

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Aplikace pro mobilní záznam testů uživatelské činnosti

Michal Žák

Vedoucí práce: Ing. Ivo Malý

Studijní program: Softwarové technologie a management, Bakalářská

Obor: Web a multimedia

26. května 2011

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Ivo Malému za dobře míněné rady k práci a trpělivost, kterou se mnou měl .

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 dne 26. 5. 2011

.....

Abstract

The bachelor thesis is aimed to testing of usability during testing users in mobile environment. The goal is to design a cell phone application which would be able to create observations during the user testing. Focus is placed on speed of receiving data while testing. The operating system Google Android was chosen as a platform. The final result is a working application for creating notes while user testing of indoor navigation SW.

Abstrakt

Tato bakalářská práce se zabývá testem použitelnosti při testování uživatelů v mobilním prostředí. Cílem je navrhnout mobilní aplikaci pro rychlý zápis poznámek při uživatelském testování. Důraz je kladen na rychlost vstupu dat při testu. Jako platforma byl vybrán operační systém Google Android. Výsledkem je funkční aplikace pro zápis poznámek při uživatelském testování navigačního SW v interiéru.

Obsah

1	Úvod	1
2	Analýza a návrh řešení	3
2.1	Analýza	3
2.1.1	Test použitelnosti v laboratoři	3
2.1.2	Test použitelnosti v mobilním prostředí	4
2.2	Volba implementačního prostředí	4
2.2.1	Symbian	4
2.2.2	Android	4
2.2.3	RIM	4
2.2.4	iOS	4
2.2.5	Windows Mobile	4
2.2.6	Výběr	5
2.3	Popis řešeného problému	5
2.4	Požadavky na implementovaný systém	5
2.5	Vymezení cílů	7
2.5.1	Přípravná fáze	7
2.6	Návrh řešení	7
2.6.1	Příprava testu	7
2.6.2	Grafické uživatelské rozhraní	8
2.6.2.1	Úvodní obrazovka	8
2.6.2.2	Obrazovka s nastavením testu	8
2.6.2.3	Obrazovka s mapou testu	9
2.6.2.4	Seznam vyfiltrovaných událostí	10
2.6.2.5	Obrazovka s novou událostí	10
2.6.2.6	Obrazovka dlouhotrvajících akcí	11
2.6.2.7	Obrazovka s právě probíhajícími událostmi	12
2.6.2.8	Okno na úpravu textu	12
2.6.3	Datové struktury	12
2.6.4	Ovládání	13
3	Realizace	15
3.1	mutt.logic	15
3.1.1	XML parsování	15
3.1.2	Datový model	15

3.1.3	Datový přenos	15
3.2	mutt.GUI	16
3.2.1	Hlavní obrazovka	16
3.2.2	Seznam vyfiltrovaných událostí	16
3.2.3	Obrazovka s mapou testu	16
3.2.4	Obrazovka dlouhotrvajících akcí	17
3.2.5	Obrazovka s novou událostí	17
3.2.6	Okno na úpravu textu	18
3.2.7	Obrazovka s právě probíhajícími událostmi	18
3.2.8	Obrazovka s nastavením testu	19
4	Testování	21
4.1	Zkouška funkčnosti	21
4.2	Příprava testu	21
4.3	Průběh testu	21
4.4	Vyhodnocení zkoušky funkčnosti	22
5	Závěr	23
5.1	Pokračování práce	23
A	Seznam použitých zkratk	27
B	Instalační a uživatelská příručka	29
B.1	Příprava telefonu	29
B.2	Instalace	29
B.3	Příprava nastavení testu	29
B.4	Průběh testu	29
B.4.1	Přidání jednorázové akce	30
B.4.2	Přidat déle trvající akci	30
B.4.3	Ukončit déle trvající akci	30
B.4.4	Ukončit test, zapsat poznámky do logu	30
C	Xml soubory	31
C.1	Soubor nastavení testu pro zkoušku funkčnosti aplikace	31

Seznam obrázků

2.1	Laboratoř pro testování	3
2.2	Náhled mapy s předpokládanými místy vzniku události	6
2.3	Úvodní obrazovka	8
2.4	Obrazovka s nastavením testu	9
2.5	Obrazovka s mapou testu	9
2.6	Seznam vyfiltrovaných událostí	10
2.7	Obrazovka s novou událostí	11
2.8	Obrazovka dlouhotrvajících akcí	11
2.9	Obrazovka s právě probíhajícími událostmi	12
2.10	Okno na úpravu textu	13
3.1	Úvodní obrazovka	16
3.2	Seznam vyfiltrovaných událostí	17
3.3	Obrazovka s mapou testu	17
3.4	Obrazovka dlouhotrvajících akcí	18
3.5	Obrazovka s novou událostí	18
3.6	Okno na úpravu textu	19
3.7	Obrazovka s právě probíhajícími událostmi	19
3.8	Obrazovka s nastavením testu	19

