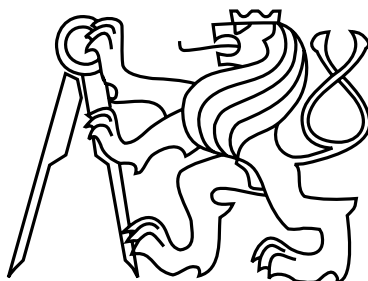


# Na tomto místě bude oficiální zadání vaší práce

- Toto zadání je podepsané děkanem a vedoucím katedry,
- musíte si ho vyzvednout na studijním oddělení Katedry počítačů na Karlově náměstí,
- v jedné odevzdané práci bude originál tohoto zadání (originál zůstává po obhajobě na katedře),
- ve druhé bude na stejném místě neověřená kopie tohoto dokumentu (tato se vám vrátí po obhajobě).



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



Bakalářská práce

**Aplikace pro záznam polohy a komunikaci zrakově postižených  
uživatelů**

*Adam Šimek*

Vedoucí práce: Ing. Jan Balata

Studijní program: Softwarové technologie a management, strukturovaný, Bakalářský

Obor: Web a multimedia

24. května 2012



## Poděkování

Rád bych poděkoval vedoucímu této práce panu Ing. Janu Balatovi za poskytnutí mnohých cenných rad a připomínek, bez kterých by tato práce mohla jenom těžko vzniknout.



## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze

.....





# Abstract

The main objective of this thesis was to create a mobile assistance application for visually impaired people, intended to help them with orientation problems. The thesis focuses on the problem of continuous user location capture, on detecting movements of the device, and resolving its context. There was a requirement of maintaining sufficient battery life of the device. Furthermore this thesis focuses on implementation of accessible user interfaces for touch enabled devices. The application is intended to be running on Android OS.

As a result it has been managed to implement a stable application – which can capture user's location – as well as a usable user interface. Thanks to application tests and feedback provided by participants of the tests a lot of output data has been acquired. This output data will be used as worthy input in further phases of this project.

# Abstrakt

Cílem této práce bylo vytvoření mobilní asistivní aplikace pro zrakově postižené uživatele, která jim má pomoci s orientací. Práce se zaměřovala na vyřešení problémů kontinuálního ukládání polohy uživatele, detekce pohybu uživatele a kontextu tohoto pohybu a odesílání naměřených dat na server. Požadavkem bylo zachování dostatečné výdrže zařízení při běhu na baterie. Dále se práce zaměřovala na implementaci přístupného uživatelského rozhraní pro dotyková mobilní zařízení. Aplikace byla určena pro běh na operačním systému Android.

Povedlo se naimplementovat stabilní aplikaci ukládající polohu uživatele včetně použitelného uživatelského rozhraní. Z testů implementovaného prototypu a zpětné vazby od participantů se povedlo získat mnoho výstupních dat využitelných v dalších fázích projektu.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Popis problému, specifikace cíle</b>	<b>3</b>
2.1	Motivace k vytvoření aplikace a úvod do problematiky . . . . .	3
2.2	Cílová skupina . . . . .	4
2.3	Požadavky na aplikaci na základě cílové skupiny . . . . .	4
<b>3</b>	<b>Analýza a návrh řešení</b>	<b>5</b>
3.1	Specifikace funkčních požadavků . . . . .	5
3.2	Nefunkční požadavky . . . . .	5
3.3	Uživatelské scénáře . . . . .	6
3.4	Detekce uživatelského kontextu . . . . .	6
3.4.1	Způsob detekce uživatelského kontextu . . . . .	6
3.4.2	Měření dat . . . . .	6
3.4.3	Naměřená data . . . . .	7
3.5	Určení polohy uživatele . . . . .	8
3.5.1	Způsoby určování polohy . . . . .	8
3.5.2	Global positioning system . . . . .	9
3.5.3	Aspekty použití GPS . . . . .	10
3.6	Optimalizace spotřeby . . . . .	10
3.6.1	Lokace uživatele . . . . .	10
3.6.2	Detekce pohybu a jeho kontextu . . . . .	11
3.6.3	Odesílání dat na server . . . . .	11
3.6.4	Spotřeba aplikace jako celku . . . . .	12
3.6.5	Optimalizace množství přenesených dat . . . . .	12
3.7	Uživatelské rozhraní . . . . .	12
3.8	Operační systém Android . . . . .	14
3.8.1	Architektura systému . . . . .	14
3.8.2	Programování pro OS Android . . . . .	14
	Architektura aplikace a programovací rozhraní . . . . .	15
	Instalace Android SDK . . . . .	15
<b>4</b>	<b>Realizace</b>	<b>17</b>
4.1	Úvod . . . . .	17
4.2	Architektura aplikace . . . . .	17

4.3	Uživatelské rozhraní . . . . .	18
4.4	Detekce uživatelského kontextu . . . . .	18
4.4.1	Výpočet rychlé Fourierovy transformace . . . . .	18
4.5	Lokace uživatele . . . . .	20
<b>5</b>	<b>Testování</b>	<b>21</b>
5.1	Testování low-fidelity prototypu . . . . .	21
	Úvod . . . . .	21
5.1.1	Popis testování . . . . .	22
	Testovací setup . . . . .	22
5.1.2	Průběh testování . . . . .	23
5.1.3	Shrnutí testů . . . . .	25
5.1.4	Doporučení pro další fáze projektu . . . . .	26
5.2	Testování highfidelity prototypu . . . . .	26
5.2.1	Úvod do testování . . . . .	26
5.2.2	Testování funkčních požadavků . . . . .	26
	Popis testů a testovací setup . . . . .	26
	Průběh testů . . . . .	27
5.2.3	Testování uživatelského rozhraní . . . . .	31
	Popis zadaných úkolů . . . . .	31
	Scénář testu . . . . .	31
	Popis zadané trasy a úkolu . . . . .	32
	Testovací setup . . . . .	32
	Průběh testů . . . . .	33
5.3	Diskuze výstupu testů . . . . .	41
5.3.1	Doporučení a návrhy na úpravy do dalších fází projektu . . . . .	41
	Uživatelské rozhraní . . . . .	41
	Ukládání polohy, úspora energie a odesílání dat . . . . .	42
	Vylepšení možností komunikace a vyhledání pomoci . . . . .	43
	Doporučená dodatečná funkcionalita . . . . .	44
<b>6</b>	<b>Závěr</b>	<b>45</b>
6.1	Zhodnocení splnění cílů projektu . . . . .	45
6.2	Doporučení do dalších fází projektu . . . . .	45
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>49</b>
<b>B</b>	<b>Obsah přiloženého CD</b>	<b>51</b>

# Seznam obrázků

3.1	Graf vektorů zrychlení zařízení položeného na stole a při chůzi (zleva) . . . . .	8
3.2	Graf vektorů zrychlení při jízdě autobusem po dlažebních kostkách a v tramvaji (zleva) . . . . .	8
3.3	Graf vektorů zrychlení při chůzi a graf odpovídajícího výstupu Fourierovy transformace (zleva) . . . . .	9
3.4	Ukázka uživatelské rozhraní aplikace. Zleva nejprve výchozí nabídka, dále nabídka vyhledání pomoci . . . . .	13
4.1	Diagram tříd zobrazující architekturu aplikace (zahrnuje pouze vybrané třídy z důvodu zachování čitelnosti) . . . . .	17
4.2	Ukázka základního pohledu na UI . . . . .	18
4.3	Vzorec pro výpočet absolutní hodnoty imaginárního čísla . . . . .	20
5.1	Uživatelské rozhraní prototypu . . . . .	21
5.2	Testovací setup . . . . .	24
5.3	Nexus S . . . . .	24
5.4	Zobrazení dat z prvního testu na mapě . . . . .	28
5.5	Zobrazení dat z druhého testu na mapě . . . . .	29
5.6	Snímky obrazovky zobrazující průběh vybíjení baterie při běhu aplikace v pohotovostním režimu. Na levém obrázku je stav před testem a vpravo stav po testu. . . . .	30
5.7	Snímky obrazovky zobrazující průběh vybíjení baterie při běhu aplikace režimu ukládání polohy. Na levém obrázku je stav před testem a vpravo stav po testu. . . . .	30
5.8	Fotografie zachycující průběh testu s uživatelem . . . . .	40



# Seznam tabulek

5.1	Vstupní dotazník . . . . .	22
5.2	Dotazník po testu . . . . .	23
5.3	Přehled participantů . . . . .	33
5.4	Participant 1 - předtestový dotazník . . . . .	34
5.5	Participant 1 - dotazník po testu . . . . .	35
5.6	Participant 2 - předtestový dotazník . . . . .	36
5.7	Participant 2 - dotazník po testu . . . . .	36
5.8	Participant 3 - předtestový dotazník . . . . .	37
5.9	Participant 3 - dotazník po testu . . . . .	38
5.10	Participant 4 - předtestový dotazník . . . . .	38
5.11	Participant 4 - dotazník po testu . . . . .	39





# Kapitola 1

## Úvod

Současný svět je skrz naskrz protkán moderními technologiemi. Již několik let je standardem osobní počítač s připojením k internetu, který je bránou do celého světa. V poslední době se tímto standardem stávají i mobilní telefony s otevřeným operačním systémem, který dovoluje instalovat uživatelem zvolený software. Takzvané chytré telefony také umožňují připojení k internetu a obsahují celou řadu různých senzorů, nejčastěji se jedná o podporu GPS, akcelerometr, gyroskop, digitální kompas.

Tyto vlastnosti v kombinaci s výkonným procesorem, dostatkem operační paměti a velkými displeji s vysokým rozlišením jsou malým zázrakem, který téměř každý z nás nosí v kapse. Bylo by škoda nevyužít možností, které nám poskytují. Co se přímo nabízí, je využití těchto funkcí a možností, které poskytují, jako základu asistivních technologií pro osoby se zrakovým postižením.



## Kapitola 2

# Popis problému, specifikace cíle

Tato bakalářská práce je součástí většího projektu NaviTerier [4], který v sobě zahrnuje asistivní a navigační aplikace pro zrakově postižené uživatele, a to jak pro venkovní využití, tak pro použití uvnitř budov.

### 2.1 Motivace k vytvoření aplikace a úvod do problematiky

Cílem této práce je pomoci zrakově postiženým uživatelům při pohybu ve ztížených podmínkách, například ve městě, kde mohou snadno ztratit orientaci. Zrak je bezesporu jedním z hlavních smyslů využívaných pro orientaci v prostoru.

V důsledku toho se lidé mající se zrakem potíže hůře orientují a je pro ně obtížnější zjistit, co je okolo nich, nebo najít správnou cestu k cíli. Toto vše je ještě umocněno, pokud jde o pro postiženého neznámé místo. Časem si totiž člověk okolí zapamatuje, takže když někde chodí často, tak už ví, co ho kde čeká a jaké se mohou vyskytnout potíže. Tyto zkušenosti má však každý pouze s místy, kudy často chodí, tam, kde nikdy nebyl, to zákonitě nemůže znát a bez zraku se orientuje mnohem hůře.

Principem je sdílení těchto zkušeností s určitými místy mezi jednotlivými uživateli aplikace. Jak bylo řečeno, slepý člověk poznává místa, kudy chodí, a pokud někudy chodí dostatečně často, je možné předpokládat, že to tam také zná. Aplikace bude toto chování sledovat a posílat získaná data na centrální server, kde se bude vytvářet databáze uživatelů a oblastí u kterých je předpoklad, že to tam znají. Pokud se nějaký další uživatel aplikace dostane do problémů, tak mu na základě nasbíraných dat bude nabídnut vhodný další uživatel, se kterým se bude moci přímo spojit a zkusit získat nějakou radu.

Důležité je rozpoznání toho, jestli uživatel jde, nebo se pohybuje například v prostředí městské hromadné dopravy. Cílem je totiž pomoci lidem s orientací při chůzi, a to na základě zkušeností ostatních uživatelů. Při jízdě autem nebo MHD si však člověk své okolí nezapamatuje moc dobře, a pokud jde o zrakově postiženou osobu, tak si ho nezapamatuje prakticky vůbec. Pro správnou funkčnost je tedy z principu nutné filtrovat pouze ty úseky, po kterých se uživatel pohyboval pěšky a je tedy pravděpodobné, že to tam zná.

## 2.2 Cílová skupina

Cílovou skupinou aplikace jsou zrakově postižení uživatelé. Zrakových postižení je celá řada a jsou tříděna do kategorií podle své závažnosti. Aktuální tabulka stupňů postižení je k dispozici na webu Světové zdravotnické organizace [17]. Tato práce je zaměřena spíše na osoby s těžším zrakovým postižením tedy kategorie 3, 4 a 5, které vlastně představují různé stupně slepoty. Podle Světové zdravotnické organizace žije na světě přibližně 50 milionů nevidomých lidí.[15]

## 2.3 Požadavky na aplikaci na základě cílové skupiny

Aplikace je zaměřená na osoby se zrakovým postižením [15]. z toho vyplývá řada omezení a nároků. Zařízení s dotykovým displejem obecně nejsou moc vhodná pro uživatele se zrakovým postižením kvůli špatné hmatové odezvě a tomu, že ovládací prvky na displeji jsou dynamické, uživatel si tedy nemůže snadno zapamatovat nějaké rozložení prvků a vytvořit si tak mentální model uživatelského rozhraní aplikace. To se ani tolik netýká přímo samotného uživatelského rozhraní, ale spíše problémů se systémovými klávesami nebo globálně nedostupné hardwarové klávesnice.

Z důvodů zhoršených možností interakce se zařízením musí být implementováno speciální uživatelské rozhraní a ovládání samotné aplikace je nutno maximálně zjednodušit, vlastně je nejlepší uživatele úplně oprostit od potřeby aplikaci ovládat. Je důležité uživatele odstínit od nutnosti textového vstupu, jelikož softwarová klávesnice není pro zrakově postižené pohodlně použitelná a analýza řeči sice již dnes funguje obstojně, problémy ale nastanou s případnou korekcí vstupu, systém předpokládá opravu pomocí textového vstupu a pro zrakově postiženého je problém text na softwarové klávesnici napsat, zejména je potíž vůbec dostat kurzor na správnou pozici. Jedinou uvažovatelnou možností je implementace speciální klávesnice, která by zadávání textu usnadnila.

K poskytnutí zpětné odezvy uživateli, předávání informací o stavu aplikace a obsahu menu je ale již syntéza řeči vhodná. Dnes je na vysoké úrovni a výstup je velice srozumitelný. Existují kvalitně zpracované hlasy i v češtině a v určité formě jsou dostupné pro všechny hlavní mobilní platformy. Pro telefony s operačním systémem Symbian je to například produkt Mobile Speak od Code Factory [3]. Telefony s operačním systémem Android mají tuto funkcionalitu přímo jako součást systému, je ale nutné instalovat doplňkové jazyky, případně i alternativní syntetizátory jako je například SVOX Classic [7].

Z hlediska návrhu aplikace je důležitý také kontext, v jakém se bude aplikace používat. Vzhledem k zaměření je téměř jisté, že aplikace bude používána venku na veřejném prostranství a v okolí bude nespočet rušivých elementů. Zrakově postižení uživatelé navíc v ruce typicky drží slepeckou hůl, takže pro ovládání telefonu mají k dispozici pouze jednu ruku. Aplikace by tedy měla být snadno ovladatelná i při tomto omezení. Dalším aspektem je použití ve venkovním prostředí, kdy mohou nastat potíže s tím, že uživatel nebude slyšet hlasový výstup aplikace. Hlasitost by tedy měla být nastavitelná tak, aby mohla být dosažena optimální úroveň hlasitosti za všech podmínek (automobily, tramvaje, jiný městský hluk).

