet"
      ]
    },
    ...
  }
}
```

Obrázek 9 Část odpovědi API ve formátu JSON na HTTP request *getRadios*.

Metoda `getStreamMobile` slouží k získání streamu konkrétní stanice ve formátu, který je vhodný pro přehrávání na mobilních zařízeních (např. mp3). Povinným parametrem je zde `shortcut` stanice, jenž jednoznačně určí, o kterou stanici se jedná. Volitelnými parametry jsou `format` (určí požadovaný formát streamu) a `bitrate` (pro požadovaný bitrate). Návrátovými hodnotami jsou informace o streamech, kde `pubpoint` označuje přímou url adresu streamu.

```
{
  "plain":{
    "content-type":"audio/x-mpegurl; charset=utf-8",
    "content":"#EXTM3U\n#EXT-X-VERSION:3\nhttp://
    /\ icecast2.play.cz /\ beat128aac\nhttp://\
    icecast2.play.cz /\ beat32aac\n#EXT-X-ENDLIST"
  },
  "redir":"http://\ icecast2.play.cz /\ beat128aac",
  "data":{
    "stream":{
      "id":"492",
      "shortcut":"beat",
      "pubpoint":"http://\ icecast2.play.cz /\ beat128aac",
      "playlist":"beat128.aac.m3u",
      "stream_type":"aac",
      "stream_bitrate":"128",
      "stream_title":null,
      "fallback_to_id":"0",
      "isActive":"1",
      "isOnWeb":"0",
      "isOnPhone":"1",
      "listeners_max":"10000",
      "listeners":"25",
      "listeners_delta":"0",
      "listeners_delta_p":"0",
      "playlist_generated":"1463000876",
      "last_update":"1463000876",
      "disabled_until":"1462999570",
      "listeners_rate":"0"
    },
    ...
  }
}
```

Obrázek 10 Část odpovědi API ve formátu JSON na HTTP request `getStreamMobile` s parametrem `shortcut = beat`.

4.2 Funkce a klíčové vlastnosti aplikace

Další částí návrhu je rozvaha, co by měla aplikace Internetové rádio umět a uživateli nabízet. Dle analýzy již existujících aplikací je stěžejní funkcí možnost uložit si stanici do „oblíbených“, což umožňuje snadnější výběr preferovaných stanic. Tím, že je aplikace

vyvíjena pro nevidomé, je znesnadněno předávání velkého množství informací tak, aby se uživatel v obsahu aplikace stále orientoval. I přesto by měly být zachovány hlavní výhody příjmu rádia přes internet. Těmi jsou nabídka dostatečného množství stanic a jejich řazení do hudebních kategorií. Jako dostatečný počet stanic se nabízí 90 stanic ± 10 stanic. Číslo bylo zvoleno na základě stanic poskytovaných API Play.cz, které má momentálně k dispozici 94 stanic a dle vyzkoušení v praxi na testovacím seznamu v kmenové aplikaci, který čítal 90 položek. Došlo tak k ověření, že jejich procházení je ještě pro uživatele možné. S každou další desítkou položek by bylo toto procházení složitější. Zmíněný počet zároveň nabízí uspokojivé množství stanic se zastoupením v mnoha kategoriích. Procházení položek by také měla usnadnit možnost vyhledávání, která bude v seznamu stanic dostupná, pokud tento seznam bude obsahovat více jak 15 položek.

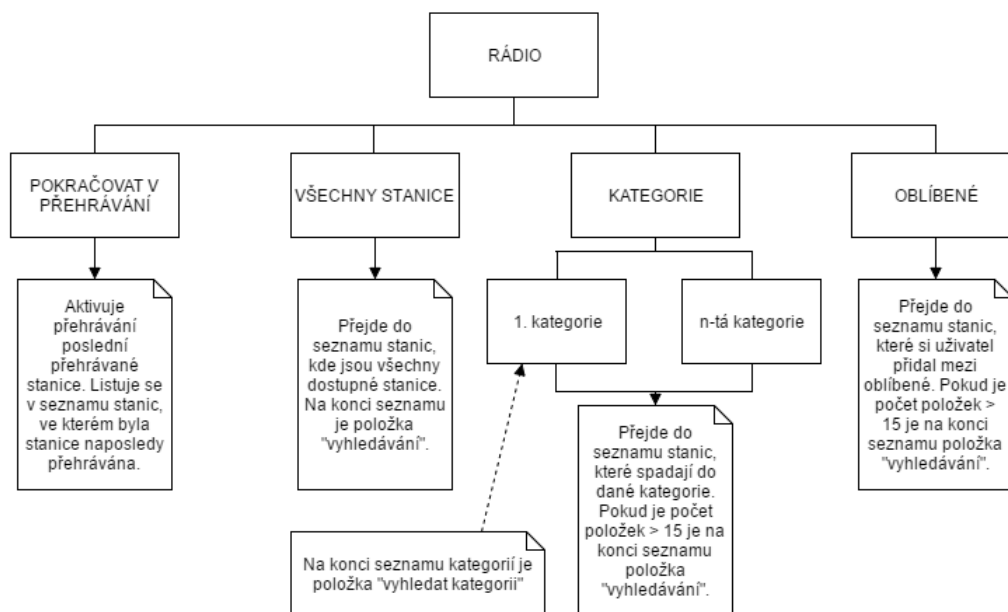
Důležitým prvkem je také samotné přehrávání - uživatel by vždy měl mít možnost vrátit se k poslední přehrávané stanici nebo seznamu stanic.

Návrh aplikace byl konzultován s expertními nevidomými uživateli ze společnosti SONS. SONS, neboli Sjednocená organizace nevidomých a slabozrakých České republiky je spolek s celostátní působností, jenž vznikl v roce 1996 sloučením obou do té doby celostátně působících občanských sdružení občanů s těžkým zrakovým postižením - České unie nevidomých a slabozrakých a Společnosti nevidomých a slabozrakých v ČR. Jak stojí na webových stránkách organizace: „Posláním naší organizace SONS je jednak sdružovat nevidomé a jinak těžce zrakově postižené občany a hájit jejich zájmy, jednak poskytovat konkrétní služby vedoucí k integraci takto postižených občanů do společnosti.“ [14] Tito uživatelé pravidelně testují kmenovou aplikaci během vývoje, a tak mohou poskytnout cenné rady v oblasti návrhu nové aplikace, která rozšíří její funkcionalitu.

Hlavním doporučením od uživatelů ze SONS bylo, aby aplikace byla jednoduchá a účelná - jedná-li se o aplikaci Internetové rádio, je nejpodstatnější, aby se uživatel dostal co nejnázem k požadované stanici a spustil její přehrávání. Není nutné uživatele zahlcovat informacemi či funkcemi, které nevyužije. Vývoj aplikace má být navíc takový, aby byla snadno rozšiřitelná. To znamená, že pokud přijde od koncových uživatelů podnětný návrh na rozšíření o určitou funkcionalitu, nebude problém ji do aplikace implementovat. Důležité také je, aby rozšiřující aplikace respektovala již zažitá postupy kmenové aplikace. Je to například posloupnost položek v menu či gesta pro ovládání aplikace. Proto budou také pro ovládání přehrávání rádia použita stejná gesta, jako tomu je u již existující aplikace hudebního přehrávače (krátký dotek dvěma prsty = pozastavení/spuštění přehrávání atd.).

4.3 Struktura aplikace

Jak již bylo dříve zmíněno, pro nevidomého uživatele je nejdůležitější, aby se v aplikaci dobře orientoval. Měl by tedy vždy vědět, kde se ve struktuře aplikace nachází a jaký je její stav. Struktura by zároveň neměla být příliš složitá, aby její procházení nezabralo velké množství času. Tato problematika byla rovněž konzultována s experty ze společnosti SONS a finální návrh rozložení aplikace je vidět ve stromové struktuře na Obrázku 11.