Seznam tabulek

2.1	Celosvětový podíl smartphonů podle mobilních OS	5
2.2	Předdefinované objekty	6

Kapitola 1

Úvod

Při uživatelském testování je důležité si zaznamenávat průběh testu. Tento záznam poté slouží k vyhodnocení testování a pro případné úpravy vyvíjeného SW. Při testech v laboratoři se využívá videozáznamu, ke kterému jsou zapisovány poznámky. S rozvojem mobilních zařízení je potřeba také testovat aplikace, které vyžadují pohyb v terénu jako jsou navigační SW. Při testování navigací je potřeba zaznamenávat poznámky spojené s interakcí uživatele s okolním prostředím. Není problém s sebou na testování do terénu vzít notebook a zapisovat poznámky na něm. Toto řešení je ale dost nepohodlné. Nabízí se použít menší mobilní zařízení, jako jsou mobilní telefony nebo tablety.

Na trhu existují nástroje pro mobilní zařízení, které umožňují popis chování uživatele. Tyto nástroje se zaměřují na popis interakce uživatele s testovaným SW. Již ale neexistují nástroje, které umožňují popisovat interakci uživatele s okolním prostředím. V této práci se pokusím takovýto nástroj navrhnout a implementovat.

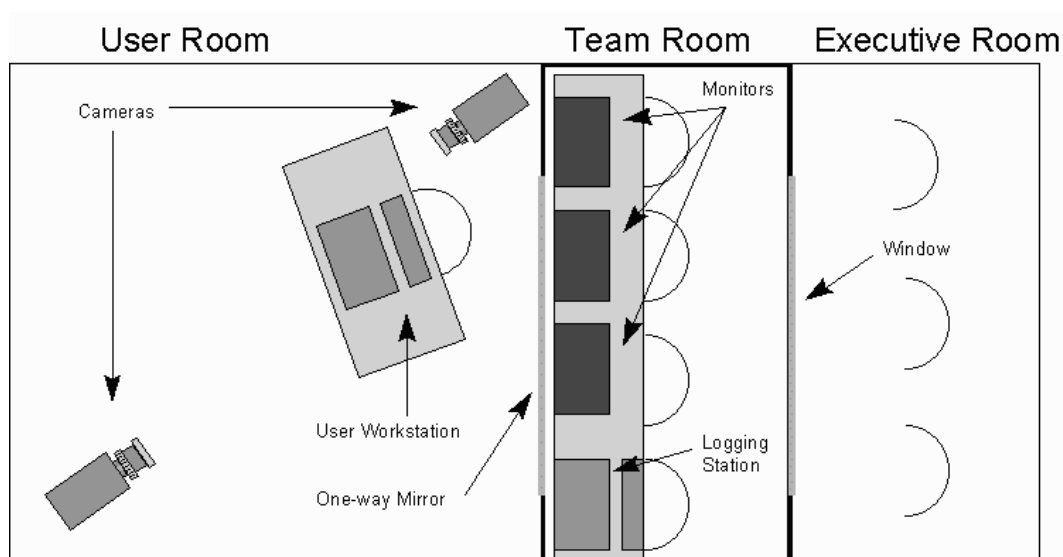
Kapitola 2

Analýza a návrh řešení

2.1 Analýza

2.1.1 Test použitelnosti v laboratoři

Při testování použitelnosti v laboratoři je připraven počítač, na kterém je testované uživatelské prostředí. V místnosti by mělo být také jednosměrné zrcadlo k účelu pozorování uživatelského chování z druhé místnosti. Na obrázku 2.1 viz [8] je vidět laboratoř pro testování.



Obrázek 2.1: Laboratoř pro testování

Osoba dostane několik úkolů, které má provést. Při tomto testování se provádí videozáznam celého testu. Ten je pak dále zpracováván. Existuje řada nástrojů, které umožňují k místům, kde se vyskytl nějaký problém, např. uživatel nevěděl co má jak udělat, přiřadit poznámky. Cílem tohoto testování, je zaznamenávat interakci uživatele s testovaným SW. Při zaznamenávání interakce u tohoto testování je výhoda použití klávesnice.