# Kapitola 3

## Analýza a návrh řešení

Analýza a návrh implementace (včetně diskuse různých alternativ a volby implementačního prostředí). v této sekci je rozpracováno konečné řešení, jsou zde zpracovány veškeré aspekty, které bylo nutno při návrhu aplikace zvážit.

### 3.1 Specifikace funkčních požadavků

V této části je uveden seznam funkčních požadavků, které by výsledná aplikace měla splňovat.

- Sbírání dat o poloze uživatelů pomocí systému GPS
- Odesílání těchto dat na centrální server
- Uživatelské rozhraní přístupné pro zrakově postižené osoby
- Detekce kontextu pohybu uživatele a odesílání těchto dat společně s polohou
- Vyhledat pomoc v podobě kontaktu na jiného uživatele, který má zkušenosti s problémovou lokalitou
- Inteligentní úprava měřících intervalů na základě pohybu uživatele

### 3.2 Nefunkční požadavky

- Běh na pozadí a snaha o co nejnížší možný počet nutných interakcí uživatele s aplikací
- Maximální úspora baterie a přenesených dat
- Běh na platformě OS Android ve verzi 2.3 Gingerbread a vyšší

### 3.3 Uživatelské scénáře

#### Základní běh na pozadí

Základním stavem aplikace bude služba běžící na pozadí, periodicky sbírající data o uživateli a reagující na konkrétní změny těchto dat na základě přednastavených a v průběhu času upravovaných kritérií. Předpokladem pro správnou funkčnost je tedy to, že uživatel bude mít telefon s běžící aplikací u sebe. v tomto režimu není pro zajištění funkce nutná žádná přímá interakce uživatele s aplikací.

#### Vyhledání pomoci

V případě, že by uživateli nastaly problémy s orientací, aplikace umožní vyhledání pomoci odesláním požadavku na centrální server, který v rámci odpovědi aplikaci předá údaje o potenciálních osobách, které by mohli uživateli pomoci s lepší orientací. Aplikace tyto výsledky seřadí podle relevance a nabídne uživateli možnost jednotlivé osoby kontaktovat, a to telefonicky.

### 3.4 Detekce uživatelského kontextu

Detekcí uživatelského kontextu je myšleno rozlišení toho, jestli uživatel jede nějakým dopravním prostředkem (například automobilem), nebo se pohybuje pěšky. Toto je nezbytné z toho důvodu, že princip celého projektu je založen na sdílení uživatelských zkušeností s terénem. Ty se ale dají získávat pouze při chůzi, protože pokud jede uživatel například autem, tak není vystaven kontaktu s prostředím, takže data, která by byla získána, by byla bezcenná. v případě úspěšné detekce kontextu bude možno data filtrovat a zachytávat pouze v kontextu chůze. Díky tomu budou v databázi uloženy pouze ty úseky, na nichž se uživatel pohyboval pěšky a u kterých můžeme tedy předpokládat, že je alespoň trochu zná, na rozdíl od úseků, které pouze projede bez jakéhokoliv kontaktu.

#### 3.4.1 Způsob detekce uživatelského kontextu

Pro detekci kontextu jsou používána data z akcelerometru zabudovaného v telefonu, u zařízení s OS Android je to prakticky standard. Pro zpracování získaných vektorů zrychlení jsou nasbíraná data zpracována pomocí rychlé Fourierovy transformace [11], která převede signál z časové domény do domény frekvenční. Poté je využito toho, že lidská chůze je periodický pohyb, který má zřetelné dominantní amplitudy, a je tedy možno ho z frekvenčního popisu signálu detekovat. Oproti tomu jízda autem nebo jiným vozidlem není periodickým pohybem, takže se oba dva typy pohybu dají dobře odlišit. Tato myšlenka rozpoznávání kontextu pohybu byla již dříve rozpracována v diplomové práci zabývající se Kontextovou inteligentní navigací [12].

#### 3.4.2 Měření dat

Data byla získávána z akcelerometru, který je přímo součástí zařízení. Výstup akcelerometru poskytuje ve formě pole obsahujícím tři vektory, pro každý směr zrychlení jeden.

Hodnoty jsou zaznamenávány vždy ve 4 sekundových intervalech, po jejichž uplynutí je z nashromážděných dat vypočítán kontext. Jelikož frekvence chůze je přibližně kolem 1 Hz, tak se tato hodnota ukázala jako dostatečná pro jisté zjištění kontextu a i jako relativně krátká pro zajištění dostačující odezvy. Akcelerometr běží přibližně na frekvenci 50 Hz, takže je po každém měření k dispozici 200-250 vektorů zrychlení pro jedno měřící okno. Toto podle Shannonova teorému [9] umožní detekovat frekvence až do 25 Hz, což pro účely této práce zdaleka postačuje. Získané hodnoty byly poté převedeny na jeden výsledný vektor zrychlení pomocí vektorového součinu. Jelikož směr pohybu není pro detekci kontextu směrodatný, není třeba osy rozlišovat. Výhodné je také to, že se není nutné zabývat filtrací gravitačního zrychlení, akcelerometr v Android OS umí běžet v režimu, který toto zrychlení rovnou filtruje. Dále byla od všech hodnot odečtena jejich stejnosměrná složka a následně byla na data aplikována Rychlá fourierova transformace [11].

### 3.4.3 Naměřená data

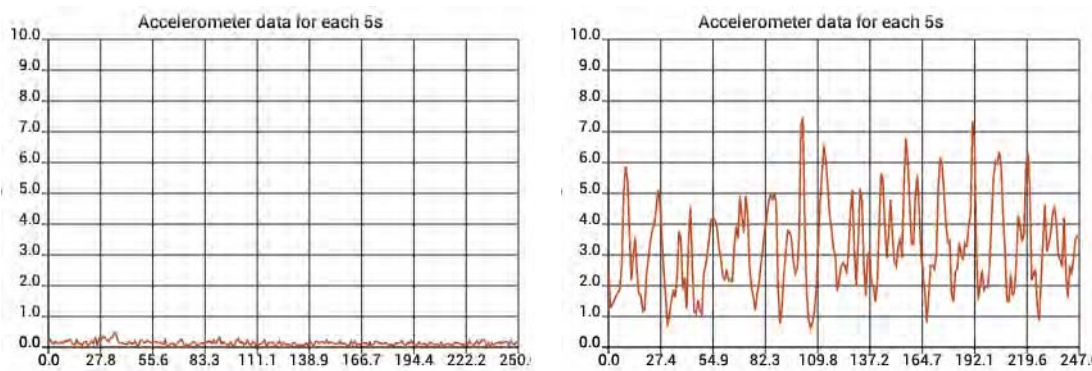
Z dat, která byla získána při testování prototypu, bylo zjištěno několik zajímavých faktů o signálu, který akcelerometr snímá během chůze, a to zejména to, že se značně liší jeho průběh v závislosti na poloze zařízení během chůze. Při přenášení zařízení v kapse je frekvence zhruba poloviční než při chůzi s telefonem v ruce, nepřímo úměrně se liší velikosti amplitud signálu. Na základě těchto měření musel být mírně upraven způsob filtrace výstupních dat. Původně byla data filtrována na základě kontroly rozmezí frekvence a následné kontroly velikosti amplitudy kvůli zajištění vyšší přesnosti detekce. Toto bylo upraveno a na základě toho, že hodnoty amplitudy signálu a frekvence si byly prakticky nepřímo úměrné, bylo přidáno spočtení součinu frekvence a amplitudy. Výsledná hodnota byla měřena pro různé polohy zařízení a nakonec bylo zvoleno dostatečné rozmezí tak, aby byla korektně rozpoznána chůze ve všech případech. Výsledkem je vcelku spolehlivá detekce kontextu uživatele, při testování prototypu se vyskytlo pouze velmi nízké procento chybných detekcí.

Data byla vizualizována přímo v zařízení pomocí grafu tak, aby bylo možné kontrolovat detekci v reálném čase. Na obrázcích 3.1 je několik grafů zrychlení pro jednotlivé kontexty pohybu. Na ose  $x$  vyjadřují čísla index zobrazené hodnoty v poli a na ose  $y$  je zobrazena velikost výsledného vektoru v  $m/s^{-1}$ .

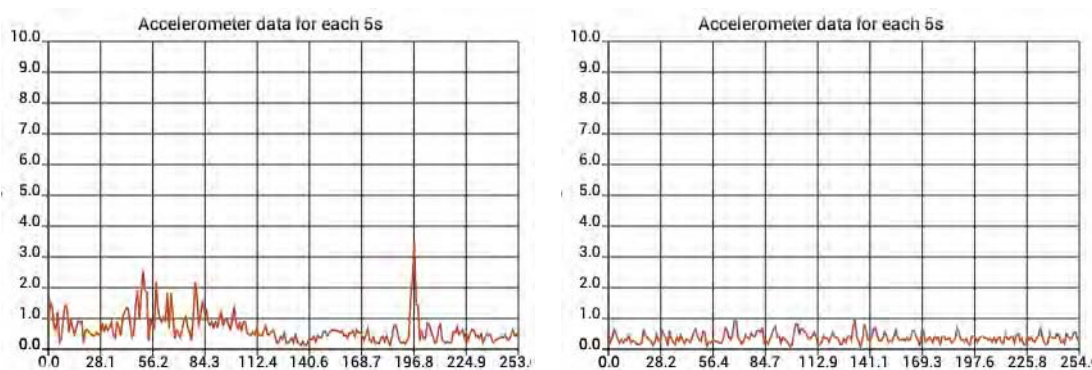
### Porovnání průběhů pro jednotlivé typy kontextu

Na následujících čtyřech obrázcích je vidět tvar křivky signálu zrychlení pro jednotlivé typy kontextu. Jako první je křivka, jejíž hodnoty byly naměřeny telefonem položeným na stole. Hned napravo je zobrazen graf dat naměřených při chůzi, na něm je krásně vidět rozdílný průběh signálu a je tam vidět i jasná perioda. To jsou přesně vlastnosti, kterých je využíváno pro detekci kontextu. Další dva obrázky zobrazují data naměřená při jízdě v MHD, konkrétně v autobuse a tramvaji. Graf zobrazující data naměřená v autobuse má amplitudy dost vysoké, to je tím, že měření bylo prováděno při jízdě po dlažebních kostkách, jízda tedy nebyla zrovna klidná. Oproti tomu graf z tramvaje jasně ukazuje, že jízda po kolejích nevydává žádné velké rázy, místy se hodnoty blížily těm v klidovém stavu.

Obrázek 3.1: Graf vektorů zrychlení zařízení položeného na stole a při chůzi (zleva)



Obrázek 3.2: Graf vektorů zrychlení při jízdě autobusem po dlažebních kostkách a v tramvaji (zleva)



## 3.5 Určení polohy uživatele

Určení polohy uživatele je klíčovou součástí aplikace, proto byl kladen důraz na pečlivé promyšlení toho, jak budou data o poloze získávána a zpracována.

### 3.5.1 Způsoby určování polohy

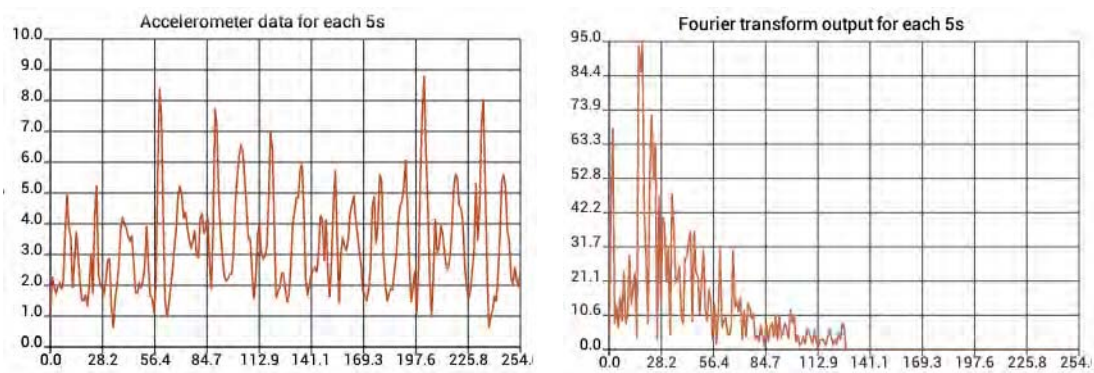
Operační systém Android poskytuje rozhraní pro zjištění aktuální polohy zařízení přímo jako součást SDK. K dispozici jsou dva základní způsoby jak polohu získávat. První způsob, určení polohy na základě dat z GPS [3.5.2](#), je přesnější, ale prvotní získání polohy trvá delší dobu. Druhý způsob, který se v OS Android nazývá network provider, určuje polohu pomocí triangulace dat o síle signálu blízkých vysílačů GSM a Wi-Fi. Způsob je to rychlejší, ale přináší s sebou mnohem nižší přesnost, v ideálních podmínkách na několik desítek metrů, většinou ale až v řádech stovek metrů [\[10\]](#). Jelikož pro využití v aplikaci jsou potřeba data co nejpřesnější, byl jako zdroj dat zvolen systém GPS.



### Graf vstupních dat a výstupu Fourierovy transformace a jeho interpretace

Na následujících dvou obrázcích je průběh naměřený při chůzi a jemu odpovídající výstup Fourierovy transformace. Graf Fourierovy transformace zobrazuje na ose  $x$  frekvenci, s jakou se v prvním grafu opakují amplitudy, které jsou v druhém grafu zobrazeny na ose  $y$ . Frekvence není na ose  $x$  zobrazena v odpovídajících jednotkách, je třeba použít přepočít, kdy vzorkovací frekvenci vydělíme délkou vstupního pole a výsledný podíl vynásobíme daným indexem, jehož frekvenci počítáme. Zde je krásně vidět, že délka výstupního pole je přibližně 250 hodnot. Nejvyšší amplituda je přibližně u hodnoty 20, vzorkovací frekvence je 50 Hz, takže máme:  $50/250 * 20 = 4$ . Když se podíváte do grafu vstupních dat, tak je vidět, že je-li délka okna 4 sekundy, tak počet vln v grafu odpovídá číslu 16, tedy 4 opakování za sekundu.

Obrázek 3.3: Graf vektorů zrychlení při chůzi a graf odpovídajícího výstupu Fourierovy transformace (zleva)



### 3.5.2 Global positioning system

Global Positioning System (dále GPS) je navigační systém původně určený čistě pro vojenské účely americké armády, s jehož pomocí je možno určit přesnou polohu a čas kdekoli na povrchu Země. Jeho provozovatelem je Ministerstvo obrany Spojených států amerických.

Systém funguje na bázi vesmírných družic pohybujících se po oběžné dráze planety Země. Původně bylo družic 18, počet se poté rozrostl na zamýšlených 24 a nakonec je dnes družic 32 (ne všechny jsou aktivní), což je také teoretické maximum systému, při dalším zvýšení počtu satelitů by byla nezbytná úprava celého systému.

Za pomoci GPS je možno přesně určit polohu s přesností v řádech desítek až jednotek metrů, v případě diferenciální GPS je dokonce dosaženo přesnosti v řádech centimetrů.