Obrázek 11 Navrhované struktura aplikace Internetové rádio.

4.4 Shrnutí návrhu

Pro fungování aplikace Internetové rádio je zásadní návrh zajištění získávání streamovacích adres rádií. To bude v případě české lokalizace řešeno prostřednictvím služby Play.cz, pro ostatní jazykové mutace bude použita služba Dirble. Vývojářská API obou služeb jsou poskytována v podobě REST webových služeb a návratová data jsou ve formátu JSON. Aplikace bude nabízet výpis všech stanic, řazení stanic podle kategorií a ukládání stanic mezi oblíbené. Seznamy, které budou obsahovat více jak patnáct položek, budou mít k dispozici položku *vyhledávání*. Uživatel bude mít také možnost vrátit se k poslední přehrávané stanici a seznamu stanic, ze kterého ke stanici dříve přistoupil.

(*compareTo(Station another)*), která umožňuje porovnávat objekty stanic podle jejich jména. Tato metoda je využívána v jiných třídách pro srovnání kolekce stanic abecedně podle jejich názvů.

5.2 Uchovávání dat - třída *RadioData*

Pro uchovávání potřebných dat při běhu aplikace slouží třída *RadioData*, která dědí od *Application* (z *Android API*). Dle *Android dokumentace* [15] je vhodné třídu s takovýmto významem implementovat jako singleton. Singleton (neboli „jedináček“) je návrhový vzor, který zajišťuje, aby v celém programu běžela pouze jedna instance dané třídy. Tato třída tak umožňuje jednoduchý přístup k datům, které uchovává, z jakékoliv jiné části aplikace prostřednictvím přístupu k její instanci přes *RadioData.getInstance()*. Obsahuje několik kolekcí *List<Station>*:

- *localStationsList* - načtené stanice pro danou jazykovou lokaci
- *selectedStationsList* - stanice, které byly vybrány pro zobrazení v seznamu
- *playlist* - aktuálně přehrávaný seznam stanic
- *favourites* - seznam oblíbených stanic

Další kolekcí uchovávající data je pak *HashMap<String>, ArrayList<Station>> categoriesMap*, kde klíč je *String*, který reprezentuje název kategorie, a hodnota je *ArrayList* stanic, ve kterém jsou stanice dané kategorie. Třída *RadioData* slouží především také k držení informací o stavu aplikace, respektive o stavu přehrávání. Jsou to informace, jaká stanice je momentálně vybrána (*selectedStation*), v jakém jazyce je telefon (*language*), zdali se uživatel dostal do seznamu stanic přes položku menu „oblíbené stanice“ (*favouritesView*) nebo jestli je potřeba synchronizovat oblíbené stanice s předchozími uloženými oblíbenými stanicemi (*needSync*). V dalších kapitolách bude popsáno v jakých situacích a jak jsou tato data a funkce využity.

5.3 API Service

Hlavní funkcionalitu získávání seznamů stanic internetových rádií, včetně potřebných údajů o nich, zajišťují třídy *ServicePlayAPI* (pro české stanice) a *ServiceDirbleAPI* (pro stanice ostatních jazykových verzí). Pro odesílání HTTP requestů je využita knihovna *Volley*. [15] Jelikož jsou zdrojová data získávána asynchronně, je zde použit návrhový vzor *Observer*, kdy *ApiListner* předá získaná data třídě *RadioData* až ve chvíli, kdy jsou kompletně doručena.

5.3.1 Knihovna *Volley*

Volley je otevřená knihovna, která je v *Android dokumentaci* doporučována pro práci s HTTP dotazy (requesty). Tato knihovna umožňuje snadné obsluhování HTTP dotazů, nabízí automatické plánování dotazů či jejich prioritizaci. Její použití využívá podle dokumentace [15] frontu dotazů (*RequestQueue*), která je realizována jako singleton - v aplikaci *Internetové rádio* jako třída *VolleySingleton*. Do této fronty jsou přidávány HTTP dotazy, které jsou následně prováděny. Vytvoření dotazů a zpracování zdrojových dat je řešeno ve třídách *ServicePlayAPI* a *ServiceDirbleAPI*. Příslušné dotazy jsou pak inicializovány a přidány do fronty v aplikaci na místě, kde je potřeba pomocí nich získat daná data.

5.3.2 ApiListener

Třída `ApiListener` je součástí návrhového vzoru `Observer` a slouží k tomu, aby zdrojová data byla zpracovávána až poté, co jsou doručena. Její metody jsou volány ve třídách `ServicePlayAPI` a `ServiceDirbleAPI`. Tyto metody předávají získaná data (stanice) do třídy `RadioData`, odkud jsou pak dále v aplikaci používána.

5.3.3 ServicePlayAPI

`ServicePlayAPI` je třída, pomocí níž jsou získávány české internetové rádio stanice. Slouží k vytvoření HTTP dotazu a ke zpracování zdrojových dat, která dostane jako odpověď na tento dotaz. To je realizováno ve dvou metodách - `getAllStation()` slouží pro získání všech stanic, `getStationUrl(final String shortcut)` vyhledá url adresu streamu dané stanice. Další metodou je `setListener(ApiListenerInterface listener)`, která nastaví proměnou `listener` instancí třídy.

Jak již bylo řečeno, API služby `Play.cz` vrací zdrojová data ve formátu JSON - v metodě `getAllStation()` je tedy nutné vytvořit objekt `JsonObjectRequest`, který je pak návratovou hodnotou metody. Při vytváření `JsonObjectRequest` je nutné předat následující parametry:

- typ dotazu (v tomto případě GET)
- url dotazu - „http://api.play.cz/json/getRadios“
- `new Response.Listener<JSONObject>`, ve kterém je pomocí anotace `@Override` u metody `onResponse` definováno vlastní chování v případě přijetí odpovědi
- `new Response.ErrorListener()`, kde je rovněž pomocí anotace `@Override` u metody `onErrorResponse` nadefinováno chování, pokud se přijetí odpovědi nezdaří.

V metodě `onResponse` probíhá zpracování přijatých zdrojových dat. Ty přijdou jako `JSONObject` pojmenovaný "data", ve kterém jsou další JSON objekty (vizte Obrázek 9 v kapitole 4.1.2). Každý tento objekt reprezentuje jednu stanici. Je tedy nutné těmito objekty iterovat a vybírat z nich potřebná data. Během iterace je získaný objekt, reprezentující stanici, uložen do lokální proměnné `stationInfo` typu `JSONObject`. Z něho jsou poté pomocí metody `stationInfo.getString(<param>)` získávány a ukládány do lokálních proměnných potřebné textové informace jako název stanice a zkratka (`shortcut`), která slouží k identifikování stanice. Parametr `<param>` metody `getString` je označení příslušného požadovaného řetězce (např. "title"). Informace o hudebních stylech, do kterých stanice spadá, jsou poskytovány v poli typu JSON, tedy `JSONArray`. Tímto polem je iterováno a názvy stylů jsou ukládány do lokálního `ArrayList` `String` objektů s názvem `styles`. Pokud je přijaté JSON pole prázdné, je do listu `styles` přidána pouze jedna položka označující, že stanice spadá do kategorie „Unknown“ („Neznámé“). Následně jsou získané informace předány do konstruktoru třídy `Station`, který vytvoří lokální objekt `station`.

```
Station station = new Station(stationInfo.getString("title"),
    stationInfo.getString("shortcut"), "cs", styles);
```

Proběhne také kontrola, zdali stanice nebyla již dříve uložena mezi „oblíbené“. Kontrola je provedena pomocí listu `favourites` z třídy `RadioData`, kde se porovnávají parametry `shortcut` stanic z listu `favourites` a nově vytvořené stanice `station`. Pokud dojde k nálezů shody zkratky stanice z listu a nově vytvořené stanice, je této stanici nastaven parametr `favourite` na `true`.