2.1.2 Test použitelnosti v mobilním prostředí

Při tomto testování dostane uživatel zařízení s testovaným SW. Poté má za úkol, stejně jako v laboratoři, udělat nějaké kroky podle předem připraveného scénáře. Je zapotřebí zaznamenat nejen interakci uživatele s uživatelským prostředím, ale i interakci s okolním prostředím. V laboratoři není problém se zápisem poznámek. Během tohoto testování máme ale k dispozici pouze zařízení s dotykovým displejem, na kterém není psaní nikterak rychlé. Další nevýhodou oproti testování v laboratoři je nutnost dělat vlastní pozorování. V laboratoři toto není nutné z důvodu video záznamu.

V této práci se zabývám konkrétně testováním použitelnosti navigační aplikace pro nevidomé. Při testování mohou nastat situace např. že uživatel zakopne o nějakou překážku. Tyto situace je potřeba zaznamenat.

2.2 Volba implementačního prostředí

Na výběr je spousta mobilních OS. Mezi největší patří Symbian, Google Android, RIM, iOS, Microsoft.

2.2.1 Symbian

Symbian je open source OS, který je převážně v mobilních zařízeních značky Nokia. Jako programovací jazyk je použit C++. Tento mobilní OS je v roce 2010 celosvětově nejpoužívanější.

2.2.2 Android

Mobilní OS Google Android je stejně jako Symbian open source. Vyskytuje se v daleko větším množství značek mobilních zařízení. Poslední dobou velmi dotahuje popularnost Symbianu. OS Android je postaven na technologii Java.

2.2.3 RIM

Research In Motion je podle serveru reselleronline.cz (viz. [1]) na třetím místě v prodeji v roce 2010. Používá se v telefonech BlackBerry. Je postaven na J2ME.

2.2.4 iOS

Tento OS vyvíjí firma Apple a dodává je do svých zařízení iPhone, iPad a iPod. Programovacím jazykem pro tuto platformu je Objective C.

2.2.5 Windows Mobile

Posledním zástupcem je Windows Mobile od firmy Microsoft. Tento OS není dostupný jako open source. Nemá takovou oblíbenost, jako již zmíněné OS. Pro psaní aplikací se dá použít jazyk C/C++ nebo C#.

2.2.6 Výběr

Mobilní OS	Počet prodaných ks v roce 2010	Podíl na trhu v roce 2010	Počet prodaných ks v roce 2009	Podíl na trhu v roce 2009
Symbian	111 576 700	37,6 %	80 878 300	46,9 %
Android	67 224 500	22,7 %	6 798 400	3,9 %
RIM	47 451 600	16,0 %	34 346 600	19,9 %
iOS	46 598 300	15,7 %	24 889 700	14,4 %
Microsoft	12 378 200	4,2 %	15 031 000	8,7 %
Ostatní	11 417 400	3,8 %	10 432 100	6,1 %
Celkem	296 646 600	100 %	172 376 100	100 %

Tabulka 2.1: Celosvětový podíl smartphonů podle mobilních OS

Podle tabulky 2.1 (viz. [1]) je vidět obrovský nárůst prodeje Smartphonů s OS Android v roce 2010 oproti roku 2009. V roce 2011 obliba Android dále narůstá. Nejen z tohoto důvodu byl vybrán Android jako platforma pro aplikaci. Symbian jako zatím nejoblíbenější OS se instaluje prakticky jen do telefonů Nokia. Naproti tomu Android se vyskytuje u mnoha výrobců mobilních zařízení.

2.3 Popis řešeného problému

Pro účel zapisování poznámek při uživatelském testování, je psaní na mobilním telefonu velice zdouhavé. Než si testující osoba zapíše nějakou poznámku, při testování už může uživatel udělat další věc, kterou je potřeba zaznamenat. Proto je cílem práce navrhnout a vytvořit takovou aplikaci, která toto urychlí.