Systém GPS byl původně určen pouze pro vojenské účely Spojených států amerických, ale roku 1983 bylo rozhodnuto, že se systém po svém dokončení dostane i mezi veřejnost. Toto předsevzetí bylo splněno až v roce 2000, kdy byla plně zrušena selektivní dostupnost signálu GPS, i díky tomu, že USA přišly na způsob, jak na zvoleném místě rušit signál vysílaný družicemi systému GPS.

### 3.5.3 Aspekty použití GPS

Global positioning system poskytuje v ideálních podmínkách spolehlivou a přesnou metodu pro zjištění polohy uživatele, jeho použití však s sebou přináší několik problémů. Například může docházet k odrazům signálu GPS a s tím související ztrátě přesnosti určení polohy. Dalším problémem je vysoká spotřeba zařízení při zapnuté GPS, kdy výdrž telefonu za nepřetržitě aktualizované polohy přes GPS klesne do řádu jednotek hodin, zařízení tak nevydrží ani jeden celý den bez připojení k napájecímu adaptéru. Dalším problémem GPS je nutné zachycení satelitů pro určení polohy, které může trvat v nepříznivých podmínkách i několik desítek minut. Existují sice podpůrné technologie pro dosažení rychlejšího zaměření, ty jsou však vázány na datové připojení, popřípadě příjem radiového signálu. z těchto důvodů je potřeba použití GPS optimalizovat například kombinací s detekcí kontextu 3.4, kdy se bude GPS spouštět pouze v případech, kdy bude aplikace detekovat chůzi.

Akcelerometr použitý pro detekci kontextu totiž na rozdíl od GPS poskytuje data okamžitě po zapnutí a stačí krátké měřicí okno pro nasbírání objemu dat potřebného pro správnou detekci kontextu.

## 3.6 Optimalizace spotřeby

Spotřeba elektrické energie je při návrhu mobilních aplikací velice důležitým faktorem, kterému je nutno přiřadit vysokou prioritu. v případě této práce pak hraje prakticky hlavní roli, aplikace která je předurčena k neustálému běhu na pozadí nesmí příliš snižovat dobu běhu na baterie, uživatelé by ji jinak přestali používat. Obecně je nízká výdrž jednou z hlavních nevýhod současných chytrých telefonů. Ty při konstantně zapnuté GPS dosahují výdrže maximálně řádu několika málo hodin, neustálé připojení k internetu zařízení má na spotřebu zařízení také velmi negativní vliv.

### 3.6.1 Lokace uživatele

Aktivní GPS je jednou z hlavních složek celkové spotřeby telefonu, například GPS v Nokia N95 má spotřebu přibližně 0.3 Wattu, přitom klidová spotřeba tohoto telefonu je pouze kolem 0.02 Wattu [6]. Navýšení spotřeby pouze samotnou GPS je tedy patnáctinásobné. z toho vyplývá nutnost maximálního zkrácení doby, po kterou musí být GPS aktivní. Zde je nutno brát v potaz ještě dostatečnou přesnost informací. Pokud je zařízení v pohybu, je třeba získávat data o poloze dostatečně často, aby bylo možno určit, kudy přesně uživatel šel. Při předpokládané rychlosti pohybu uživatele 1-2 m/s bylo jako interval pro jednotlivá měření zvoleno 5 sekund, za které se uživatel přesune o 5-10 metrů.

Zajímavé je, že GPS řídí svou činnost částečně nezávisle na potřebách samotné aplikace. Pokud si aplikace vyžádá polohu získanou pomocí GPS a ta je aktuálně vypnutá, systém pro získání polohy nejprve musí spočítat frekvence jednotlivých satelitů. To se dá urychlit pomocí stažení předpokládaných frekvencí pomocí datových přenosů. Bližší info v kapitole o GPS 3.5.2. i poté se ale GPS přijímač musí naladit, přesné frekvence jsou totiž ovlivněny například i Dopplerovým efektem. z tohoto důvodu zůstává GPS zapnutá i přibližně ještě 30 sekund poté, co aplikace přestane žádat o aktualizace polohy. Opětovné zaměření satelitů totiž po 30 sekundách neaktivity trvá v průměru kolem 5 sekund [6]. Prakticky tedy nemá

cenu GPS vypínat, pokud jsou plánované intervaly mezi aktualizacemi polohy nižší než 30 sekund. Informace o spotřebě při používání GPS a zaměření na její minimalizaci byly čerpány z článku o projektu EnTracked, který se zabýval právě optimalizací spotřeby při určování polohy při zachování dostatečné přesnosti [6].

Cílem této práce je sledování polohy uživatele pouze ve specifických případech pohybu při chůzi, takže nebylo nutné řešit sledování polohy z dlouhodobého hlediska (v horizontu dní), ale stačilo monitorovat polohu ve chvílích, kdy uživatel jde. Pro detekci pohybu v reálném čase se z hlediska spotřeby a i přesnosti ukázal nejlepší akcelerometr.

### 3.6.2 Detekce pohybu a jeho kontextu

Pro detekci kontextu byl použit akcelerometr. Ten sám o sobě nemá příliš vysokou spotřebu, ta se pohybuje přibližně na úrovni spotřeby zařízení v pohotovostním režimu (tedy cca 0.02W [6]). Větším problémem se ukázalo samotné zpracování výstupních dat. Při nejvyšší možné frekvenci získávání aktuálních dat (přibližně 50Hz) a jejich následném zpracování pomocí rychlé Fourierovy transformace se ukázalo jako celkem náročné. Výdrž zařízení klesala na úroveň výdrže při zapnuté GPS. Dalším problémem je, že sice samotný akcelerometr nemá vysokou spotřebu, pokud ale běží, tak není možné, aby zařízení přešlo do režimu stand-by. Celková spotřeba zařízení se tedy při použití akcelerometru navýší celkem mnoho. Přesné rozdíly jsou vidět ve výsledcích testů spotřeby, které byly provedeny. Výsledky jsou v kapitole o testování 5.2.2. Jelikož pro samotnou detekci pohybu není nutné data zpracovávat Fourierovou transformací a není ani potřeba aktualizovat data tak vysokou frekvencí, byl proces detekce kontextu rozdělen na dvě části. v první části se zjistí, jestli se zařízení hýbe. To se provádí jednoduchým porovnáním velikosti vektoru zrychlení, který je získán vektorovým součtem hodnot zrychlení na všech třech osách. Pokud tento vektor přesáhne určenou hranici, tak se akcelerometr přepne na maximální výkon a na hodnoty sesbírané během 4 sekundových intervalů je vždy aplikována Fourierova transformace. Spočtený kontext se vyhodnotí jako validní, pokud se zopakuje minimálně dvakrát po sobě. To má za úkol snížit pravděpodobnost falešné detekce chůze. Tímto dvoukrokým způsobem detekce kontextu se výrazně podařilo snížit spotřebu aplikace.

### 3.6.3 Odesílání dat na server

Odesílání dat o poloze na server je také velice energeticky náročná činnost. z principu fungování aplikace naštěstí není nutné posílat data vždy okamžitě, i s takovou možností bylo ovšem původně počítáno. Při využívání datových přenosů určují výslednou spotřebu dva základní faktory (pokud není brána v potaz technologie připojení), a to množství odeslaných dat a délka intervalů mezi jednotlivými připojeními na server. Při každém vytvoření nového připojení se přijímač přepne do režimu vysokého výkonu a poté se zahájí přenos dat. Tento přechod trvá přibližně 2 sekundy. Pro minimalizaci prodlevy sítě je tedy spojení ponecháno otevřené a přijímač běží na plný výkon po dalších 5 sekund. Poté přijímač dalších 12 sekund setrvá v režimu nižšího výkonu, předtím než se přepne do pohotovostního režimu. Výsledkem je tedy 20 sekund v režimu s vyšší spotřebou pro každý přenos [14]. Logicky je tedy při snaze o co nejnižší spotřebu aplikace nevhodné, aby se data posílala často a po malých částech. Výhodnější je posílat je v co nejdelších možných intervalech po větších částech najednou.

Ideální je například vyčkat na připojení zařízení do zdroje elektrické energie a až poté začít přenos. Podrobnější informace jsou v sekci testování [5.2.2](#).

### 3.6.4 Spotřeba aplikace jako celku

Při návrhu architektury aplikace a způsobu měření dat byla brána jako hlavní měřený faktor výsledná spotřeba v poměru ku přesnosti dat. Díky nízké spotřebě akcelerometru v pohotovostním režimu a naopak vysoké spotřebě GPS přijímače za běhu byl navržen stavový automat aplikace.

V základním režimu běží pouze akcelerometr a je nastavena co nejnižší možná frekvence obnovování dat, to znamená pouze několik vzorků za sekundu. Pokud je opakovaně po sobě naměřeno zrychlení větší než určená mez, přepne se akcelerometr do režimu s frekvencí 50Hz a je zjištěn kontext pohybu. Blíže je to popsáno v kapitole [3.6.2](#). Pokud výsledný průběh pohybu odpovídá průběhu chůze, tak se aktivuje GPS a začnou se ukládat data o poloze uživatele. Spolu s tím se akcelerometr přepne zpět do úspornějšího režimu a způsob detekce se obrátí. Pokud se několikrát po sobě naměří podhraniční zrychlení, tak se znovu spustí detekce kontextu. Během chůze se totiž zrychlení pohybuje kontinuálně nad touto hranicí, je tedy v tuto chvíli pravděpodobné, že se uživatel zastavil, případně změnil kontext pohybu. Jestliže se vyhodnotí průběh signálu jako chůzi neodpovídající, tak se zastaví monitorování polohy a vypne GPS. Další informace jsou v sekci doporučení do dalších fází projektu [5.3.1](#).

### 3.6.5 Optimalizace množství přenesených dat

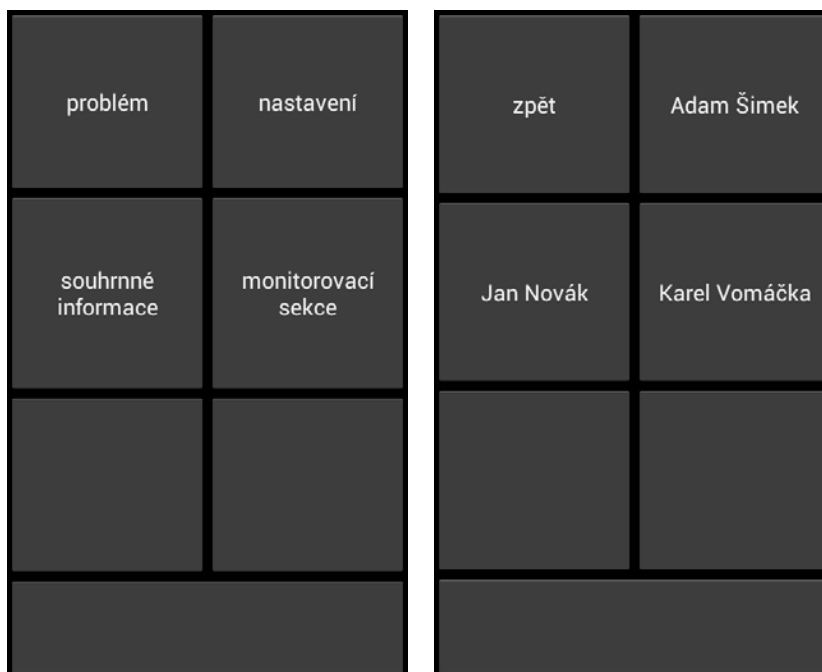
Minimalizace množství přenesených dat je také velice důležité. Členové cílové skupiny nemusí mít nutně paušální datový tarif, případně mohou mít data vypnutá úplně. S tím vším musí aplikace počítat. Jelikož součástí řešení není finální serverová část a odesílání dat přes datové přenosy bylo využíváno pouze pro testovací účely, nakonec tomu nebyl dán takový důraz. Rozhodně to ale patří mezi jeden z hlavních bodů doporučení do případných dalších fází projektu a je to zmíněno v kapitole [5.3.1](#). Data byla v rámci analýzy návrhu a testování [5.2.2](#) odesílána pomocí protokolu HTTP v reálném čase ihned po naměření. Přenášena byla datová struktura ve formátu JSON [\[1\]](#), jedna položka v sobě zahrnovala identifikační klíč uživatele, jeho aktuální zeměpisnou polohu a přesný čas měření. Co se týče možné optimalizace, uvažováno bylo prodloužení intervalů posílání dat na server a komprimování dat metodou gzip před odesláním HTTP požadavku.

## 3.7 Uživatelské rozhraní

Uživatelské rozhraní aplikace bylo třeba navrhnout s ohledem na cílovou skupinu, která má vzhledem ke svému postižení potíže s ovládáním dotykových zařízení kvůli chybějící hmatové odezvě.

Cílem tedy bylo zjednodušení celého uživatelského rozhraní tak, aby samotná orientace mezi ovládacími prvky činila uživateli minimální potíže. Výsledkem je matice šesti přibližně čtvercových polí, které jsou uspořádány v řádcích po dvou, celkem ve třech sloupcích. Mezi poli je vytvořena mřížka pomocí haptické odezvy (vibrace telefonu), takže uživatel má při přejíždění prstem po displeji odezvu o tom, když překročí hranici mezi poli. Každému poli

Obrázek 3.4: Ukázka uživatelské rozhraní aplikace. Zleva nejprve výchozí nabídka, dále nabídka vyhledání pomoci



je přiřazena právě jedna akce, která se při prvním stisku, nebo po přejetí prstem přes pole, uživateli oznámí pomocí syntézy řeči. Při druhém stisku stejného pole po sobě se zvolená akce vybere a vykoná. Vše je znovu doplněno o odezvu předanou uživateli pomocí syntézy řeči. Cílem nebylo pouze o zjednodušení uživatelského rozhraní, ale i o celkové zjednodušení ovládání aplikace. To spočívá v minimalizaci nutnosti interakce uživatele s aplikací. Uživatelské rozhraní by mělo být potřeba ideálně pouze při prvním přihlášení a konfiguraci a dále při vyhledání pomoci. Odstraněna byla veškeré přebytečné nastavení a volby, které uživatel nutně nepotřebuje.

V průběhu testování návrhu UI přímo s uživateli z cílové skupiny bylo třeba udělat několik úprav. Velikým problémem jsou systémové klávesy (pouze pokud jsou dotykové - hardwarové klávesy problém nedělají). Tyto dotykové klávesy mají v OS Android pevně přiřazené funkce, které není možné vždy přepsat, takže se stávalo, že se uživatelé například omylem vraceli na domovskou obrazovku. Toto má bohužel prakticky jediné řešení v podobě výběru přístroje, který má systémové klávesy hardwarové, nikoliv kapacitní. Na základě zpětné vazby získané z uživatelských testů low-fidelity prototypu 5.1 byla přidána haptická odezva i na okraje obrazovky a mezi šest hlavních kláves a řadu systémových dotykových kláves byl přidán pruh, který zamezuje nechtěným stiskům těchto kláves v případě zamýšleného stisku jednoho ze dvou tlačítek nejspodnější řady. Při dotyku prstem i při najetí prstem na pruh telefon vibruje stejně jako v případě najetí k okraji obrazovky.

Dalším problémem jsou dnes tolik oblíbené kapacitní dotykové displeje. Ty jsou rozšířeny kvůli vysoké citlivosti, která je pohodlná při ovládání prstem. v případě zrakově postižených

osob je vysoká citlivost naopak nevýhodná, displej totiž reaguje často prakticky ještě, než se uživatel dotkne obrazovky, a zrakově postižená osoba, orientující se při ovládání zařízení zejména podle hmatu tak neví, jestli se dotkla, nebo nedotkla. Tento problém byl odstraněn přidáním haptické odezvy ke každému dotyku displeje.