Následně je do listu lokální proměnné *stationList* typu `ArrayList<Station>` přidána nově vytvořená stanice *station*, která je také přidána do jednotlivých `ArrayListů` v lokální mapě `Map<String, ArrayList<Station>>` s názvem *categories*. Přidání probíhá na základě stylů, do kterých stanice spadá a jež jsou uloženy v listu *styles*. Tyto styly jsou porovnány s klíči reprezentující jednotlivé kategorie v mapě *categories*. Pokud se klíč a styl stanice shodují, je stanice přidána do `ArrayListu` svázaného s daným klíčem. V případě, že styl není nalezen mezi klíči, je přidán jako nový klíč a stanice přidána do `ArrayListu` tohoto klíče. V dalším kroku jsou volány metody `ApiListeneru`. Konkrétně metoda `setLocalStations(stationList)`, která předá *stationList* se získanými stanicemi do listu *localStationsList* třídy `RadioData` a dále pak metoda `setCategoriesMap(categories)`, která do proměnné *categoriesMap* třídy `RadioData` předá mapu, kde klíči jsou názvy kategorií a hodnotami jsou `ArrayListy` stanic spadajících do dané kategorie.

Po použití metody `getAllStation()` je tedy k dispozici prostřednictvím třídy `RadioData` list stanic (*localStationsList*) a mapa kategorií s listy příslušných stanic (*categoriesMap*). Pro přehrávání stanice je však nutné zjistit url jejího streamu. To je realizováno prostřednictvím metody `getStationUrl(final String shortcut)`, do které jako parametr vstupuje *shortcut* stanice. Metoda pak obdobně jako `getAllStation()` vytvoří `JsonObjectRequest` se stejnými parametry s výjimkou url adresy dotazu. Ta je vždy „`http://api.play.cz/json/getStreamMobile/`“, následně je za ní přisloučen řetězec "*shortcut*" a také "`/mp3`", což udává preferovaný formát streamu (ne vždy je ale formát mp3 dostupný, častým formátem je také aac). Opět je zde pomocí `@Override` definována metoda `onResponse` `Response` listeneru. Zdrojová data jsou rovněž reprezentována `JSON` objektem, který obsahuje další objekty (vizte Obrázek 10 v kapitole 4.1.2). Je tedy nutné projít `JSON` objekt „*data*“ a vybrat objekt "*stream*", který obsahuje všechny dostupné informace o streamu stanice. Pro aplikaci je potřebné url streamu, které je pod označením "*pubpoint*" a formát streamu pod označením "*stream_type*". Oba tyto údaje jsou uloženy do lokálních proměnných typu `String` a následně prostřednictvím metody `setUrlAndType` `API Listeneru` přiřazeny dané stanici.

Metoda `getStationUrl(final String shortcut)`, jejímž prostřednictvím lze předat `HTTP` dotaz na vysílací stream url stanice do fronty dotazů (`RequestQueue`), je v aplikaci volána až před samotným spuštěním přehrávání stanice.

5.3.4 ServiceDirbleAPI

Třída `ServiceDirbleAPI` slouží stejně jako třída `ServicePlayAPI` k vytvoření `HTTP` dotazu a zpracování zdrojových dat. Dotazuje se prostřednictvím `API` služby `Dirble`, která je v aplikaci internetového rádia použita pro získání stanic, pokud je aplikace v jiné než české jazykové mutaci. Na rozdíl od `ServicePlayAPI` stačí k získání všech stanic včetně jejich streamovacího url vytvořit pouze jeden `HTTP` dotaz. Na druhou stranu tento dotaz vrátí maximálně 30 stanic, je tedy třeba využít stránkování a těchto dotazů vytvořit více.

Pro získání všech stanic a informací o nich je ve třídě `ServiceDirbleAPI` implementována metoda `getAllStations(final String country, int page)`. Parametrem *country* je kód země, na jejíž stanice se bude metoda dotazovat, *page* je číslo stránky stanic pro zobrazení. V url adrese dotazu, která je v metodě implicitně zadána, je nastaven největší možný počet stanic k zobrazení, a to třicet.

V metodě dojde k vytvoření `JSONArrayRequest`, který metoda následně vrátí. U nového `JSONArrayRequest` je nutné nastavit stejné parametry, jak je již popsáno v kapitole 5.3.3. Do url adresy dotazu jsou přidány parametry metody `getAllStations` - `country` (kód země) a `page` (stránka k zobrazení). Při inicializování nového `JSONArrayRequest` je rovněž vytvořen objekt `Response.Listener<JSONArray>()`, ve kterém je specifikováno chování metody `onResponse`. V ní je zpracováváno přijaté JSON pole JSON objektů, které reprezentují jednotlivé stanice. Přes toto pole je iterováno a z JSON objektů jsou potřebné informace ukládány do lokálních proměnných. Jedná se o:

- jméno - uloženo do lokální proměnné typu `String` jako `name`
- adresu streamu - ta je získána z JSON pole streamů jako první položka a je uložena do lokální proměnné typu `String` jako `url`
- názvy kategorií, do kterých stanice spadá - názvy kategorií jsou získány z JSON pole a uloženy do `List<String>` `styles`

Následně je vytvořen lokální objekt `Station`, do jehož konstruktoru jsou přidány parametry:

```
Station station = new Station(name, country, styles, url);
```

Nově vytvořená stanice je také zkontrolována, zdali již dříve nebyla uživatelem uložena mezi oblíbené. Tento algoritmus je stejný jako již popsáný algoritmus v `ServicePlayAPI` v kapitole 5.3.3. Stejný algoritmus, jako je v `ServicePlayAPI`, je také použit pro vytvoření `HashMap` s názvy kategorií jako klíči a listy stanic do nich patřících jako hodnoty těchto klíčů. Následně je již tento list stanic předán pomocí API Listeneru třídy `RadioData` do proměnné `localStationsList` a mapa kategorií do proměnné `categoriesMap`. Poté jsou stanice z třídy `RadioData` dostupné pro použití v aplikaci.

5.4 Ukládání nastavení uživatele - třída `PreferencesService`

Jelikož aplikace umožňuje ukládat vybrané stanice mezi „oblíbené“ je potřeba zajistit uchování informací, jaké stanice mezi oblíbené patří, i po vypnutí aplikace. K tomu je využita třída `SharedPreferences`, která poskytuje obecný framework pro ukládání primitivních dat do paměti telefonu a jejich opětovné načítání. [15] Ve třídě `PreferencesService` jsou implementovány tři metody:

- `void saveStationList(Context context, List<Station> stations, String name)`
- `List<Station> loadStationList(Context context, String name)`
- `void deletePref(Context context)`

Metoda `saveStationList` slouží pro uložení listu oblíbených stanic prostřednictvím třídy `SharedPreferences`. Kontext (`context`) aplikace je zde využit pro získání patřičného souboru, do kterého budou informace ukládány, `stations` je ukládaný list oblíbených stanic a `name` je jméno, pod kterým budou stanice uloženy. Pro uložení listu objektů stanic jako primitivních dat je využita knihovna `Gson`, která zprostředkovává serializaci Java objektů na JSON.

Obdobným způsobem je řešeno i načítání listu oblíbených stanic s tím rozdílem, že `Gson` knihovna je využita k deserializaci JSON dat na `ArrayList` stanic.

Metoda `deletePref` pro mazání souborů uložených jako `SharedPreferences` je implementována, v současném stavu aplikace ale není použita.