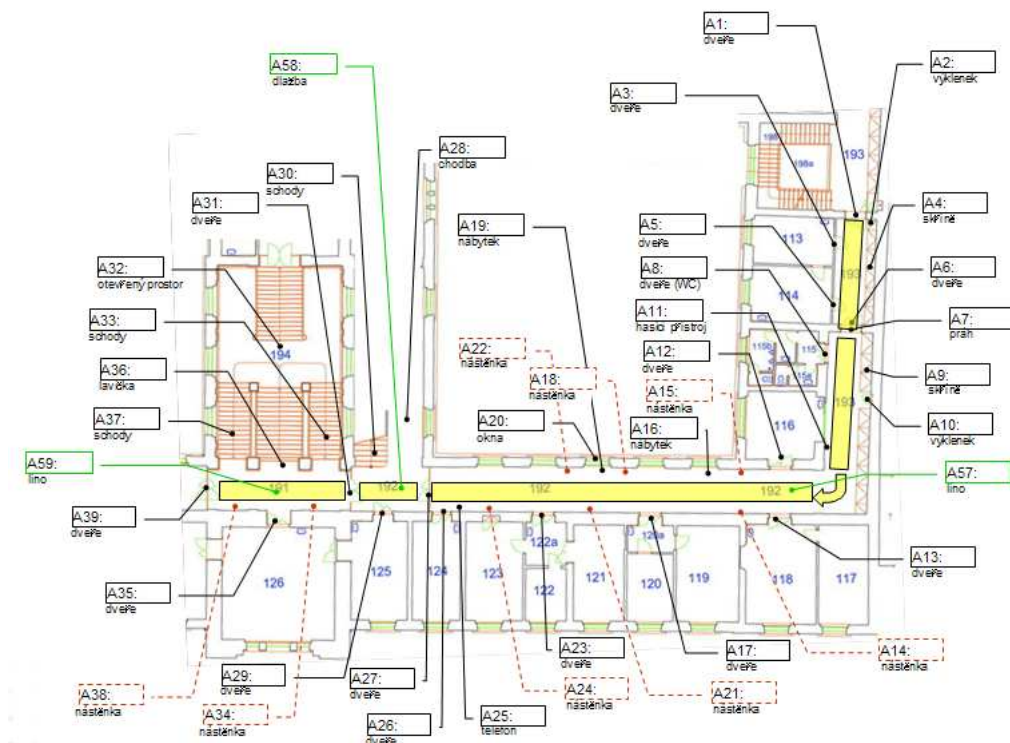
Zadavatel testu zná předem místo, kde se test bude odehrávat. Předem si může vytipovat místa, na kterých by mohl nastat problém a k těmto místům přiřadit události jaké by mohly nastat. Např.: uživatel narazil do zdi, zakopl o květináč atd. Mohou nastat dva druhy událostí:

- jednorázová událost - událost se stane v jeden okamžik a v ten samý okamžik skončí
- dlouhodobá událost - událost se stane v jeden okamžik, poté probíhá dále a v další okamžik skončí. Mezi tím mohou nastat další jednorázové nebo dlouhodobé události.

2.4 Požadavky na implementovaný systém

Aplikace dostane přednastavená data, kterými jsou mapa s oblastí, na které bude probíhat test a popis k předpokládaným místům vzniku událostí.

Na obrázku 2.2 vidíme ukázkou mapy, kde jsou vyznačena předpokládaná místa vzniku událostí. Doposud byl problém řešen tak, že k uvedenému obrázku je tabulka 2.2 a pouze se zapisovalo, zda-li k události došlo nebo ne. To signalizuje sloupec stav. Tabulky se nehodí pro další zpracování. Pokud chceme data z tabulek dále zpracovávat, musíme je znovu přepisovat.



Obrázek 2.2: Náhled mapy s předpokládanými místy vzniku události

ID	Popis	stav(0/1)	detail
A1	dveře	1	
A2	výklenek	0	
A3	dveře	1	
A4	skříň	1	
A5	dveře	0	

Tabulka 2.2: Předdefinované objekty

Cílem je tyto 2 soubory spojit do jedné aplikace. Při výběru místa na mapě by se měl zobrazit připravený popis k onomu místu.

Před spuštěním testu musí být možnost nastavit způsob zapisování času do logu. Jsou dvě možnosti:

- čas se bude měřit jako na stopkách od doby, kdy test začal
- jako čas události se použije systémový čas mobilního zařízení

Výstupem aplikace bude xml log s jednotlivými událostmi. Každá událost bude mít časovou značku kdy vznikla, popř. u dlouhodobých událostí také kdy skončila. Dále pak událost bude mít informace o souřadnicích x,y , kde vznikla. Druhý výstup z aplikace bude obrázek s mapou obsahující grafické značky místa vzniku události.

2.5 Vymezení cílů

Cílem je navrhnout konečnou aplikaci. Ta musí umožňovat:

- načíst předem připravená data,
- zobrazit mapu,
- vyhledat podle místa na mapě zapsanou poznámku,
- rozlišit, zda-li se jedná o jednorázovou, nebo o dlouhodobou událost
- zapsat poznámku do souboru.