## 3.8 Operační systém Android

Operační systém Android (dále pouze OS Android) je open-source projekt založený na Linuxovém jádře, určený pro běh na mobilních zařízeních s dotykovou obrazovkou. Vyvíjen je společností Google Inc., v současnosti aktuální verze je 4.0 Ice Cream Sandwich, ale aplikace bude cílena primárně na starší verzi 2.3 Gingerbread, která je nyní nejvíce rozšířená. Jednotlivé verze jsou mezi sebou navzájem kompatibilní, takže aplikace jsou schopny běhu na vyšších verzích systému. Běh na nižších verzích může být limitován v případě použití API, která nebyla v těchto verzích ještě k dispozici.

OS Android se jeví jako vhodná platforma pro realizaci této práce, zejména z důvodu širokého záběru na trhu s mobilními telefony. Zařízení s Androidem se dá koupit takřka ve všech cenových kategoriích, což je pro náš projekt výhodou. Další velkou výhodou je cenová politika aplikací na platformě Android. Zatímco řešení pro platformu Symbian stojí několik tisíc korun [5], tak například čtečka SVOX i s modulem pro češtinu stojí 55 Korun českých [7].

### 3.8.1 Architektura systému

OS Android je rozdělen do hlavních vrstev, z nichž každá má specifický účel.

- Jádro systému
- Android application framework
- Android runtime
- Application framework
- Uživatelské aplikace

### 3.8.2 Programování pro OS Android

Aplikace pro OS Android jsou vytvářeny buďto v programovacím jazyce Java pomocí standardního Android Software Development Kitu (dále jenom Android SDK), nebo v případě potřeby vyššího výkonu - například pro tvorbu 3D her - pomocí Android Native Development Kitu v jazyce C++. K vývoji byl použit standardní Java SDK, je to obvyklejší cesta tvorby aplikací a vývojářské nástroje jsou mnohem pokročilejší.

### Architektura aplikace a programovací rozhraní

Aplikace na platformě Android se sestávají ze dvou základních stavebních kamenů. Jsou to aktivity – potomci třídy `Activity.class` – a služby – potomci třídy `Service.class`. Zásadní rozdíl mezi aktivitou a třídou je, že aktivita má představovat jeden určitý pohled UI aplikace, tedy je reprezentací toho, co se uživateli zobrazí na displeji telefonu. Oproti tomu služby jsou určeny pro běh na pozadí, případně pro běh jiného kódu, který přímo nepřísluší do uživatelského rozhraní. Aktivity jsou po opuštění pozastaveny, jejich stav se dále drží v operační paměti, ale systém nám nezaručí, že tomu tak zůstane. Pokud je paměti nedostatek, tak aktivitu z paměti odstraní a ta se tak musí po dalším spuštění inicializovat znovu. Toto chování je doplněno o patřičná zpětná volání, která mohou vývojáři využít. Služby běží po opuštění aplikace dál, dokud jsou nezbytné nebo dokud je neukončíme. Dalším důležitým systémovým blokem je `Intent`, což je vlastně obslužný objekt určený ke spouštění jiných aktivit a služeb. `Intentu` je možno přidávat další data, která se pošlou do cílové aktivity Pro vývojáře jsou připraveny aplikační rozhraní pro komunikaci s hardwarem telefonu (sensory) nebo například pro ukládání dat (SQLite databáze).

### Instalace Android SDK

Instalace Android SDK je bezproblémová, Google poskytuje ke stažení [8] kompletní SDK a zásuvný modul do vývojového prostředí Eclipse. Dají se použít i jiná IDE, je možno si SDK nainstalovat i bez Eclipse IDE a vše poté řešit přes ADB (android debug bridge), což je nástroj pro příkazovou řádku, přes který se dají do telefonu instalovat vývojářské aplikace.





## Kapitola 4

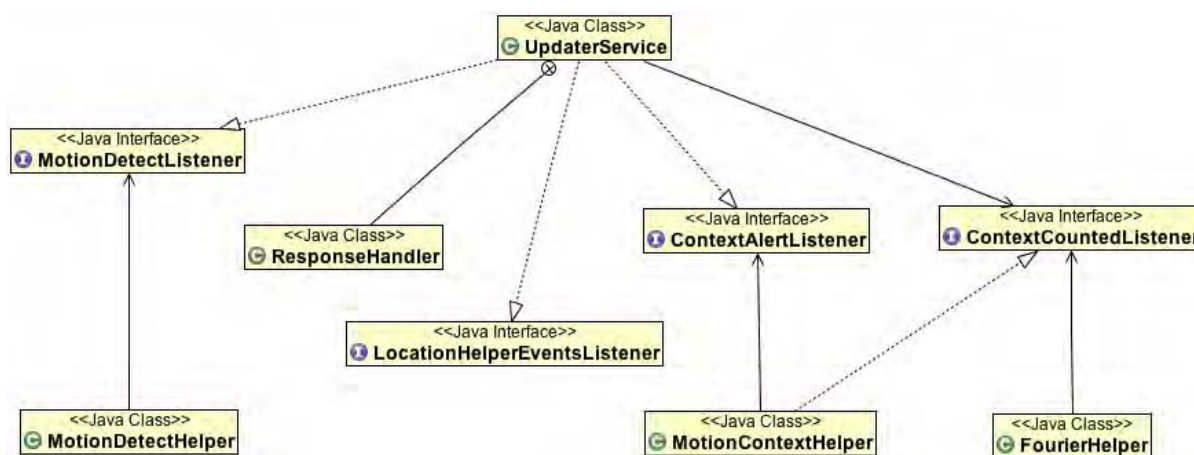
# Realizace

### 4.1 Úvod

Tato kapitola popisuje průběh realizace této práce. Cílem bylo vytvořit funkční aplikaci tak, aby mohlo být provedeno veškeré nutné testování.

### 4.2 Architektura aplikace

Aplikace je rozdělena do šesti základních balíčků, které obsahují související třídy.



Obrázek 4.1: Diagram tříd zobrazující architekturu aplikace (zahrnuje pouze vybrané třídy z důvodu zachování čitelnosti)

V balíčku *activities* jsou jednotlivé aktivity aplikace, to není ničím zajímavé. z pohledu této práce je zajímavý obsah balíčku *helpers*, který obsahuje pomocné třídy pro výpočet Fourierovy transformace, detekci pohybu, detekci kontextu, také je tam pomocná třída pro převod textu na řeč. v balíčku *interfaces* jsou rozhraní používaná pro komunikaci mezi třídami. Balíček *services* obsahuje službu, která se stará o běh aplikace na pozadí, a je to vlastně taková základní třída celé aplikace. v balíčku *ui* jsou potom definovány speciální třídy pro upravené uživatelské rozhraní.

### 4.3 Uživatelské rozhraní

Implementace uživatelského rozhraní nebyla úplně triviální. Díky vizuální jednoduchosti by se mohlo zdát, že bude snadné ho vytvořit, díky nestandardním ovládacím prvkům a zejména nestandardním akcím, které jsou na UI navazovány, musel být kód několikrát celý přepsán. Cílem byla také modulárnost a znovupoužitelnost veškerého kódu, stejný princip je totiž použit v rozdílných kontextech ovládání, bylo tedy třeba zachovat flexibilitu a dostatečnou míru abstrakce.

Obrázek 4.2: Ukázka základního pohledu na UI



### 4.4 Detekce uživatelského kontextu

Detekce uživatelského kontextu je implementována pomocí pomocné třídy *MotionContextHelper.java*, která zpracovává data z akcelerometru. Implementována je pomocí návrhového vzoru Singleton. Sama si řídí spouštění akcelerometru včetně nastavení jednotlivých frekvencí sběru dat. Nasbíraná data poté vždy předá dál ke zpracování a jako výstup dostane aktuální naměřený kontext. Měření probíhá vždy několikrát ve smyčce, dokud není kontext dostatečně ověřen. V ukázce kódu 4.1 je zobrazen kód metody *startContextResolve()*, která spouští detekci kontextu. Po spuštění měření je přiřazeno zpětné volání, které je zavoláno po zjištění aktuálního kontextu. Proces zjištění kontextu totiž nějakou dobu trvá, musí to proto být provedeno asynchronně.

#### 4.4.1 Výpočet rychlé Fourierovy transformace

Hlavním krokem nutným pro zjištění kontextu je výpočet rychlé Fourierovy transformace pro převedení signálu z domény časové na frekvenční. Pro výpočet byla použita volně dostupná open-source knihovna JTransforms [16], která je podle testů [13] nejrychlejší dostupnou implementací FFT v Javě. Knihovna pracuje in-place to, takže pole, nad kterým je zavolána, přímo transformuje a hodnoty změnil na hodnoty výstupu FFT. Jelikož výstupem FFT jsou komplexní čísla, ve výsledném poli jsou vždy na sudém indexu reálné hodnoty čísel

Listing 4.1: Ukázka kódu metody spouštějící detekci kontextu

```
1  public void startContextResolve(ContextAlertListener ccl){
2  list = new ArrayList<Float>();
3  this.setOnContextAlertListener(ccl);
4  FourierHelper.setOnContextCountedListener(this);
5  registerAccelerometerUpdates();
6
7  contextUpdatehandler = new Handler()
8      {
9      @Override
10     public void handleMessage(Message msg)
11     {
12         Bundle dataBundle = new Bundle();
13         float vectors[] = new float[list.size()];
14         int i = 0;
15
16         Iterator<Float> inputConverter = list.iterator();
17
18         while(inputConverter.hasNext()){
19             vectors[i++] = inputConverter.next();
20         }
21         list.clear();
22
23         dataBundle.putFloatArray("vectors", vectors);
24         msg.setData(dataBundle);
25         FourierHelper.processFourierData(msg);
26     }
27 };
28
29 Message msg = Message.obtain();
30 msg.arg1 = COUNT_CONTEXT_FOREGROUND;
31 contextUpdatehandler.sendMessageDelayed(msg, 4000);
32
33 }
```

a na lichém indexu je imaginární část komplexního čísla. Bylo nutné zjistit velikost těchto čísel, ta se určí jako druhá odmocnina ze součtu druhé mocniny reálné a druhé mocniny imaginární části každého čísla. Jde vlastně o absolutní hodnotu imaginárního čísla, podle vztahu 4.3, kde  $a$  je hodnota z každého sudého indexu výstupního pole FFT a  $b$  hodnota každého následujícího lichého indexu. Těmito hodnotami se dále naplní nové pole, index

$$|a - bi| = \sqrt{a^2 - b^2}$$

Obrázek 4.3: Vzorec pro výpočet absolutní hodnoty imaginárního čísla

určuje frekvenci, hodnota určuje velikost amplitudy. Frekvence se z indexu zjistí, když index vydělíme délkou pole a vynásobíme vzorkovací frekvencí. Lze tedy snadno určit, jaká amplituda se nám opakuje s jakou frekvencí.

## 4.5 Lokace uživatele

V systému Android je přístup k datům GPS velmi snadný. Je to vidět v následující ukázce kódu. Stačí vytvořit instanci třídy `LocationManager` a dále specifikovat, jaký způsob získání polohy chceme použít, a určit interval, v jakém chceme polohu aktualizovat.

Listing 4.2: Ukázka kódu použitého pro získání GPS polohy

```

1   LocationManager locationManager = (LocationManager) this.getSystemService(
      Context.LOCATION_SERVICE);
2   locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5, 0,
      this);
```

Instanci `LocationManageru` poté stačí přiřadit instanci třídy implementující rozhraní `LocationListener`, jehož metody budou volány jako zpětná volání akcí vyvolaných `LocationManagerem`. Zpětné volání nám vrátí instanci třídy `Location`, která obsahuje informace o aktuální poloze, přesnosti, nebo například rychlosti pohybu zařízení. Pro ukončení získávání dat z GPS stačí odregistrovat zpětné volání.

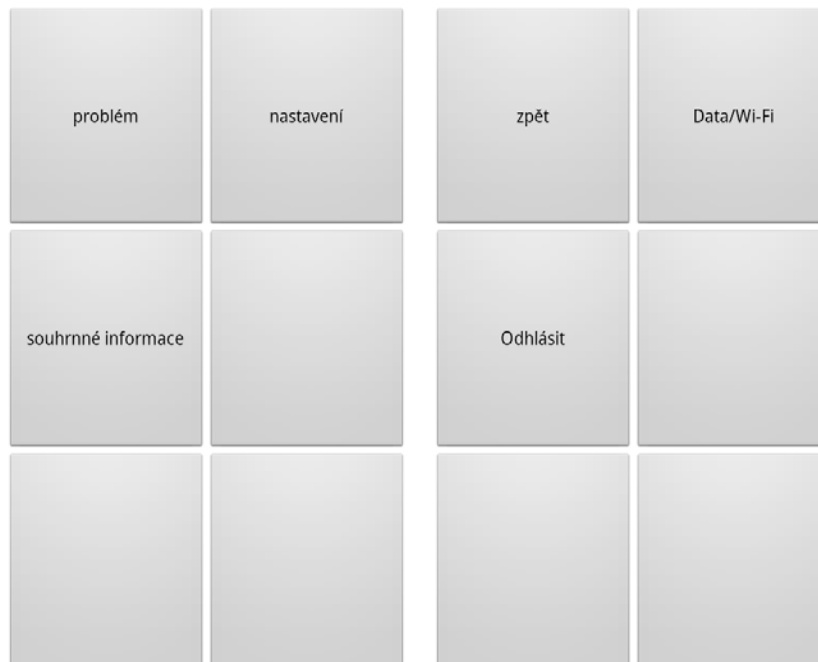
## Kapitola 5

# Testování

### 5.1 Testování low-fidelity prototypu

#### Úvod

Při testování low-fidelity prototypu bylo získáno velké množství cenných dat a myšlenek, které mohly být dány dohromady a pro druhé kolo testování implementovány v rámci již složitějšího prototypu, ve kterém se uživatel mohl pohybovat v několika úrovních, volit možnosti z nabídky, nebo se vracet o úroveň výše. Na obrázcích [5.1](#) je vidět výsledná podoba testovaného prototypu.



Obrázek 5.1: Uživatelské rozhraní prototypu

### 5.1.1 Popis testování

Test proběhl se dvěma participanty, tedy tak akorát, aby byly splněny podmínky a byly načerpány potřebné informace. Pouze se ve výsledku ukázal jako relativně krátký interval 30 minut, který byl vyhrazen pro každý testovací blok.

Pro test byl využit speciálně upravený telefon Samsung Nexus S. Kvůli dotykovým klávesám, kterými se tento telefon, běžící na platformě Android, ovládá a které se při prvním kole testování ukázaly jako prakticky nepoužitelné pro zrakově postižené uživatele, musel být blok dotykových tlačítek pro testování přelepen páskou, aby se jejich citlivost snížila na přijatelnou mez a nedocházelo k nechtěným stiskům.

Pro participanty byly připraveny dva formuláře - vstupní a výstupní. Obsahovaly několik otázek ohledně jejich zkušeností s mobilní technikou, případně jejich vztahu k ní a několik dalších obecných otázek.

Níže je seznam otázek a odpovědí jednotlivých participantů jak pro vstupní, tak výstupní dotazník.