5.5 Spuštění aplikace

Před tím než je samotná aplikace Internetové rádio spuštěna, probíhá kontrola, zdali je telefon připojen k wi-fi síti. Pokud tomu tak není, je uživatel vyzván, aby se připojil k některé z wi-fi sítí v dosahu. Do té doby není možné aplikaci spustit. Stejně tak je kontrolována kvalita připojení. V případě, že je signál wi-fi sítě příliš slabý, aplikaci není možné spustit, a to z důvodu, že by mohlo docházet k častým výpadkům připojení nebo by jeho kvalita znemožňovala plynulé přehrávání streamů rádií. Pokud je wi-fi připojení v pořádku, je aplikace spuštěna a pomocí metody `Locale.getDefault().getLanguage()` zjištěno jazykové nastavení telefonu. To je reprezentováno řetězcem označujícím jazyk. Jedná se o mezinárodní kódy jazyků podle normy ISO 639-1 (například čeština má označení „cs“, angličtina „en“, němčina „de“). Toto označení je následně uloženo do proměnné `language` instance třídy `RadioData`, odkud je pak dále používáno. Podle jazyka telefonu je poté rozhodnuto, které API pro získání stanic bude použito. Pokud je jazyk v českém jazyce, je vytvořena instance třídy `ServicePlayAPI`, na níž je zavolána metoda `getAllStation()` vracející `JsonObjectRequest`. Tento objekt je přidán do `RequestQueue` (fronty požadavků) přes třídu `VolleySingleton`. Před tímto je na instanci třídy `ServicePlayAPI` nastaven `ApiListener`, který zajistí, že kolekce v `RadioData`, uchováující údaje o stanicích, jsou naplněny až ve chvíli, kdy jsou získána všechna data ze služby.

```
ServicePlayAPI spa = new ServicePlayAPI();
ApiListener apiListener = new ApiListener();
spa.setListener(apiListener);
VolleySingleton.getInstance(this)
.addToRequestQueue(spa.getAllStation());
```

Obrázek 13 Přidání dotazu pro získání českých stanic do `RequestQueue`.

Pro ostatní jazyky je tomu obdobně s tím rozdílem, že je použita třída `ServiceDirbleAPI` a do její metody pro získání `JsonObjectRequest` vstupuje jako parametr `country`, což je zkratka jazyka telefonu, a číslo požadované stránky zdroje stanic (blíže popsáno v kapitole 4.1.1 Dirble).

```
ServiceDirbleAPI sda = new ServiceDirbleAPI();
ApiListener apiListener = new ApiListener();
sda.setListener(apiListener);
VolleySingleton.getInstance(this)
.addToRequestQueue(sda.getAllStations(country, 1));
```

Obrázek 14 Přidání dotazu pro získání stanic z Dirble do `RequestQueue`.

5.5.1 Zobrazení stanic

Zobrazení listu stanic řeší přímo třída `BlindRadioCategoryListFragment`. Ta je univerzálně používána pro výpis seznamu stanic - je tedy využívána pro zobrazení všech stanic, stanic dané kategorie nebo listu oblíbených stanic. Položky, které má zobrazit, jsou dány listem `selectedStationsList` ze třídy `RadioData`. Pokud uživatel zvolí, že

chce zobrazit všechny stanice, je nastaven jako *selectedStationsList* list *localStationsList* z *RadioData* (v *localStationsList* jsou uloženy načtené stanice ze serveru konkrétní služby) a třída *BlindRadioCategoryListFragment* tak zobrazí všechny stanice, které jsou pro telefon momentálně dostupné. Při výpisu stanic je realizováno pravidlo, které při počtu položek větším než 15 přidá na poslední místo v seznamu také položku „vyhledávání“ (seznam je v kruhu, na poslední položku lze přejít z první položky posunem o jedno vzad). Třída *BlindRadioStationsListActivity* je pouze pomocnou třídou od kmenové aplikace, které je aplikace Internetové rádio součástí.

5.5.2 Vyhledávání

Třída *BlindRadioSearchActivity* slouží pro vyhledávání v seznamech stanic, ale také pro vyhledávání v kategoriích. To záleží, z jakého seznamu je do ní vstoupeno (to je mezi aktivitami předáváno parametry). Pokud je do ní vstoupeno z výpisu stanic, jsou z listu *selectedStationsList* z *RadioData* přesypána jména stanic do lokálního listu *searchList* s objekty typu *String*. V případě příchodu ze seznamu kategorií jsou do toho listu vloženy klíče z mapy kategorií z *RadioData* a poté je tento list abecedně seřazen.

Vyhledávací algoritmus je implementován v metodě *int searchInList(List<String> list, String phrase)*, do které jako parametr „list“ vstupuje *searchList* a jako parametr „phrase“ hledaná fráze zadaná uživatelem. Při neúspěšném hledání vrátí metoda *int* hodnotu -1, pokud je nalezen výsledek je vrácená hodnota číslo indexu vyhledané položky. Jako první jsou z *phrase* odstraněny všechny znaky, které nejsou písmena „A-Z“ a „a-z“, číslice nebo bílé mezery. Toho je dosaženo následujícím příkazem:

```
phrase = phrase.replaceAll("[^\\w\\s]", "");
```

Následuje kontrola, zdali *phrase* není prázdný řetězec. Pokud ano, je vrácena hodnota -1 a běh metody končí. V opačném případě dojde k procházení zadaného listu a porovnávání jeho položek se zadanou frází. Před porovnáváním probíhá ještě kontrola počtu znaků zadané fráze. Uživatel totiž může zadat pouze počáteční písmena pro vyhledávání nebo celý název stanice. Pokud je tedy zadaná fráze kratší než 4 znaky, dochází k porovnávání pouze počátečních znaků dané položky v listu a zadané fráze. Pokud je znaků v zadané frázi 4 a více, porovnávají se všechny znaky a skupiny znaků v položce seznamu se zadanou frází. Kdyby nebylo toto rozlišení, mohlo by dojít k případům, kdy by uživatel zadal pouze počáteční písmeno, které by chtěl vyhledat. V seznamu by však nemusela být žádná stanice začínající tímto písmenem a uživateli by mohla být vrácena stanice, která má toto písmeno vprostřed názvu. Naopak pokud by chtěl uživatel vyhledávat dvouslovné názvy, musel by vždy zadat i první slovo - nebylo by možné vyhledávat pouze druhým slovem. To by byla pro uživatele nepříjemnost - spousta názvů stanic totiž začíná slovem „Rádio“. Pokud tedy dojde k shodě ve vyhledávání, je metodou vrácena hodnota indexu, na které se stanice nachází a na tuto stanici je uživatel „přenesen“ ve výpisu stanic.

5.5.3 Výběr a přehrávání stanice

Při vybrání stanice je její instance uložena do třídy *RadioData* do proměnné *selectedStation* a stejně tak je v této třídě do proměnné *playlist* nastaven list *selectedStationsList*. Toto rozdělení seznamu stanic na *playlist* a *selectedStationsList* je z důvodu

možnosti „Pokračovat v přehrávání“ v menu aplikace. Rozdělení na dva seznamy umožňuje uživateli procházet různé seznamy stanic bez toho, aby aplikace ztratila informaci, který seznam byl reálně přehrávaný naposledy.

Poté, co je stanice vybrána, je spuštěna třída `BlindRadioPlayActivity`, která slouží jako „obrazovka přehrávače stanic“. Po spuštění této aktivity proběhne kontrola jazyka. Pokud je telefon v českém jazyce a stanice jsou tak načteny ze služby `Play.cz`, je potřeba před spuštěním přehrávání stanice zjistit její url adresu streamu. To je provedeno pomocí třídy `ServicePlayAPI` a její metody `getStationUrl(final String shortcut)`, kde `shortcut` je zkratka stanice uložené v `RadioData.getInstance().getSelectedStation()`. Jakmile je url adresa streamu získána, je zavolána metoda `startPlaying()`. V případě ostatních jazyků je tato metoda volána rovnou (stanice již mají informace o url adresách streamů).