#### Vstupní dotazník

Otázky pokládané v rámci vstupního dotazníku jsou uvedeny v tabulce 5.1

test	Participant 1	Participant 2
Z jakého důvodu se účastníte testování?	Vnitřní motivace - několik let testerem	Vnitřní motivace
Vlastníte mobilní telefon?	Ano	Ano
Používáte telefon s dotykovým displejem?	Ano	Ne
Máte s využíváním dotykových displejů (nebo jiných dotykových zařízení) nějakou zkušenost?	Ano	Ano
Používáte ve svém telefonu datový tarif?	Ano	Data využívám příležitostně, ale nemám tarif

Tabulka 5.1: Vstupní dotazník

#### Výstupní dotazník

Otázky pokládané v rámci výstupního dotazníku jsou uvedeny v tabulce 5.2

#### Testovací setup

Test probíhal v prostorách ULABu ČVUT v Praze na Karlově náměstí, níže jsou vidět fotografie místnosti, ve které testování probíhalo.

	Participant 1	Participant 2
Jak se Vám testování líbilo	Líbilo	Líbilo
Měl(a) jste problém s vykonáním zadaných úkonů, činilo Vám to subjektivně potíže?	Rozhodně ne, bylo to lehké	Spíše ne
Jaké vám připadalo uživatelské rozhraní testované aplikace	Ovládání je celkem snadné	Ovládání je celkem snadné
Dovedete si představit použití takového rozhraní v běžném životě (na ulici, doma, ..)	Bez nejmenších potíží	Ano, ale má to zatím „mouchy“
Co se Vám na aplikaci líbilo a co byste naopak změnil(a)?	+vibrace mezi tlačítky +rychlá reakce hlasu na klik -přidání mezikroku k vytáčení telefonního čísla	+přerušování text-to-speech -chtělo by to přidat mezery mezi tlačítky
Přišel/přišla byste na testování znovu?	Ano	Ano

Tabulka 5.2: Dotazník po testu

Jako zařízení pro testování byl použit Samsung Nexus S, speciální úpravy jsou popsány v úvodu sekce.

### 5.1.2 Průběh testování

Testování probíhalo celkem se dvěma participanty v blocích po 30 minutách. Každý blok byl jiný den, tedy mohly být pro test s druhým participantem do aplikace zahrnuty připomínky, které měl k prototypu první participant, a s dalším participantem již mohla být otestována lehce vylepšená verze aplikace. Pro testování byl připraven krátký seznam uživatelských scénářů, ale během testování se ukázalo, že velmi limitovaný čas, který je pro testování k dispozici, je lepší využít pro abstraktní diskuzi nad prototypem s participantem než pro pouhé plnění, v této fázi projektu stejně velice primitivních, případů užití. Na začátku každého testu bylo participantovi řečeno pár slov o této práci, mezitím měl participant k dispozici telefon bez spuštěné aplikace. Následovalo krátké shrnutí, co může od uživatelského rozhraní očekávat, načež byla na na telefonu spuštěna aplikace a začalo samotné testování. Oba participanti nejdřív celou aplikaci prozkoumali, což byl také první případ užití, ale během toho již začali vznášet připomínky, co by se dalo vylepšit, a komentovat, co se jim líbí. Níže bude detailnější popis průběhu testu s každým participantem zvlášť



Obrázek 5.2: Testovací setup



Obrázek 5.3: Nexus S



## Participant 1

Participantem byl velice pokročilý uživatel, který aktivně používá čtyři mobilní telefony, takže šlo o experta z oboru. Mimo jiné se několik let aktivně podílí na testování aplikací pro nevidomé. První kontakt s aplikací byl bezproblémový, přestože začal aplikaci zkoušet ještě před řádným zapnutím. Vyzdvihoval zejména rychlé reakce syntézy řeči na akce uživatele a vůbec její dobré chování, to prý není u konkurence standardem.

Hlavní připomínkou bylo, aby aplikace četla akce jednotlivých tlačítek už při hoveru, tedy přejíždění prstem po nich. Toto fungovalo pouze při kliku. Další připomínku měl k moc citlivým reakcím kapacitního dotykového displeje, který občas reaguje, aniž by se ho uživatel pořádně dotknul, a aplikaci chyběla haptická odezva na dotyk displeje (z důvodu obav z překombinování při společné implementaci s vibrační mřížkou mezi tlačítky).

Jediný problém nastal, když se uživatel dostal k vytáčení telefonního čísla. Tam se dostal do kontaktu se standardní aplikací pro telefonní hovory bez jakékoliv optimalizace pro nevidomé a tak nebyl moc schopen hovor zavěsit a vrátit se do aplikace.

Navrhnul možnost pomoci z druhé strany, kdy by uživatel odeslal pouze SMS s žádostí o pomoc a vyhledaná osoba by se rozhodla, zda mu zavolá, nebo odpoví pomocí SMS zprávy. v souvislosti s tím přišel s nápadem zapracovat do aplikace klávesnici fungující na stejném principu UI.

Subjektivně také podotknul, že by tlačítka mohla být klidně menší a na obrazovku by se jich tak dalo vměstnat více najednou.

Jinak byl s aplikací velmi spokojen, neměl moc dalších výtek k jejímu chování. i bylo vidět, že se mu to vyloženě líbí. Takovou perličkou na závěr k tomuto participantovi bych uvedl, že během testu vyřizoval několik telefonních hovorů, a v průběhu jednoho dokonce ještě psal SMS. Úctyhodný výkon.

## Participant 2

Druhým participantem byl další zkušený uživatel s vystudovanou informatikou na dvou vysokých školách, který měl ale již oproti prvnímu participantovi mírně chladnější přístup k těmto technologiím, Jelikož nepoužívá telefon s dotykovým displejem a ani to neplánuje.

Přesto se v aplikaci rychle zorientoval, opět byly vyzdviženy rychlé reakce syntézy řeči na akce uživatele a vibrační mřížka mezi tlačítky již i v kombinaci s haptickou odezvou na jakýkoliv dotyk uživatele.

Návrh uživatelského rozhraní se mu velice líbil, připomínky měl spíše k samotnému principu aplikace, ale to bylo mimo rámec mého testování. Jedinou větší výtku měl k neexistenci mezer mezi tlačítky, kdy uživatel sice při prvním kliku blízko hrany tlačítka zvolí jednu možnost, při jejím potvrzení se ale nemusí trefit a snadno může kliknout vedle. Nedojde však k zvolení jiné volby, to by to musel znovu potvrdit, každopádně je to však hodnotná připomínka, v sekci doporučení do další fáze projektu se tomu budu věnovat více.

### 5.1.3 Shrnutí testů

Již z průběhu testování bylo vidět, že testování bude úspěšné. Bylo získáno spoustu užitečných informací o interakci nevidomých uživatelů s dotykovým zařízením. Další pozitivum

bylo kladné přijetí návrhu UI oběma testovanými. Oba byli spokojeni a vidí v tom potenciál pro další vývoj.

#### 5.1.4 Doporučení pro další fáze projektu

Jako hlavní věc nutnou k přepracování se ukázal způsob volby akcí. V low-fidelity prototypu to fungovalo tak, že uživatel jezdil prstem po displeji a automaticky se mu předříkávali akce jednotlivých kláves. Poté, co prst zvedl, se poslednímu aktivnímu tlačítku přiřadil pomyslný focus a poté ho již stačilo stisknout a akce se potvrdila a vykonala. Zde je několik možností pro modifikaci. Například by se mohla akce vykonat rovnou po zvednutí prstu z displeje, ale toto by muselo být zkombinováno s nějakou prodlevou, jelikož by poté uživatel byl prakticky nucen vybrat nějakou volbu a nesměl by ani omylem odtrhnout prst od displeje. To je nepřijatelné a mohlo by to působit zmatečně, nicméně není úplně špatná cesta a teoreticky by se tím dalo ovládání zrychlit, jen by se to celé muselo pořádně promyslet a co nejlépe zkombinovat se současným stavem.

Další doporučení je zamyšlení nad velikostí tlačítek. Druhý participant zmiňoval, že mu tam chybí mezera a poté se může snadno nestrefit správně. Výborné řešení by dle mého názoru bylo přiřadit tlačítku, které má pomyslný focus, větší plochu pro stisk, takže pro zvolení vedlejší volby, by uživatel musel kliknout dále. Zde ale není jasné jak to řešit, každopádně by to mohlo velice zpřesnit interakci s aplikací.

Poslední doporučení byla brána spíše jako dobrý nápad. Tím je implementace klávesnice pro Android s UI na základě prototypu aplikace.

## 5.2 Testování highfidelity prototypu

### 5.2.1 Úvod do testování

Testování high-fidelity prototypu bylo rozděleno do dvou základních částí. První částí je testování funkční stránky aplikace, tedy zaznamenávání polohy, rozpoznání kontextu pohybu, dopadu na výdrž baterie a spotřebu dat při přenosu naměřených hodnot na server. Druhou částí je test uživatelského rozhraní přizpůsobeného cílové skupině.

### 5.2.2 Testování funkčních požadavků

#### Popis testů a testovací setup

Cílem této části testování bylo ověření projektu po funkční stránce. Tato část testů byla prováděna jako testování bez uživatelů.

- Rozpoznání kontextu uživatele
- Zaznamenávání polohy
- Odesílání dat a následné hrubé zpracování - využitelnost výstupu
- Dopad běhu aplikace na výdrž zařízení při běhu na baterii
- Množství odeslaných dat

### **Rozpoznání kontextu a záznamu polohy**

Testování rozpoznání kontextu a zaznamenávání polohy probíhalo jako simulace reálného využití. Trasa byla navržena tak, aby bylo možno sledovat všechny možné předpokládané podmínky ve kterých se aplikace bude používat. Byla zvolena pevná trasa obsahující úseky absolvované pěšky následované přesunem vozidly MHD, a to jak tramvají, autobusem, tak zejména metrem. Důraz byl kladen především na rychlost reakcí aplikace při změnách kontextu tak, aby se minimalizovala doba odezvy a tedy i ztráta výstupních dat. Veškerá naměřená data byla odesílána v reálném čase na testovací server, kde byla uložena v jednoduché SQL databázi a poté zpracována a vizualizována pomocí nástroje Google Fusion Tables.

### **Testování energetické náročnosti**

Spotřeba a dopad na výdrž zařízení byla testována pomocí zabudovaných nástrojů systému Android a vývojářských výstupů dostupných přes Android Debug Bridge. Testována byla jak spotřeba v pohotovostním režimu aplikace, tak při zaznamenávání polohy. Měření bylo prováděno i v rámci testů rozpoznání kontextu a zaznamenávání polohy. Výstup je tedy rozdělen na tři samostatné části.

### **Testování množství odeslaných dat**

Množství odeslaných dat bylo testováno pomocí k tomu určených nástrojů integrovaných přímo v rámci OS Android ve verzi 4.0.4. z těch lze získat přesný objem dat, jaký aplikace spotřebovala, je rozdělen i na spotřebu dat při připojení přes síť Wi-Fi i přes datové přenosy mobilních sítí.

### **Testovací setup**

Pro testování byl použit telefon Samsung Nexus S s aktuální verzí softwaru OS Android 4.0.4.

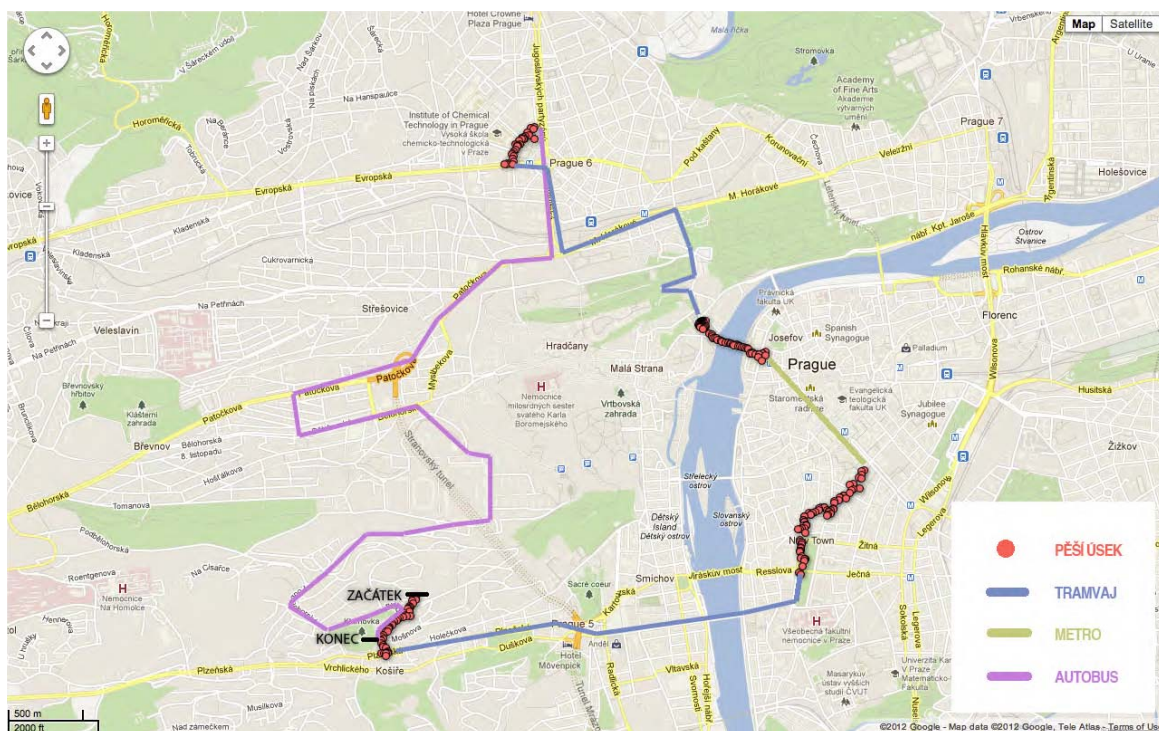
### **Průběh testů**

V této části je zpracován průběh testů funkčních požadavků, tedy testů bez uživatele.

### **Rozpoznání kontextu a záznam polohy**

Test proběhl ve dvou kolech, původně bylo zamýšleno, aby oba testy byly co nejvíce shodné, ale nakonec se lehce lišila trasa, základ byl stejný, ale vzhledem k nevyhovujícím intervalům MHD byla při druhém testu trasa zlehka upravena oproti původnímu plánu. Tato změna nicméně nijak nenarušila samotné testování, většina trasy byla navíc stejná, včetně těch nejzajímavějších úseků. Rozdíl byl pouze skoro na konci testu, takže lze data z obou testů mezi sebou porovnat.

Níže je vidět mapka průběhu testu. Na prvním obrázku jsou vidět data naměřená při prvním testu doplněná o úseky v MHD, tedy ve chvílích, kdy nebylo ukládání polohy aktivní.



Obrázek 5.4: Zobrazení dat z prvního testu na mapě

Z obrázku 5.4 je vidět, že aplikace reaguje na změny kontextu poměrně rychle (z principu je reakční doba kolem 10 sekund) a tak je lokace pomocí GPS zapnuta prakticky pár kroků poté, co se uživatel rozejde. Vše ale proběhne dostatečně rychle na to, aby na mapě nevznikala bílá místa, kdy aplikace měla snímat polohu, ale nečinila tak.

Při druhém testu se toto bohužel přihodilo, je to možné vidět na obrázku 5.5.

Při analýze dat z testu bylo zjištěno, že tato prodleva byla způsobena nezvykle dlouhým hledáním satelitů GPS, ukládání lokace byla zapnuté, ale protože telefon delší dobu neposkytoval patřičná data, tak aplikace musela čekat.

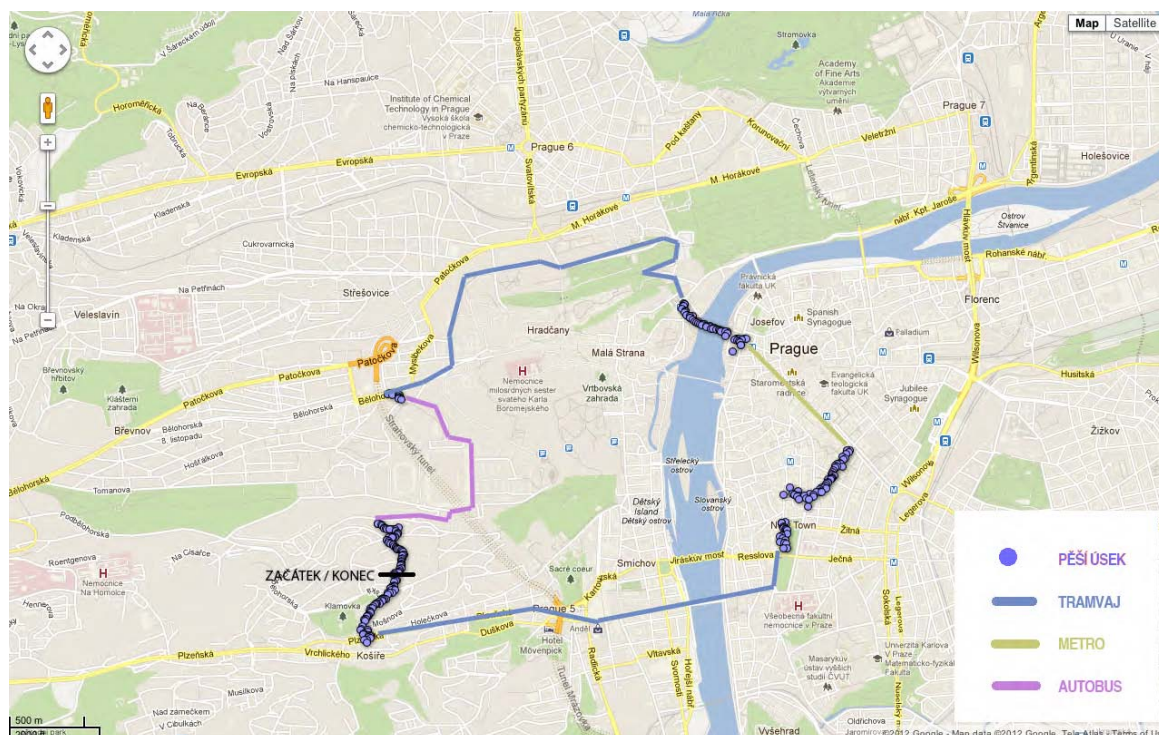
### Testování spotřeby elektrické energie

Při testování spotřeby byly využity nástroje přímo zabudované v systému Android, pomocí kterých je možno přesně zjistit, kolik jaká aplikace spotřebovala energie. Tyto informace telefon zobrazí v podobě procentuálního podílu na aktuálním úbytku stavu baterie, vždy je to vztaženo k poslednímu plnému nabití.

Testování proběhlo taktéž na dvě samostatné části, jelikož aplikace z důvodu úspory energie rozlišuje několik stavů, v jakých běží a jak snímá data.

#### Testování spotřeby energie v pohotovostním režimu

Jelikož je aplikace určena k nepřetržitému běhu na pozadí telefonu, tak bylo třeba co nejvíce optimalizovat její spotřebu v pohotovostním režimu, to je totiž stav, ve kterém se bude telefon velmi pravděpodobně nacházet nejvíce.



Obrázek 5.5: Zobrazení dat z druhého testu na mapě

Detaily o běhu aplikace v pohotovostním režimu byly zmíněny v analýze 3.6.2. Test probíhal tak, že byl telefon s vypnutou Wi-Fi, vypnutými datovými přenosy a bez dalších aplikací běžících na pozadí ponechán v klidu přes noc na polici a na základě porovnání úrovně nabití akumulátoru před a po testu se zjistilo, jakou má aplikace přibližně spotřebu a nakolik se vybíjení zrychlí v porovnání s čistým pohotovostním režimem, bez žádných aplikací běžících na pozadí.

Na obrázku 5.6 je možno vidět úbytek nabití baterie, kdy za přibližně 10 hodin ubylo 11 % z kapacity baterie. Tato hodnota implikuje téměř nereálné čtyři dny výdrže, v praxi to ale znamená, že běh aplikace v pohotovostním režimu nemá na výdrž baterie žádný citelný dopad.

### Testování spotřeby energie při ukládání polohy

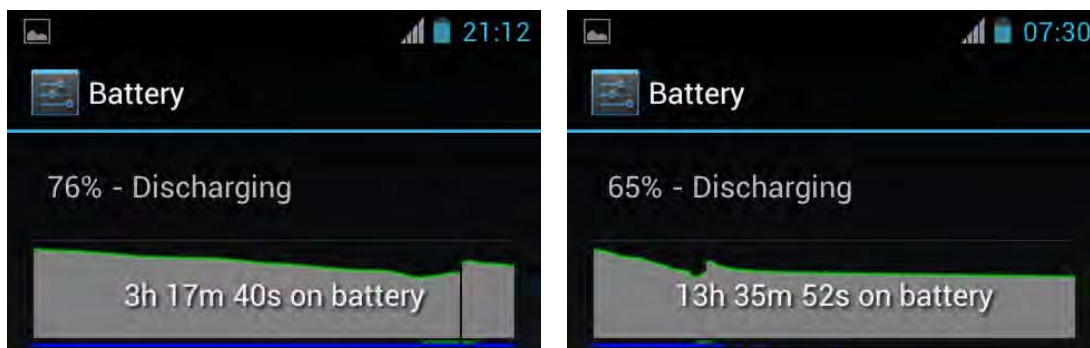
Spotřeba při ukládání polohy byla testována společně s testováním samotného ukládání polohy a rozpoznávání kontextu. Test tedy proběhl ve dvou kolech, kdy byl pokaždé zjišťován procentuální úbytek ze stavu baterie.

Z výsledků testů, které jsou zobrazeny na obrázku 5.7, lze vidět procentuální úbytek z prvního testu. Je vidět že při aktivním sledování polohy spotřebuje telefon kolem 10 % za hodinu. Během testu nebyl navíc po celou dobu aktivní přijímač GPS, při přesunech v MHD se GPS vypnula a aplikace se přehodila zpátky do úspornějšího režimu.

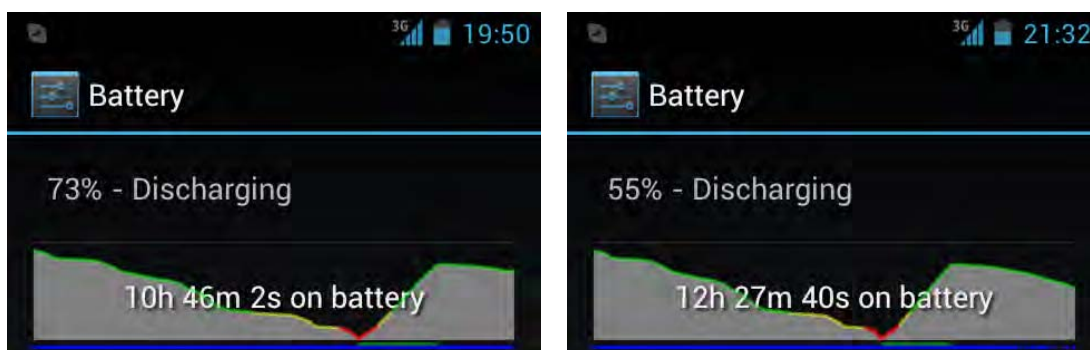
Je tedy zřejmé, že aktivní GPS má skutečně ohromný vliv na výdrž zařízení. z výdrže v řádu jednotek dní udělá výdrž v řádu jednotek hodin, a to je skutečně rozdíl.



Obrázek 5.6: Snímky obrazovky zobrazující průběh vybíjení baterie při běhu aplikace v pohotovostním režimu. Na levém obrázku je stav před testem a vpravo stav po testu.



Obrázek 5.7: Snímky obrazovky zobrazující průběh vybíjení baterie při běhu aplikace režimu ukládání polohy. Na levém obrázku je stav před testem a vpravo stav po testu.



Na druhou stranu byla testovací trasa poměrně dlouhá, obsahující delší úseky v MHD, tak i relativně dlouhé pěší úseky, při kterých byl přijímač GPS po celou dobu aktivní. Po absolvování celé trasy byl úbytek stavu baterie 18 %, tedy by byl za ideálních podmínek telefon na plné nabití schopen absolvovat celou trasu ještě více než pětkrát.

### Testování vlivu odesílání dat na spotřebu elektrické energie

Vliv odesílání dat byl testován ještě v průběhu vývoje aplikace, před samotným testováním. z testů bylo zjištěno, že pokud je současně aktivní GPS a ještě se k tomu posílají data přímo na server v reálném čase, tak se spotřeba zvedne o dalších přibližně 15-20 %. To je relativně vysoká daň za to mít aktuální data, na druhou stranu to oproti samotné GPS není zase tak výrazné. v případě prodloužení intervalů odesílání dat se rozdíl ve spotřebě oproti úplně vypnutým datovým přenosům aplikace prakticky neprojeví.

V rámci testování byl telefon normálně používán, takže data byla aktivní, byla zapnuta synchronizace emailů a dalších služeb. v případě úplného vypnutí datových přenosů se spotřeba při aktivním ukládání polohy nijak nesníží, GPS je prostě natolik majoritní složkou, že

to vliv datových přenosů přebije. Rozdíly jsou zde maximálně v jednotkách procent v rámci několika hodin.

### **Testování objemu přenesených dat**

Objem přenesených dat byl také testován již v průběhu implementace aplikace. Nejdříve bylo na testovací server odesíláno pro finální využití zbytečně velké množství dat, a to byl objem přenesených dat v řádu několik megabajtů za den.

Po odladění měřicí části aplikace byla odesílaná data minimalizována pouze na nezbytně nutné položky pro lokaci a identifikaci uživatele a výsledek byl takový, že se spotřeba začala pohybovat v řádu desítek až stovek kilobajtů za den.

Nebylo využito komprimování http požadavků a všechny identifikátory byly odesílány v nezkrácené formě. Zde je prostor pro další úspory, navíc při větších intervalech mezi odesíláním odpadne spousta redundantních dat pro identifikaci uživatele.

Výsledná spotřeba dat je tedy velice uspokojující, i v rámci poměrně přísných limitů spotřeby dat, jaké dnes poskytovatelé mobilního internetu mají.

### **5.2.3 Testování uživatelského rozhraní**

Cílem testování uživatelského rozhraní bylo otestovat, jestli je ovládání navrženo optimálně v souladu s cílovou skupinou. Tato část probíhala s participantem patřícím do cílové skupiny a úkolem bylo vykonání několika navazujících úkonů, které jsou předpokládány při reálném nasazení projektu. Při testování byl kladen důraz na následující aspekty:

- Úspěšné dokončení všech zadaných úkonů
- Rychlost orientace napříč aplikací
- Sběr zpětné odezvy od participanta pro případné další vylepšení uživatelského rozhraní

#### **Popis zadaných úkolů**

- Přihlášení do aplikace
- Zapnutí automatického ukládání polohy po spuštění
- Vyhledání pomoci a spojení telefonního hovoru s navrženou osobou

#### **Scénář testu**

1. Seznámení s detaily testu a jeho předpokládaným průběhem, krátké představení práce
2. Vyplnění předtestového dotazníku
3. Začátek testu
4. Průběh testu
5. Konec testu

## 6. Vyplnění potestového dotazníku

Úkolem participanta bude dostat se ze zadaného místa po letmo popsané trase do zadaného cílového bodu. v průběhu testu půjde sám bez doprovodu. s sebou si ponese mobilní telefon s nahranou aplikací pro sledování polohy a v případě jakéhokoliv problému bude mít za úkol zkusit pomocí aplikace vyhledat pomoc a jedné z navržených osob zavolat a požádat o pomoc. Test bude probíhat v areálu ČVUT FEL na Karlově náměstí, a to na nádvoří mezi budovami b a E.

Samotný průběh testu bude následující. Participant obdrží telefon se spuštěnou aplikací a jeho prvním úkolem bude se do aplikace přihlásit. Po úspěšném přihlášení začne hlavní část testu, kdy vyrazí po popsané trase směrem k cíli. Pokud během cesty narazí na potíže, tak vyhledá pomoc a spojí se s nějakou z navržených osob. Ta se mu pokusí poradit a participant se na základě těchto informací pokusí zorientovat a úspěšně dorazit do zadaného cíle. v případě, že se vyskytnou další potíže, tak vyhledá pomoc znovu. v případě, že se podaří dorazit do cíle bez toho, aby se vyskytly jakékoliv problémy, participant přesto v cíli zkusí vyhledat pomoc pomocí aplikace a zavolat jedné z navržených osob.

Během testu je využito metody shadowing [2], kdy participanta necháme jít samotného, ale sledujeme ho z bezpečné vzdálenosti, tak aby se dalo předejít případným problémům. Popsaná trasa totiž neodpovídá stoprocentně skutečnosti, tím chceme participanta dostat do stresu a pokusit se namodelovat skutečný případ užití aplikace, když bude uživatel vyhledávat pomoc a skutečně přitom bude cítit, že je ztracen.

## Popis zadané trasy a úkolu

Participant jde navštívit svého kamaráda do menzy v areálu ČVUT na Karlově náměstí a nikdy předtím tam nebyl. Kamarád mu cestu popsal bohužel pouze letmo, a tak participant došel pouze do areálu, ale samotnou menzu už najít nedokáže. Ve vrátnici budovy E se zeptá vrátného na cestu a ten mu poradí, participant se tedy vydá od hlavního vchodu budovy E po popsané trase směrem k menze.

1. Jděte od schodů ke hlavnímu vchodu budovy E směrem doleva (relativně k pohledu směrem z vchodu do budovy ven)
2. Přibližně po 50 metrech zatočte doprava a pokračujte rovně
3. Po dalších přibližně 100 metrech zatočte znovu doprava a podél bloku pokračujte rovně, menza by měla být po pravé straně. (Zde jde o mírně klamavou informaci, ve skutečnosti je po cestě ještě jedna odbočka vpravo, na tu ale participanta schválně neupozorňujeme.)

## Testovací setup

Testování probíhalo na zařízení Samsung Nexus s a OS Android 4.0.4. v průběhu testování bude aktivní nahrávání obrazovky zařízení a ukládání dotyků displeje. Celý test bude také zaznamenáván na video. Předtestový a potestový dotazník bude vytvořen pomocí služby Google Docs.



### Průběh testů

V této části je zpracován průběh testů high-fidelity prototypu s uživateli. Testování proběhlo celkem se čtyřmi participanty. V tabulce 5.3 je vidět jejich přehled.

	Věk	Zkušenosti s mob. tech.	Vzdělání
Participant 1	více než 55 let	průměrné	SŠ s maturitou
Participant 2	více než 55 let	průměrné	SŠ s maturitou
Participant 3	více než 20-35 let	velmi dobré	VŠ
Participant 4	více než 20-35 let	velmi dobré	VOŠ

Tabulka 5.3: Přehled participantů

#### 1. participant

Prvním participantem byla paní, která se již účastnila testů low-fidelity prototypu, nejdříve sice nevěděla, o co jde, ale během krátkého představení cílů práce si vzpomněla.