Pro přehrávání streamů je použit „Freeware Advanced Audio (AAC) Decoder for Android“ [16], což je knihovna, která poskytuje dekodér a přehrávač internetových audio streamů. Třída `BlindRadioPlayer` je navržena jako singleton tohoto přehrávače a zajišťuje tak vytvoření pouze jedné instance pro přehrávání. Toto zamezuje náhodnému vytvoření více instancí přehrávače a tím také nekontrolovatelnému přehrávání více streamů najednou. V metodě `startPlaying()` je tedy nastaven a definován `PlayerCallback` na instanci třídy `BlindRadioPlayer`. Pomocí definování `PlayerCallback` lze reagovat na chyby, které přehrávač zachytí. Pokud stream není možné přehrát, je o tom uživatel, díky nastavení `PlayerCallback`, informován. Poté je již na instanci `BlindRadioPlayer` zavolána metoda `void playAsync(String url)`, která spustí přehrávání streamu na url adrese, jež je parametrem této metody. Toto url je získáno z vybrané stanice - `selectedStation` z `RadioData`.

```
BlindRadioPlayer.getInstance().playAsync(RadioData.
    getInstance().getSelectedStation().getSrcUrl());
```

Pokud je streamovací adresa funkční a `PlayerCallback` nezachytí žádnou chybu, je spuštěno přehrávání audio dat vybrané stanice. Přehrávání lze pozastavit nebo se během něj posouvat na další nebo předchozí stanice vybraného playlistu - jako `selectedStation` se nastaví následující či předchozí položka v `playlist` a následně je zavolána metoda `startPlaying()`. Přehrávání lze také gestem pozastavit, v tomto případě je na instanci přehrávače zavolána metoda `stop()`.

Právě přehrávanou stanici lze také uložit nebo odebrat z oblíbených stanic. Při této volbě je pozastaveno přehrávání a otevře se dialogové okno, které nabídne danou možnost. V případě, že stanice již je uložena v oblíbených:

```
RadioData.getInstance().getSelectedStation().isFavourite()
```

je uživateli nabídnuta možnost „Odebrat z oblíbených“ a pokud ji potvrdí, je tato stanice odstraněna z listu oblíbených v `RadioData` a její hodnota `favourite` je nastavena na "false". V opačném případě má uživatel k dispozici možnost „Přidat do oblíbených“ a pokud se tak rozhodne, je hodnota `favourite` stanice nastavena na "true". Po provedení kteréhokoliv z těchto úkonů je uživatel o dané akci informován. Poslední kontrolou před zavřením tohoto dialogového okna je ověření, zdali se uživatel nenachází v seznamu oblíbených stanic. Jestliže se nenachází v tomto seznamu, je proveden návrat na poslední vybranou stanici a přehrávač je uveden do stavu, ve kterém byl před otevřením tohoto dialogového okna (přehrávání nebo pozastavení). V opačném případě dojde k návratu

do hlavního menu aplikace, aby mohlo být provedeno opětovné načtení aktualizovaného seznamu oblíbených stanic.

Důležitým aspektem je, že pokud se uživatel posouvá mezi stanicemi během jejich přehrávání, je po návratu na seznam stanic na položce, kterou právě přehrával, a nikoliv na položce, ze které se dostal do přehrávání. To je důležité ke správné orientaci uživatele a k případnému návratu k přehrávání položky, kterou právě opustil.

5.6 Listování kategoriemi

Výpis kategorií funguje obdobně jako výpis stanic. Pro zobrazení jmen položek seznamu kategorií jsou brány klíče proměnné typu `HashMap categoriesMap` z `RadioData`. Po zvolení položky je její textová hodnota použita pro vyhledání mezi klíči v `categoriesMap`. Pomocí tohoto klíče, reprezentujícího název kategorie, je vybrána jemu příslušící hodnota, kterou je `ArrayList` stanic patřící do dané kategorie. Tento list stanic je předán do `selectedStationsList` v `RadioData` a následně je spuštěna třída `BlindRadioStationsListFragment`, kde dojde k zobrazení stanic vybrané kategorie. Princip zobrazení stanic včetně další práce s nimi byl již popsán v sekci 5.5.1.

5.7 Oblíbené stanice

Uživatel si může zařazovat stanice mezi „oblíbené“. Tyto stanice mu jsou dostupné z položky oblíbených stanic v hlavním menu aplikace. Informace, jaké stanice má uživatel vybrané mezi oblíbenými, je uchovávána i po ukončení aplikace nebo vypnutí telefonu. Toho je využito prostřednictvím třídy `SharedPreferences`, ukládání a načítání listu oblíbených stanic zajišťuje třída `PreferencesService` (vizte kapitola 5.4). Při spuštění aplikace je tedy volána metoda `loadStationList(getContext(), "favourites")`, která vrátí uživateli list oblíbených stanic, jenž je následně předán do proměnné typu `List<Station> favourites` z `RadioData` a dále je v aplikaci při práci s oblíbenými stanicemi používán tento list. Pokud ještě nebyly uloženy žádné stanice mezi oblíbené, vrátí metoda `loadStationList(getContext(), "favourites")` prázdný list.

Při zvolení položky „Oblíbené“, v hlavním menu aplikace, proběhne kontrola, zdali je potřeba synchronizovat oblíbené stanice. Nutnost kontroly je dána boolean proměnnou `needSync` v `RadioData`, která je po spuštění aplikace nastavená na "true". Synchronizace oblíbených stanic slouží k zajištění, že mezi načtenými oblíbenými stanicemi nebude stanice, kterou již nebylo reálně možné získat ze serveru dané služby. Toto není častý jev, nicméně může se stát, že stanice, kterou má uživatel uloženou jako oblíbenou prostřednictvím `SharedPreferences`, přestane být poskytována službou, jejíž API je využíváno. V aplikaci by pak tato stanice nebyla jinak dostupná než přes oblíbené stanice, což by mohlo narušit konzistentnost aplikace, zvláště v české verzi, kdy je url adresa streamu stanice získávána až těsně před jejím přehráváním. V případě, že potřeba synchronizovat stanice, je spuštěna metoda `synchronizeFavourites()`. Tato metoda porovnává načtené oblíbené stanice se všemi stanicemi získanými se serveru (uloženy v proměnné `localStationsList` v `RadioData`). Pokud stanice z oblíbených stanic není mezi stanicemi v `localStationsList`, je tato stanice z `favourites` odstraněna. Po synchronizaci je nastavena hodnota `needSync` na "false". Následně je list `favourites` nastaven jako `selectedStationsList` v `RadioData` a je spuštěna třída `BlindRadioStationsListFragment` a dále již aplikace pokračuje podle implementace popsané v sekci 5.5.1. Před

spuštěním `BlindRadioStationsListFragment` je navíc ještě nastavena proměnná boolean `isFavourite` z `RadioData` na "true". To slouží k určení chování aplikace, pokud je stanice odebrána z oblíbených. V případě, že je tomu tak učiněno ze seznamu oblíbených stanic, je uživatel po odstranění stanice navrácen na položku „Oblíbené“ v menu, aby mohlo dojít k aktualizaci zobrazovaného seznamu oblíbených stanic.

Při běhu aplikace jsou uživatelem nově zvolené oblíbené stanice přidávány do *favorites* v `RadioData`. Po ukončení aplikace rádia je tento list, s nově přidanými oblíbenými stanicemi, uložen do paměti telefonu prostřednictvím třídy `PreferencesService` a její metody `void saveStationList(Context context, List<Station> stations, String name)`, kde *name* určuje název uloženého listu. Tento list je poté vyvoláván pomocí metody `loadStationList(getContext(), "favorites")` při spuštění aplikace, jak bylo popsáno výše. Společně s ukládáním listu oblíbených stanice je také nastaven příznak `needSync` na "true".