##### Předtestový dotazník

Shrnutí otázek a odpovědí z předtestového dotazníku je zobrazeno v tabulce 5.4

**Popis průběhu testu s prvním participantem** Od začátku participant vyrazil správným směrem, ale již v první zatáčce se ztratil, takže na základě předtestových pokynů využil aplikaci a skrz volbu vyhledání pomoci úspěšně vytočil jedno z nabízených čísel. Na základě pomoci podané přes telefon úspěšně našel menzu. Po příchodu ke dveřím se chtěl ujistit, že je opravdu na místě a měl v úmyslu znovu použít aplikaci, test byl ale v tuto chvíli ukončen, bylo by již zbytečné znovu telefonovat. Během testu nenastaly žádné potíže, aplikace fungovala tak, jak měla, a po ukončení hovoru se korektně vrátila zpět do hlavní nabídky, takže bylo vše bez potíží připravené pro další použití.

##### Dotazník po testu

Shrnutí otázek a odpovědí z dotazníku po testu je zobrazeno v tabulce 5.5

##### Shrnutí testu

Jelikož se během testů s prvním participantem nevyskytly žádné potíže, tak lze test považovat za maximálně úspěšný. Sám participant neměl s ovládáním aplikace žádné problémy, vše se mu zdálo jednoduché a dalo dle jeho slov by se to dalo rychle naučit. Vyzdvihoval snadnost a jednoduchost ovládání a zejména okamžitou odezvu syntézy řeči na uživatelské akce. Jediná výtku byla směřována k občas příliš složitým popiskům, které aplikace uživateli předcítá.

Otázka	Odpověď
Vlastníte mobilní telefon?	ANO
Používáte telefon s dotykovým displejem?	NE
Máte s využíváním dotykových telefonů (nebo jiných dotykových zařízení) nějakou zkušenost?	ANO - negativní
Důvody předchozí odpovědi	uživatel nemá žádné pevné "záchytné" body
Používáte ve svém telefonu datový tarif?	NE
Míváte někdy potíže s orientací	potíže nemívám (participantka má vodícího psa)
Nevadilo by Vám, kdyby Vás kontaktovali cizí lidé kvůli pomoci?	NE
Věk	více než 55 let
Dosažené vzdělání	SŠ s maturitou

Tabulka 5.4: Participant 1 - předtestový dotazník

## 2. participant

Druhým participantem byl starší pán. Neúčastnil se testování low-fidelity prototypu, ale po krátkém úvodu a vyzkoušení uživatelského rozhraní aplikace se mohlo přejít rovnou k testu, nebyl potřeba žádný delší tutoriál.

### Předtestový dotazník

Shrnutí otázek a odpovědí z předtestového dotazníku je zobrazeno v tabulce [5.6](#)

### Popis průběhu testu s druhým participantem

Participant vyrazil od startu správným směrem, správně zatočil kolem bloku doprava a pokračoval po trase. Před další odbočkou doprava se, jak později zmínil, úmyslně ztratil a místo doprava odbočil doleva. Poté použil aplikaci a vyhledal pomoc. Vyhledání a i spojení telefonního hovoru proběhlo znovu bez potíží a participant se dále vydal již znovu po správné trase směrem k menze. Cestou ale zatočil doprava a místo, aby došel do menzy, došel tedy zpět k místu startu testu. Zde byl test ukončen, aplikace byla otestována a nemělo smysl znovu vyhledávat pomoc a navigovat participanta zpět tak, aby menzu úspěšně našel.

### Dotazník po testu

Shrnutí otázek a odpovědí z potestového dotazníku je zobrazeno v tabulce [5.7](#)

Otázka	Odpověď
Jak se Vám testování líbilo?	Líbilo
Měl(a) jste problém s vykonáním zadaných úkonů, činilo Vám to subjektivně potíže?	Spíše ano
Jaké vám připadalo uživatelské rozhraní testované aplikace?	Dobré, člověk by se to rychle naučil
Dovedete si představit použití takového rozhraní v běžném životě? (na ulici, doma, ..)	ANO - bez potíží
Co se Vám na aplikaci líbilo a co byste naopak změnil(a)?	+ jednoznačné prostředí  + 1x zmáčknou promluví, zmáčknou 2x a akce se vykoná  - zjednodušit pokyny v nastavení
Přišel/přišla byste na testování znovu?	ANO

Tabulka 5.5: Participant 1 - dotazník po testu

### Shrnutí testu

Test sice nebyl ukončen podle předpokládaného scénáře, ale samotné testování aplikace proběhlo bez potíží. Vše fungovalo tak, jak mělo, a participant neměl větších výtek. Zmínil ale, že by bylo dobré, kdyby aplikace po zvolení a potvrzení akce přestala reagovat na dotyky uživatele až do té doby, než se kompletně zobrazí další obrazovka s aktualizovaným obsahem. Jako další věc k vylepšení zmínil zlepšení syntézy hlasu, nepřišel mu moc věrný. Jelikož ale není samotná implementace syntézy řeči součástí této práce, tak se s tím nedá nic dělat. Jedinou možností by bylo vyvinout lepší modul pro převod textu na řeč.

### 3. participant

Třetím participantem byl uživatel, který testuje často a ještě má k tomu velice rozsáhlé znalosti přímo z oboru. Účastnil se i testování low-fidelity prototypu, takže si prakticky ihned vzpomněl, o co jde.

#### Předtestový dotazník

Shrnutí otázek a odpovědí z předtestového dotazníku je zobrazeno v tabulce [5.8](#)

#### Popis průběhu testu s třetím participantem

Třetí participant menzu úspěšně našel, ale musel cestou využít aplikaci, protože se ztratil.

Otázka	Odpověď
Vlastníte mobilní telefon?	ANO
Používáte telefon s dotykovým displejem?	NE
Máte s využíváním dotykových telefonů (nebo jiných dotykových zařízení) nějakou zkušenost?	ANO - spíše negativní
Důvody předchozí odpovědi	problémy s kapacitními HW tlačítky
Používáte ve svém telefonu datový tarif?	pouze doma internet přes Wi-Fi
Míváte někdy potíže s orientací	občas ano
Nevadilo by Vám, kdyby Vás kontaktovali cizí lidé kvůli pomoci?	NE
Věk	více než 55 let
Dosažené vzdělání	SŠ s maturitou

Tabulka 5.6: Participant 2 - předtestový dotazník

Otázka	Odpověď
Jak se Vám testování líbilo?	Líbilo
Měl(a) jste problém s vykonáním zadaných úkonů, činilo Vám to subjektivně potíže?	Rozhodně ne, bylo to lehké
Jaké vám připadalo uživatelské rozhraní testované aplikace?	Dobré, člověk by se to rychle naučil
Dovedete si představit použití takového rozhraní v běžném životě? (na ulici, doma, ..)	ANO - je to na dobré cestě
Co se Vám na aplikaci líbilo a co byste naopak změnil(a)?	- vypnutí snímání - vylepšení hlasu
Přišel/přišla byste na testování znovu?	ANO

Tabulka 5.7: Participant 2 - dotazník po testu

Neodbočil na první odbočce doprava, ale místo toho zašel do výklenku mezi budovami. To ho zmátlo, jelikož se odtamtud dalo jít pouze zpět, tak použil aplikaci a vyhledal pomoc.

Otázka	Odpověď
Vlastníte mobilní telefon?	ANO
Používáte telefon s dotykovým displejem?	NE
Máte s využíváním dotykových telefonů (nebo jiných dotykových zařízení) nějakou zkušenost?	ANO - spíše negativní
Důvody předchozí odpovědi	problémy s kapacitními hardwarovými tlačítky
Používáte ve svém telefonu datový tarif?	ANO - pouze tethering
Míváte někdy potíže s orientací?	NE
Nevadilo by Vám, kdyby Vás kontaktovali cizí lidé kvůli pomoci?	NEVÍM - otázka ochrany soukromí
Věk	20-35 let
Dosažené vzdělání	VŠ

Tabulka 5.8: Participant 3 - předtestový dotazník

Během hovoru nenastaly žádné komplikace, participanta se podařilo vrátit na správnou trasu. Poté, co se participant dostal ven z výklenku, se už dostal k menze bez větších obtíží.

#### Dotazník po testu

Shrnutí otázek a odpovědí z potestového dotazníku je zobrazeno v tabulce [5.9](#)

#### Shrnutí testu

Test proběhl bez potíží, vše fungovalo správně, jak mělo. Na základě diskuze po testu s tímto participantem vznikla celá řada připomínek a návrhů na další fáze projektu, tento test byl zřejmě nejhodnotnější ze všech. Participant aplikaci nevytýkal žádné konkrétní věci, pouze se mu nezdá princip fungování celého projektu. Toto se právě stalo námětem k diskuzi, jejíž závěry jsou shrnuty v sekci doporučení do dalších fází projektu [5.3.1](#).

### 4. participant

Čtvrtým participant byl velice zkušený uživatel mobilní techniky, který často testuje a sám vlastní několik telefonů, ať už s klasickou hardwarovou klávesnicí, nebo s ovládáním pomocí dotykového displeje. Zároveň šlo o velmi pokročilého uživatele, který si v rychlosti ovládání telefonů v ničem nezádá se zdravými lidmi.

#### Předtestový dotazník

Shrnutí otázek a odpovědí z předtestového dotazníku je zobrazeno v tabulce [5.10](#)

Otázka	Odpověď
Jak se Vám testování líbilo?	Líbilo
Měl(a) jste problém s vykonáním zadaných úkonů, činilo Vám to subjektivně potíže?	Rozhodně ne, bylo to lehké
Jaké vám připadalo uživatelské rozhraní testované aplikace?	Snadné
Dovedete si představit použití takového rozhraní v běžném životě? (na ulici, doma, ..)	Bez potíží
Co se Vám na aplikaci líbilo a co byste naopak změnil(a)?	- princip - zapojení neslepců do navigace - pomoc, než se participant ztratí
Přišel/přišla byste na testování znovu?	ANO

Tabulka 5.9: Participant 3 - dotazník po testu

Otázka	Odpověď
Vlastníte mobilní telefon?	ANO
Používáte telefon s dotykovým displejem?	ANO
Máte s využíváním dotykových telefonů (nebo jiných dotykových zařízení) nějakou zkušenost?	ANO - spíše pozitivní
Důvody předchozí odpovědi	líbí se mi to
Používáte ve svém telefonu datový tarif?	ANO
Míváte někdy potíže s orientací?	NE
Nevadilo by Vám kdyby Vás kontaktovali cizí lidé kvůli pomoci?	NEVÍM - otázka ochrany soukromí
Věk	20-35 let
Dosažené vzdělání	VOŠ - absolutorium

Tabulka 5.10: Participant 4 - předtestový dotazník

### Popis průběhu testu se čtvrtým participantem

Čtvrtý participant cestu k menze zvládl naprosto bez pomoci, menzu bez zaváhání našel. To nás překvapilo, tak po příchodu k menze alespoň zkusil zavolat kamarádovi, aby pro něj přišel před menzu, tak, jak to bylo v popisu průběhu testu. Po testu participant uvedl, že vlastně našel menzu omylem. Dostal totiž zavádějící informace tak, aby se cestou ztratil, minul ale odbočku, kam měl zatočit a ztratit se, takže nakonec došel do menzy na první pokus a zcela sám, bez vyhledání pomoci skrz aplikaci.

### Dotazník po testu

Shrnutí otázek a odpovědí z dotazníku po testu je zobrazeno v tabulce 5.11

Otázka	Odpověď
Jak se Vám testování líbilo?	Líbilo
Měl(a) jste problém s vykonáním zadaných úkonů, činilo Vám to subjektivně potíže?	Rozhodně ne, bylo to lehké
Jaké vám připadalo uživatelské rozhraní testované aplikace?	Příliš snadné, klidně přidat funkce
Dovedete si představit použití takového rozhraní v běžném životě? (na ulici, doma, ..)	ANO
Co se Vám na aplikaci líbilo a co byste naopak změnil(a)?	<ul style="list-style-type: none"> <li>- přehození tlačítka zpět na konec nabídky</li> <li>- chybí nastavení</li> <li>- chybí sdělení aktuální polohy</li> <li>- potvrzení volání v případě překlepu</li> <li>+ pokládání tlačítkem</li> <li>- mělo by to využívat systémový screenreader</li> </ul>
Přišel/přišla byste na testování znovu?	ANO

Tabulka 5.11: Participant 4 - dotazník po testu

### Shrnutí testu

Test proběhl bez potíží, možná našel participant menzu až příliš snadno. Nicméně využití se vyzkoušelo, sice až v rámci záložního úkolu, ale aplikace fungovala a to je hlavní. Tento

participant nebyl sice úplně reprezentativním zástupcem cílové skupiny díky svému ohromnému zapálení do mobilní techniky. Umí ovládat telefonu prakticky lépe než spousta lidí bez zrakového postižení a má výborný přehled v oboru. Na druhou stranu byl ale přínosným participantem, jelikož se již účastnil velikého počtu testů všech možných asistivních technologií a aplikací pro zrakově postižené. Jedinou výtkou byla přílišná jednoduchost rozhraní a málo funkcí aplikace. To je ale ukázková výtka, která se vztahuje k jeho zápalu do mobilní techniky.

### Shrnutí testů s uživateli

Uživatelské testy rozhraní aplikace dopadly velice dobře, nikdo neměl s ovládáním větší problémy, vždy se povedlo rychle vyhledat pomoc. Prvotní orientace byla také rychlá, i když zde byly citelné rozdíly mezi jednotlivými participanty, dva z nich byli opravdoví odborníci a ti to měli tedy svým způsobem snazší díky velikým zkušenostem.

Obrázek 5.8: Fotografie zachycující průběh testu s uživatelem





## 5.3 Diskuze výstupu testů

### 5.3.1 Doporučení a návrhy na úpravy do dalších fází projektu

#### Uživatelské rozhraní

##### Zmenšení tlačítek

V testované verzi aplikace vypadalo uživatelské rozhraní tak, že tlačítka zabírala celý displej a nebyly mezi nimi takřka žádné rozestupy, ale pouze několikapixelové ohraničení. Maximalizace tisknutelné plochy se může zdát výhodná, ale bohužel je zde problém, pokud uživatel klikne na tlačítko moc blízko okraje. Jelikož při prvním stisknutí pouze hlasový syntetizátor přečte, jakou akci tlačítko při potvrzení vykoná, je nutné tlačítko stisknout dvakrát po sobě, aby se zvolená akce skutečně provedla. a zde je problém. Při kliknutí blízko okraje se stávalo, že druhý stisk, který měl zvolenou akci potvrdit, byl již zaznamenán sousedním tlačítkem, takže se místo vykonání akce přečetla akce jiného tlačítka. To je lehce matoucí, jelikož uživatel toto nepředpokládá, on chce vybrat a potvrdit akci. Řešení jsou nasnadě dvě.

První možností je zmenšení tlačítek tak, aby mezi nimi byly dostatečné mezery a nedocházelo tak k nechtěným překlikům. Toto řešení má však nevýhodu ve zmenšení akční plochy, takže uživatel musí přesněji cílit na jednotlivá tlačítka, což činí ovládání méně pohodlným.

Druhou možností je adaptivní zmenšování tlačítek na základě aktuálně vybrané akce. v případě, že by uživatel vybral nějakou volbu, tak tlačítko sloužící pro její potvrzení by se dočasně zvětšilo oproti sousedním tlačítkům, která v sobě skrývají nějakou jinou akci. Nechtěné překliky přes kraj tlačítka by tak byly vyhodnoceny správně, ale nevýhodou tohoto řešení je právě ta flexibilita. Musela by se testováním najít optimální hranice, kdy je korekce dotyků ještě únosná a kdy už to bude pro uživatele naopak matoucí. Nesmělo by se totiž stávat, aby se potvrdila akce i v případě, že chce uživatel skutečně volbu změnit.

Poslední možností je nechat vše tak, jak to je, a spolehnout se na to, že si uživatelé na chybějící mezery zvyknou a naučí se aplikaci ovládat takovým způsobem, aby k omylům nedocházelo.

##### Změna vibrační mřížky mezi tlačítky

Tlačítka jsou oddělena pomyslnou mřížkou, kterou když uživatel přejede prstem, tak telefon zavibruje. v aktuální verzi zařízení vibruje pouze v případě změny tlačítka, takže pokud uživatel nechá prst na hraně, tak zařízení po nastaveném čase vibrovat přestane. Vhodnější by bylo, aby zařízení vibrovalo po celou dobu, po kterou se uživatel bude dotýkat hran mezi tlačítky.

##### Dočasné pozastavení reakcí na dotyky displeje po potvrzení volby

V případě, že uživatel potvrdí nějakou akci, tak aplikace toto nejdříve oznámí pomocí syntézy řeči a až poté začne zvolenou akci vykonávat. Pokud se však uživatel během čtení akce tlačítka omylem dotkne displeje, tak se akce přeruší a vybere se jiné tlačítko. Toto chování je matoucí, zrakově postižení uživatelé se občas displeje omylem dotknou a ani to

nevědí, takže potom nastane situace, kdy místo vykonání údajně zvolené akce bude vybrána jiná akce a přečte se její popis.

Řešením je dočasně deaktivovat reakce na dotyky tlačítek od okamžiku potvrzení akce, až do okamžiku přechodu do jiné aktivity.

## **Ukládání polohy, úspora energie a odesílání dat**

### **Intervaly měření**

Základní doporučení je vytvořit algoritmus který bude rozhodovat, jestli má aplikace běžet v úsporném, nebo pohotovostním režimu. Mezi těmito režimy je totiž veliký rozdíl ve spotřebě a cílem tedy je, aby aplikace co nejvíc běžela v úsporném režimu a do pohotovostního režimu, kdy detekuje změny kontextu s větší frekvencí, ale zároveň má aplikace výrazně vyšší spotřebu energie, přecházela opravdu jenom ve chvílích, kdy je to nutné. Ideální by bylo, aby aplikace ze začátku kontrolovala změny kontextu rovnoměrně a časem upravovala intervaly tak, aby v časech, kdy je pravděpodobné, že uživatel někam půjde, kontrolovala častěji, a naopak šetřila baterii například v noci, kdy uživatel pravděpodobně bude spát.

Výsledkem by byla výrazně snížená průměrná spotřeba aplikace během delšího časového úseku a zároveň i větší přesnost měření. Povedlo by se tím dosáhnout rovnováhy mezi těmito dvěma základními položkami. z počátku by se sice aplikace musela adaptovat na konkrétního uživatele, ale časem by se naučila, jak se uživatel chová, a šetřila by energii víc a víc.

Podmínkou nutnou pro funkčnost tohoto řešení a zároveň také největším problémem je to, že se uživatel musí chovat deterministicky, musí být v jeho chování nalezen vzor, podle kterého se aplikace zkalibruje. v případě, že bude uživatel chodit ven pokaždé úplně jindy, tak to fungovat nebude a ke snížení spotřeby nedojde. Respektive nedojde buďto ke snížení spotřeby, nebo bude aplikace reagovat pozdě na změny kontextu a bude docházet ke ztrátě cenných dat.

### **Prodleva při vypínání GPS**

Dalším krokem pro vylepšení funkčnosti by bylo přidání prodlevy před vypnutím GPS tak, aby se přijímač zbytečně nevypínal při krátkých přerušeních chůze. To vede ke zbytečné zvýšené spotřebě energie a také k tomu, že se musí častěji vyhledávat satelity, což může znamenat ztrátu informací o poloze dokud se satelity znovu nechytí, což může v městské zástavbě trvat klidně i minutu.

Například pokud uživatel na kraji silnice zastaví a bude čekat, než mu padne zelená, tak v případě, že bude čekat více než přibližně 15 sekund, aplikace situaci vyhodnotí jako změnu kontextu a vypne měření pomocí GPS. Když se potom uživatel rozejde, tak aplikaci nějakou dobu trvá, než se vše znovu změří, a ona vyhodnotí kontext jako chůzi a GPS zapne. Pokud toto celé bude trvat déle než minutu, je pravděpodobné, že se budou muset znovu vyhledávat satelity, a tím se zvýší spotřeba a zároveň budou z tohoto úseku chybět data o poloze uživatele.

Ideální by bylo, aby se v tomto případě ponechala GPS po nějakou dobu aktivní, ale zároveň aby se přestala poloha ukládat, případně rovnou odesílat na server, z důvodu zamezení vzniku nadbytečných záznamů, které zvyšují datovou a paměťovou náročnost aplikace.

### Úspora přenesených dat

Další úspory přenesených dat je možné docílit použitím komprimace všech HTTP požadavků, toto by asi pomohlo nejvíce. Další možností je odstranit veškeré možné redundatní údaje, například pokud by se poloha v čase nezměnila o danou konstantu, tak by se nová poloha neodeslala a aplikace by to automaticky vyhodnotila tak, že uživatel je na stejném místě.

Existuje také možnost posílání dat přes mobilní sítě úplně vypnout, aplikace by pak sledovala, kdy se zařízení připojí na Wi-Fi a po připojení by všechna nasbíraná data odeslala. Datová náročnost by tak z hlediska úspory FUP klesla na nulu.

### Vylepšení možností komunikace a vyhledání pomoci

Současný princip vyhledávání pomoci má několik problémů, zejména to, že ve chvíli, kdy se uživatel ztratí, tak neví, kde je, to znamená, že až vytočí číslo někomu, kdo by mu měl pomoci, tak mu jenom těžko může popsat, kde se nachází, a druhá strana mu tak nemůže snadno poradit. Řešením by bylo předat pomáhající straně polohu ztraceného uživatele na základě dat z GPS, ale zde je problém, že v současné době aplikace uvažuje nevidomého uživatele i na druhé straně, takže odpadá možnost zobrazit polohu na mapě. Zeměpisné souřadnice jsou sice přesné, bohužel však jsou bez jejich zobrazení na mapě uživateli k ničemu a například přibližná poloha pomocí určení ulice a popisného čísla nestačí. Není to jednoduše dostatečně přesné a hlavně uživatel na druhé straně pravděpodobně vůbec neví, jaké čísla popisné mají domy, kolem kterých chodí, a tak by mu tato informace nebyla k ničemu.

Tento problém je řešitelný pomocí změny principu fungování aplikace. Například by na druhé straně mohl být operátor servisního centra, který by viděl přesnou polohu na mapě a mohl by tak přesně a rychle ztracenému poradit a navést ho správným směrem.

Další možností by bylo volné rozšíření aplikace tak, aby na straně pomáhajících byli i uživatelé bez zrakového postižení. Například dobrovolníci, kteří by se zapojili do projektu. Byli by srozuměni s tím, že je někdo může kontaktovat a požádat o pomoc, a když by se to stalo, tak by se prostě zobrazila poloha ztraceného uživatele na mapě, a dobrovolník by tak hned věděl, kde se dotyčný nachází a mohl mu poradit. Aplikace by sbírala i jeho data o poloze, případně by byla možná varianta s výběrem lokality, kde se dobrovolník často nachází a zná to tam a je tedy schopný lidem nacházejícím se v této lokalitě pomoci. To v případě strachu o soukromí.

Další možností by bylo kontaktování dobrovolníků i podle jejich aktuální polohy. Ti by poté mohli v případě, že by se nacházeli nablízku, přímo osobně pomoci. Zde je však otázka, nakolik by to bylo funkční, když se ztracený může někoho zeptat osobně, a jak dlouho by třeba taková osobní pomoc trvala. Také je otázkou, kdo by vůbec byl ochoten přerušit nějakou svou práci a jít ztracenému pomoci.

## **Doporučená dodatečná funkcionalita**

### **Zjištění aktuální polohy**

Bylo by dobré, kdyby aplikace uměla uživateli v rychlosti pomocí syntézy řeči sdělit jeho aktuální polohu v podobě ulice a čísla popisného. Toto určení polohy sice není úplně přesné, mohlo by to však někdy stačit pro rychlé zorientování a nebylo by tak třeba vůbec do pomoci zapojovat nikoho dalšího. Uživatel by měl tuto funkci jako další položku v hlavním menu.

S tímto souvisí například poskytnutí informací o nejbližších možných bodech zájmu a podobně. Toto je ovšem již dost daleko od původního záměru aplikace, šlo by spíše o jistou formu navigačního softwaru pro nevidomé.

### **Vyhledání pomoci předtím, než se uživatel ztratí**

Šikovné by bylo, kdyby aplikace uměla vyhledat pomoc dopředu, tedy ještě než se uživatel ztratí. Ten by poté ještě předtím, než by se případně dostal do nějaké stresové situace, mohl zjistit, jestli se na místě vyskytuje nějaký problém, a když by o něm věděl dopředu, tak by to pro něj byla jistě výhoda.

# Kapitola 6

## Závěr

### 6.1 Zhodnocení splnění cílů projektu

V rámci této práce se povedlo vyřešit problém dlouhodobého sledování polohy uživatelů, ukládání informací o jejich pohybu a základní interpretace těchto dat. Většina funkcionality byla i přímo naimplementována a otestována a na základě těchto testů bylo navrženo nespočet vylepšení do dalších fází projektu. V průběhu práce na projektu, ať už se jednalo o analýzu, nebo o implementaci či testování vyvstalo spoustu problémů, které nebyly zpočátku vidět a nebylo s nimi počítáno. Výsledkem této práce není sice hotový plně funkční projekt schopný například komerčního nasazení, ale spíše zpracování problematiky jako celku. Práce poskytla zajímavá data využitelná při tvorbě uživatelských rozhraní pro zrakově postižené. Poskytla také ale hodnotné informace týkající se optimalizace aplikací využívající sensory integrované do zařízení z hlediska spotřeby elektroické energie.

Úspěšný byl i návrh a testování uživatelského rozhraní, které se ukázalo jako plně funkční a to nejen pro potřeby této aplikace, ta si totiž díky důrazu na jednoduchost vystačí s maximálně jednoduchým rozhraním, ale také například pro tvorbu nějakých jiných, z hlediska ovládání složitějších aplikací.

### 6.2 Doporučení do dalších fází projektu

Výstup této práce se dá využít jako podklad pro návrh výsledného aplikace, která by byla schopná nasazení v ostrém provozu. Nemusí to však nutně být řešení stejného principu jako je tato aplikace. Jelikož je práce součástí projektu NaviTerier [4], tak se výstup této práce může použít v nějaké jiné formě, co však pravděpodobně zůstane, je cílení na zrakově postižené uživatele. Dokončené části projektu jsou nicméně plně funkční a dále rozšiřitelné, takže je možné aplikaci dále rozvíjet, tedy implementovat doporučení a návrhy, které vzešly z testů 5.3.



# Literatura

- [1] Douglas Crockford. Introducing json, 1999.  
<<http://paulbourke.net/miscellaneous/dft/>>, stav z 21.5.2012.
- [2] Pawan Vora. Shadowing method.  
<<http://www.usabilityhome.com/Shadow.htm>>, stav z 21.5.2012.
- [3] Code Factory. Introducing mobile speak, 2012.  
<<http://www.codefactory.es/en/products.asp?id=316>>, stav z 21.5.2012.
- [4] CVUT. Naviterier - navigation system in buildings for the visually impaired, 2008.  
<<http://ulab.cz/naviterier/>>, stav z 21.5.2012.
- [5] Galop. Speciální software pro mobilní telefony - ceník, 2012.  
<<http://www.galop.cz/cenik.php?sk=24>>, stav z 21.5.2012.
- [6] Godsk, Kjærgaard, Toftkjær, and Torben. Entracked: Energy-efficient robust position tracking for mobile devices. 2009.
- [7] Google. Svox - modul pro Češtinu, 2012.  
<[https://play.google.com/store/apps/details?id=com.svox.classic.langpack.ces\\_cze\\_fem](https://play.google.com/store/apps/details?id=com.svox.classic.langpack.ces_cze_fem)>, stav z 12.3.2012.
- [8] Google.inc. Odkaz na android software development kit, 2012.  
<<http://developer.android.com/sdk/index.html>>, stav z 9.2.2012.
- [9] M. Khan. Nyquist shannon theorem - signal to noise ratio - calculation & formula, 2012.  
<<http://www.enggpedia.com/computer-engineering-encyclopedia/dictionary/digital-logic-design/1623-nyquist-shannon-theorem-of-signal-to-noise-ratio-calculation-a-formula>>, stav z 9.2.2012.
- [10] M. Orlich. Základní lokalizační metody v gsm, 2012.  
<<http://access.feld.cvut.cz/view.php?cislocclanku=2006022801>>, stav z 25.4.2012.
- [11] Paul Bourke. Fast fourier transform, 1993.  
<<http://paulbourke.net/miscellaneous/dft/>>, stav z 21.5.2012.

- [12] Petr Podhorský. Kontextová inteligentní navigace - klientská aplikace, *Diplomová práce*, 2009.  
<[https://dip.felk.cvut.cz/browse/pdfcache/podhop2\\_2009dipl.pdf](https://dip.felk.cvut.cz/browse/pdfcache/podhop2_2009dipl.pdf)>, stav z 21.5.2012.
- [13] Piotr Wendykier. Výkonnostní testy knihovny JTransform.  
<<https://sites.google.com/site/piotrwendykier/software/jtransforms>>, stav z 10.1.2012.
- [14] Reto Meier. Reducing the battery impact of apps that downloads content over a smartphone radio, 2012.  
<<http://stackoverflow.com/questions/9950379/reducing-the-battery-impact-of-apps-that-downloads-content-over-a-smartphone-rad>>, stav z 21.5.2012.
- [15] SONS. Kdo je zrakově postižený?, 2012.  
<<http://www.sons.cz/kdojezp.php>>, stav z 21.5.2012.
- [16] P. Wendykier. Jtransforms is the first, open source, multithreaded fft library written in pure java.  
<<https://sites.google.com/site/piotrwendykier/software/jtransforms>>, stav z 9.2.2012.
- [17] WHO. Icd-10 version:2010, 2012.  
<<http://apps.who.int/classifications/icd10/browse/2010/en#/H53-H54>>, stav z 16.5.2012.



## Příloha A

# Seznam použitých zkratek

**GPS** Global positioning system

**GSM** Global System for Mobile Communications

**HTTP** Hypertext transfer protocol

**JSON** Javascript Object Notation

**MHD** Městská hromadná doprava

**OS** Operační systém

**SDK** Software development kit

**SQL** Structured query language

**UI** User interface



## Příloha B

### Obsah přiloženého CD

```
.
├── build
│   └── BlindAssistant.apk
├── images
│   ├── test1.png
│   └── test2.png
├── readme.txt
├── src
│   ├── BlindAssistant
│   │   └── text
│   ├── bp.pdf
│   └── latex
│       └── bp.tex
```