5.8 Pokračování v přehrávání

Uživatel má možnost vrátit se k poslední přehrávané stanici přes položku v menu „Pokračovat v přehrávání“. Do té je možné vstoupit pouze v případě, že *playlist* z `RadioData` není prázdný - to znamená, že uživatel již přehrával nějakou stanici. Při zvolení „Pokračovat v přehrávání“ je spuštěna aktivita `BlindRadioPlayActivity`, do které je předáno číslo indexu stanice. Toto číslo bylo uloženo *stationId* v `RadioData` v momentě, kdy naposledy opouštěl přehrávání. Číslo indexu je použito pro určení pozice poslední přehrávané stanice v playlistu. Během přehrávání je tak možné přecházet na další stanice playlistu, ze kterého uživatel naposledy poslouchal poslední přehrávanou stanici.

6 Testování

Testování proběhlo ve dvou fázích. Nejprve byla aplikace otestována expertními testery z organizace SONS. Toto testování shrnul pan Michal Jelínek a jeho vyjádření je dostupné v příloze A. Všechny návrhy na opravu byly realizovány a jsou již zahrnuty v kapitole 5. Výjimkou je návrh číslo 3, jehož realizace by byla větším zásahem do logické struktury aplikace. Jeho implementace však bude zvažena pro budoucí úpravu. Stejně tak řešení zmíněných problematických míst bude vyžadovat hlubší analýzu. Bod 2 je způsobován externím dekodérem pro přehrávání (kapitola 5.5.3) a jeho analýza a případná oprava se bude týkat rovněž budoucí úpravy.

Druhou částí testování bylo kvalitativní testování s běžnými uživateli v laboratoři použitelnosti (Usability Lab) v budově Fakulty elektrotechnické na Karlově náměstí.

6.1 Test použitelnosti

Test použitelnosti je metoda kvalitativního testování, která může odhalit drobné problémy testované aplikace. Umožňuje vývojářům prozkoumat, zdali se uživatel v daném prostředí dobře orientuje a zdali je pro něj používání aplikace snadné a příjemné. Test probíhá v laboratoři použitelnosti, kde participant testování plní sadu úkolů vztahující se k testované aplikaci. Tento průběh je zaznamenáván a následně je provedena analýza, ve které jsou také brány v potaz participantovy odpovědi na otázky a jeho vyjádření k aplikaci. Výsledkem testování je zpráva o nalezených problémech a návrhu jejich řešení.

6.2 Usability lab

Usability lab neboli laboratoř použitelnosti jsou zpravidla dvě místnosti - místnost participantů (participant room) a místnost pozorovatelů (observer room). V místnosti participantů provádí účastník testování zadané úkoly. Místnost se snaží simulovat běžné podmínky, ve kterých bude aplikace používána, proto samotné pozorování organizátory probíhá v místnosti pozorovatelů, která je oddělená od místnosti participantů. Do místnosti pozorovatelů je v reálném čase přenášén audiovizuální záznam z místnosti participantů. Organizátoři si tak mohou vést poznámky bez toho, aby byl účastník testování rušen během plnění úkolů. Také je možné se k záznamu testování vrátit při pozdější analýze.

Pro záznam testování a jeho přenos mezi místnostmi je použit systém kamer a mikrofonů a také speciální software určený pro laboratoře použitelnosti od společnosti TechSmith Morae. Ten mimo jiné umožňuje zápis poznámek k úkolům s časovými údaji. Lze tak hodnotit i dobu trvání vykonání úkolu.

6.3 Uživatelé

Cílovou skupinou jsou nevidomí uživatelé používající mobilní telefon se systémem Android či jeho potenciální uživatelé. Tito lidé by také měli mít zájem poslouchat alespoň jednou měsíčně rádio.

6.3.1 Screener

Pro výběr vhodných participantů je použit tzv. screener, což je dotazník s otázkami, podle kterých lze určit, zdali je osoba vhodná na testování aplikace. Dále jsou pomocí něj získány základní informace o participantovi jako jeho věk apod. Screener je součástí Přílohy B.

6.3.2 Dotazník před testem

Dalším z použitých dotazníků je „Dotazník před testem“ (pre-test). Tento dotazník slouží ke zjištění dodatečných informací o vybraném participantovi. Otázky jsou většinou otevřeného typu a participant na ně odpovídá osobně před zahájením samotného testování. Zaměření těchto otázek směřuje již více k tématu testované aplikace. Použitý dotazník pro účely testování aplikace Internetové rádio je rovněž součástí Přílohy B.

6.3.3 Dotazník po testu

Po ukončení průchodu zadaných úkolů je participant dotázán na několik otázek z „Dotazníku po testu“ (post-test). Tyto otázky se týkají samotného testování a použitelnosti aplikace. Slouží tak jako vhodné doplnění poznatků získaných během provádění úkolů. Tento dotazník je součástí Přílohy B.

6.4 Průběh testování

V případě testování s nevidomými uživateli je v místnosti s participantem také moderátor testu. Ten se v úvodu sezení dotáže participanta na otázky z „Dotazníku před testem“ a poté mu předčítá jednotlivé úkoly testování. V případě testování aplikace Internetové rádio se jednalo o následující úkoly:

1. V menu telefonu naleznete aplikaci Rádio a tu spusťte.
2. Dlouhým stiskem zamykacího tlačítka aktivuje nápovědu, kterou si projděte – informace budete potřebovat v dalších úkolech (k nápovědě se můžete takto kdykoliv v aplikaci vrátit).
3. Spusťte stanici Country rádio.
4. Uložte si Country rádio do „oblíbených“, poté se vraťte se do hlavního menu aplikace Rádio.
5. Z kategorie Pop vyberte libovolnou stanici, kterou spusťte. Během přehrávání stanice přejděte na následující funkční stanici a tu uložte do „oblíbených“.
6. Vyhledejte rádio Proglas (použijte vyhledávání) a spusťte.
7. Zjistěte, jaké stanice máte uložené v „oblíbených“ a poté se vraťte k přehrávání poslední přehrávané stanice.
8. Pozastavte přehrávání.
9. Odstraňte stanici Country rádio z oblíbených stanic.

V místnosti pozorovatelů probíhá během plnění úkolů jejich zaznamenávání a psaní poznámek. Participant může svůj postup komentovat, aby byly organizátorům známy jeho myšlenkové pochody. Po absolvování úkolů je participant dotázán na otázky z „Dotazníku po testu“.

Testování se zúčastnili tři participanti - jeden muž a dvě ženy. Jejich odpovědi na otázky z dotazníků a stručné hodnocení prováděných úkolů jsou dostupné v příloze C.

6.5 Poznatky z testování

Všichni tři participanti používají telefon s Android aplikací pro nevidomé, do které byla aplikace Internetové rádio vytvářena. Ovládání rozhraní telefonu pro ně tedy nebyl problém a mohli se plně věnovat testované aplikaci. Účastníci se shodli, že od aplikace Internetové rádio očekávají především snadné spuštění rádiové stanice, mají také jednu nebo více oblíbených stanic. U dvou z nich jde především o konkrétní obsah, u kterého by mělo převažovat mluvené slovo a zpravodajství. Rychlost plnění úkolů se odvíjela především od toho, jak dlouho participant vlastní telefon s aplikací pro nevidomé. Pokud se jednalo o pokročilého uživatele, snadněji se zorientoval i v aplikaci Internetové rádio a uvědomoval si i zažité standardy (např. „vyhledávání“ na poslední položce v seznamu). U participanta, který neměl práci s telefonem ještě dostatečně zažitou, trvalo zorientování se v aplikaci Internetové rádio o něco déle a také častěji používal nápovědu, aby zjistil gesta a kde v aplikaci vykonávat dané úkoly. Obecně však nedělal žádný z úkolů participantům větší problémy a jejich vykonání hodnotili ve většině případů jako snadné či velmi snadné.

V závěrečném hodnocení označovali participanti aplikaci Internetové rádio jako snadno použitelnou a přehlednou, nejvíce oceňovali rozdělení stanic podle žánrů do kategorií. Funkce rádia považují za dostačující, nenapadlo je nic, co by jim v aplikaci chybělo. Jeden z participantů navrhol jako vylepšení aplikace možnost zjistit jméno aktuálně přehrávané skladby rádiové stanice. Pokud by měli možnost, všichni participanti by aplikaci Internetové rádio používali pro poslech rádia.

6.6 Nálezy a návrhy na zlepšení

Během testování nebyla objevena žádná zásadní chyba použitelnosti. V následující tabulce jsou tak návrhy na zlepšení některých věcí, které by umožnily rychlejší zorientování se nových uživatelů v aplikaci či by mohly rozšířit funkcionalitu aplikace.

	Nález	Návrh řešení
1	Po kliknutí na položku „Kategorie“ v hlavním menu si uživatel nemusí být zcela jistý, zdali vypsáný seznam představuje názvy kategorií nebo již názvy stanic nějaké kategorie.	Přidat do listu kategorií oznámení uživateli, aby si zvolil požadovanou kategorii. Bude tak jasné, že se nachází v seznamu kategorií.
2	Informace v nápovědě: „Možnost pokračovat v přehrávání spustí poslední přehrávaný playlist“ může být pro uživatele matoucí. Nemusí mu být zřejmé, co je myšleno playlistem.	Informaci nahradit: „Možnost pokračovat v přehrávání spustí poslední přehrávanou stanicí z naposledy použitého seznamu.“
3	Funkcionalitu přehrávání rádia by bylo možné rozšířit o možnost zjistit jméno aktuálně přehrávané skladby.	API služby používané pro české stanice tuto možnost poskytuje a bylo by ji tak možné implementovat. Pro tuto funkci u stanic jiných jazyků by bylo nutné využít nejspíš externí aplikaci.

Tabulka 1 Nálezy a návrhy

7 Závěr

Práce popisuje analýzu a implementaci mobilní aplikace internetového rádia pro nevidomé. Před její implementací bylo analyzováno několik aplikací poskytující vysílání internetových rádií, což vedlo k získání užitečných poznatků při vývoji aplikace. Klíčovými body pro realizaci Internetového rádia byly především specifické požadavky a potřeby nevidomých uživatelů, kterým bylo i díky konzultacím s těmito uživateli vyhověno, což byl hlavní a elementární cíl práce - uživatelsky příjemná aplikace, která nabídne zrakově postiženým rozšíření využití mobilního telefonu ke zkvalitnění jejich života. Čímž poslech rádia nepochybně je.

Důležitou částí práce bylo testování aplikace nevidomými uživateli. To mělo odhalit případné chyby použitelnosti a mimo jiné také zjistit názor participantů na kvalitu vyvinuté aplikace. Výsledek byl pozitivní - nebyly zjištěny žádné závažnější problémy v použitelnosti a uživatelé byli s aplikací spokojeni. Hodnotili ji jako snadno použitelnou, přehlednou a splňující jejich očekávání, a to tak, že by ji na svém telefonu rádi využívali. Tímto tak lze považovat cíl práce za splněný.

Příloha A

Hodnocení funkce Rádio jako součást aplikace pro nevidomé

Toto hodnocení aplikace Internetové rádio poskytl pan Michal Jelínek z organizace SONS, který patří mezi hlavní testery aplikace pro Android, jejíž je Internetové rádio součástí. Následující text v příloze A je tak citací vyjádření pana Jelínka.

A.1 Úvod

Funkce umožňující přehrávání streamovaných rádií byla jednou z těch, o které byl již mezi prvotními uživateli, potenciaálními uživateli a testery aplikace umožňující ovládání systému Android velký zájem. Její zařazení mezi nabízené funkce je tedy krok správným směrem.

A.2 Uživatelské prostředí

Uživatelské prostředí Internetového rádia se drží zavedených principů celkového ovládání jeho kmenové aplikace. Uživatel tedy zvládne ovládání s běžnými znalostmi jejího prostředí. Nabízené funkce posluchače uspokojí. Výběr z poměrně slušného množství stanic, možnost jejich výběru dle žánrů, uložení mezi oblíbené a rychlý přechod k naposled poslouchané stanici je maximum, o co by cílový uživatel v českém prostředí mohl stát.

A.3 Problematická místa

1. Rádio neukončuje přehrávání při opuštění funkce např. při použití tlačítka HOME. To by mělo být při opuštění zajištěno vždy. Při hlasitější produkci uživateli hrozí, že neuslyší hlasový výstup.
2. Rychlé procházení stanic v režimu jejich přehrávání zapříčiní restart celé aplikace.

A.4 Návrhy na úpravu

1. Funkce pro vyhledání stanice by měla být v jejich seznamu umístěna jako poslední, nikoli první. Tím bude zajištěno umístění shodné s ostatními funkcemi umožňujícími vyhledávání.
2. Při přehrávání je možné stanici uložit do oblíbených. Po uložení je nyní nutné přehrávání spustit ručně – mělo by automaticky pokračovat.

3. Po odstranění stanice z oblíbených jsou oblíbené opuštěny návratem do nadřazené nabídky. Uživatel by měl v oblíbených zůstat.
4. Vyhledávání stanic umožňuje vyhledávat pouze pomocí počátečních písmen. Uživatel je po vyhledání přesunut na první stanici, jejíž název těmito písmeny začíná. Mnoho názvů stanic začíná slovy radio či rádio. Vhodnější by bylo vyhledávání v celých názvech a po vyhledání zobrazit jen vyfiltrovaný seznam, jako je tomu např. u kontaktů.

A.5 Shrnutí

Funkce Rádio je vítaným rozšířením Android aplikace pro nevidomé. Po opravě zmíněných chyb a případném zapracování návrhů nebude funkce z uživatelského hlediska takřka co vytknout.

12. 5. 2016 Michal Jelínek (jelinek@sons.cz)

Příloha B

Dotazníky

Tato příloha obsahuje dotazníky použité při testování.

B.1 Screener

Email:

Telefon:

1. Jaký je Váš věk?
2. Byl/a jste někdy účastníkem testu použitelnosti?
 - a) Ano
 - b) Ne
3. Na jaké úrovni používáte mobilní telefon?
 - a) Pouze na volání a sms zprávy.
 - b) Volání, sms zprávy a další základní funkce jako budík, kalendář apod.
 - c) Využívám téměř všechny možnosti telefonu i ty pokročilejší jako webový prohlížeč, emailového klienta apod.
4. Jak často posloucháte rádio? (nezáleží prostřednictvím jakého zařízení)
 - a) Vůbec.
 - b) 1-3x měsíčně.
 - c) 1-3x týdně.
 - d) Denně.

B.2 Dotazník před testem

1. Jaký používáte mobilní telefon (typ, operační systém, typ přizpůsobení)?
2. Co Vám na něm vyhovuje (plusy)?
3. Co Vám na něm naopak vadí (mínusy)?
4. Co očekáváte od mobilní rádio aplikace (co by měla umět)?
5. Máte oblíbené stanice nebo vybíráte podle toho, co která stanice momentálně hraje?

B.3 Dotazník po testu

1. Jak byste hodnotil/a náročnost úkolů?
2. Co Vám způsobovalo největší potíže?
3. Jaký máte celkový pocit z aplikace Rádio? Je snadné/obtížné se v ní orientovat?
4. Používal/a byste tuto aplikaci, kdybyste chtěl/a poslouchat rádio?
5. Je něco, co Vám v aplikaci Rádio chybí?

Příloha C

Participanti - získaná data

Tato příloha obsahuje získaná data od participantů testování. Znění otázek je v příloze B.

C.1 Participant 1

C.1.1 P1 - Screener

1. 61.
2. Ano.
3. Pouze na volání a sms zprávy.
4. Denně.

C.1.2 P1 - Dotazník před testem

1. Chytrý telefon Android s aplikací BlindShell.
2. Jsem s ním spokojený po všech stránkách.
3. Po skončení hovoru se nevrátí tam, kam bych chtěl.
4. Snadné spuštění, měla by mít zpravodajské stanice.
5. Ano, mám - Radiožurnál, ČRo 2, Rádio Sever a regionální stanice. Preferuji mluvené slovo.

C.1.3 P1 - Dotazník po testu

1. Škála pro jednotlivé úkoly: velmi snadné – snadné – přiměřené – náročné – velmi náročné
 - I velmi snadné
 - II velmi snadné
 - III velmi snadné
 - IV snadné
 - V snadné
 - VI velmi snadné
 - VII velmi snadné
 - VIII velmi snadné
 - IX velmi snadné
2. Správně použít gesto pro pozastavení, aby bylo rozeznatelné.
3. Líbí se mi, byl bych s ním spokojený. Je jednoduché.
4. Ano.
5. Ne.

C.2 Participant 2

C.2.1 P2 - Screener

1. 46.
2. Ano.
3. Volání, sms zprávy a další základní funkce jako budík, kalendář apod.
4. 1-3x měsíčně.

C.2.2 P2 - Dotazník před testem

1. Chytrý telefon Android s jednou z prvních verzí BlindShell.
2. Jsem spokojená.
3. Bylo by třeba propojit lépe kontakty s výpisem zmeškaných hovorů. Ze zmeškaných hovorů by se mělo jít dostat rovnou na konkrétní kontakt.
4. Chtěla bych si spustit stanici bez jakéhokoliv nastavování, vše by mělo být přednastavené.
5. Poslouchám Rádio Classic, Český rozhlas 2. Mám ráda mluvené slovo - BBC zprávy, Rádio 7.

C.2.3 P2 - Dotazník po testu

1. Škála pro jednotlivé úkoly: velmi snadné – snadné – přiměřené – náročné – velmi náročné
 - I velmi snadné
 - II velmi snadné
 - III snadné
 - IV snadné
 - V přiměřené
 - VI velmi snadné
 - VII přiměřené
 - VIII snadné
 - IX snadné
2. Neuvědomila jsem si, že je v menu položka Kategorie. Bylo by lepší upravit jména samotných kategorií.
3. Je dobré.
4. Ano, používala, na poslouchání večer třeba.
5. Nic mě nenapadá.

C.3 Participant 3

C.3.1 P3 - Screener

1. 23.
2. Ano.
3. Volání, sms zprávy a další základní funkce jako budík, kalendář apod.
4. 1-3x měsíčně.

C.3.2 P3 - Dotazník před testem

1. Chytrý telefon s Blindshell jako služební telefon – mám ho 3-4 měsíce, jinak Nokii C5 (tlačítkový telefon s hlasovou odezvou).
2. Na Nokii mi vyhovuje rychlost, při psaní SMS a procházení. Na BlindShell se mi líbí v nové verzi, že při psaní nemusím přiklepávat druhým prstem.
3. Jsem spokojená s Nokií, hlavně z důvodu tlačítek a rychlejšího psaní.
4. Poslech stanic, přechod mezi nimi, přečtení písničky co zrovna hraje.
5. Občas poslouchám Evropu 2.

C.3.3 P3 - Dotazník po testu

1. Škála pro jednotlivé úkoly: velmi snadné – snadné – přiměřené – náročné – velmi náročné
 - I velmi snadné
 - II velmi snadné
 - III přiměřené
 - IV velmi snadné
 - V velmi snadné
 - VI snadné
 - VII snadné
 - VIII snadné
 - IX velmi snadné
2. Psaní na klávesnici a pomalejší odezvy telefonu.
3. Je jednoduché a přehledné, dobré je rozdělení do žánrů. Myslím si, že to bude mít úspěch. Orientace je snadná a přehledná.
4. Kdybych ho měla, tak bych ho asi používala.
5. Nic, je to pro mě dostačující. Mohlo by to říkat název písničky, ale to je takové navíc.

Příloha D

Obsah přiloženého CD

Na přiloženém CD nosiči se nachází:

```
/
├── Integrace služby internetového
│   ├── audio rádia.pdf ..... Pdf soubor s textem této
│   │                               práci.
│   └── radio/ ..... Tento adresář obsahuje
│       ├── zdrojové soubory
│       ├── aplikace a adresáře
│       └── reprezentující balíčky v
│           aplikaci.
└── strings.xml ..... Xml soubor s textovými
    řetězci použitými v
    aplikaci.
```

Literatura

- [1] Pew Research Center. *Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies*. [online]. [23. 04. 2016]. URL: http://www.diaipoimansi.gr/PDF/pew_research%201.pdf.
- [2] Cnet.com. *The Android Atlas*. [online]. [23. 04. 2016]. URL: <http://www.cnet.com/android-update/>.
- [3] World Health Organization. *Visual impairment and blindness*. [online]. [23. 04. 2016]. URL: <http://www.who.int/mediacentre/factsheets/fs282/en/>.
- [4] Petr Svobodník. “Zpřístupnění mobilních telefonů se systémem Android pro nevidomé uživatele”. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačové grafiky a interakce, 2013.
- [5] Android.com. *Android Interfaces and Architecture*. [online]. [28. 04. 2016]. URL: <https://source.android.com/devices/>.
- [6] Grant Allen. *Android 4. Průvodce programováním mobilních aplikací*. 1. vydání. Computer Press, 2013. ISBN: 9788025137826.
- [7] TuneIn. *About TuneIn*. [online]. [05. 05. 2016]. URL: <http://tunein.com/about/>.
- [8] Milan Menšík. *XiiaLive*. [online]. [05. 05. 2016]. URL: <http://www.svetandroida.cz/xiialive-201004>.
- [9] Google Play. *Application information*. [online]. [05. 05. 2016]. URL: <https://play.google.com/store>.
- [10] Mack Steve a Rayburn Dan. *Hands-On Guide to Webcasting: Internet Event and AV Production. (Hands-On Guide Series)*. 1. vydání. Focal Press, 2005. ISBN: 0240807545.
- [11] Håkan Nylén. *Dirble*. [online]. [10. 05. 2016]. URL: <https://dirble.com/>.
- [12] Michael Jakl. *REST - Representational State Transfer*. [online]. [10. 05. 2016]. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.7334&rep=rep1&type=pdf>.
- [13] Play.cz. *Play.cz API Documentation*. [online]. [10. 05. 2016]. URL: <https://bitbucket.org/playcz/api.play.cz/wiki/Home>.
- [14] SONS (Sjednocená organizace nevidomých a slabozrakých ČR). *SONS - o nás*. [online]. [10. 05. 2016]. URL: <http://www.sons.cz/>.
- [15] Android. *Android developers*. [online]. [14. 05. 2016]. URL: <http://developer.android.com/>.
- [16] Spoledge. *AACDecoder - Freeware Advanced Audio (AAC) Decoder for Android*. [online]. [14. 05. 2016]. URL: <https://code.google.com/archive/p/aacdecoder-android/>.