2.5.1 Přípravná fáze

Abychom test mohli začít, potřebujeme si připravit data. Nashromáždít informace o lokalitě, kde test bude probíhat, vytipovat místa možných problémů. Tyto informace je potřeba nějakým způsobem předat budoucí aplikaci, která z nich bude číst. Tento krok je důležitý z důvodu urychlení výsledného zapisování poznámek. Pokud bychom žádná data nepřipravili, výsledná rychlost, která je klíčová, bude stejná jako při psaní do souboru na dotykové klávesnici.

2.6 Návrh řešení

2.6.1 Příprava testu

Před samotným testováním je potřeba si připravit test. Jako způsob jsem zvolil vytvoření xml souboru s potřebnými informacemi o testu s následujícím DTD.

```
<!ELEMENT note (fileMap,location+)>
<!ELEMENT location (name,description+,coord)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT coord (x,y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

Xml soubor má kořenový element note. Ten má za potomky element fileMap a 1 nebo více elementů location. V elementu fileMap je uložen název souboru s mapou testu. Tento soubor musí být ve stejném adresáři jako xml soubor. Element location popisuje jednotlivá místa předpokládaných událostí na mapě. Místo musí mít název, 1 nebo více popisů událostí a souřadnice události na mapě. Element coord popisuje souřadnice x,y v obrázku, což představuje jednotlivé pixely počítány od levého horního rohu obrázku.

Způsob přenosu do telefonu jsem zvolil přes web. Tento soubor stáhnu do telefonu přes http protokol. Spolu s ním aplikace stáhne soubor s mapou uvedeným v xml souboru.

2.6.2 Grafické uživatelské rozhraní

Pokud je již příprava testu hotová, může se přistoupit k mobilní aplikaci a začít testovat. Nejprve je potřeba stáhnout výše zmiňovaný soubor do telefonu a nastavit data. To se provede po stisknutí tlačítka Stáhnout nastavení v Úvodní obrazovce (obr. 2.3).

2.6.2.1 Úvodní obrazovka



Obrázek 2.3: Úvodní obrazovka

První obrazovka, která se po spuštění aplikace objeví (obr. 2.3) obsahuje dva aktivní prvky. Tyto prvky jsou tlačítka:

- Spustit test
- Stáhnout nastavení

Po stisknutí tlačítka „Spustit test“ se zobrazí obrazovka s mapou testu (obr. 2.5), po stisknutí druhého tlačítka „Stáhnout nastavení“ se zobrazí obrazovka s nastavením testu (obr. 2.4).

2.6.2.2 Obrazovka s nastavením testu

V horní části obrazovky je text „Umístění nastavení pro test“. Pod tímto textem je text „URL“. Dále je textové pole s předvyplněným textem „http://“. Nakonec jsou pod tímto umístěny tlačítka „OK“ a „Storno“. Po stisknutí OK se z vyplněné adresy zkontroluje, zda-li se tu nachází potřebný soubor k přednastavení testu, pokud ano, tak se soubor stáhne a aplikace se vrátí na úvodní obrazovku. Nyní je nastavení testu připravené. Po stisknutí tlačítka Storno se aplikace vrátí na úvodní obrazovku (obr. 2.3) rovnou bez nějakého nastavování.



Obrázek 2.4: Obrazovka s nastavením testu



Obrázek 2.5: Obrazovka s mapou testu

2.6.2.3 Obrazovka s mapou testu

Tato obrazovka (obr. 2.5) obsahuje okno, ve kterém se zobrazí obrázek s připravenou mapou. Při zapisování poznámek při testu má tato obrazovka klíčovou roli. Odtud se zapisují události

a vždy se na ní aplikace vrací po zapsání události. V levém horním rohu obrazovky jsou umístěny tlačítka lupy + a -. Při stisknutí těchto tlačítek se mapa na obrazovce buď zvětší, nebo zmenší. Při každém dotyku na obrazovku v jakémkoliv místě se zobrazí obrazovka se seznamem vyfiltrovaných událostí. Dále tato obrazovka obsahuje menu, které je dostupné při stisknutí klávesy menu u telefonu s OS Android. Toto menu má jednu položku s názvem „Stop“. Po stisknutí této položky se test ukončí a vytvoří se z databáze log soubor.

2.6.2.4 Seznam vyfiltrovaných událostí



Obrázek 2.6: Seznam vyfiltrovaných událostí

Tato obrazovka (obr. 2.6) obsahuje text „Události“. Pod tímto textem je text „Co se stalo?“. Dále pokračuje seznam s názvy událostí, které jsou dostupné místu předchozího dotyku na obrazovku. Po stisknutí jakéhokoliv řádku kromě posledního, se zobrazí dialogové okno pojmenované „Poznámky“, které obsahuje 2 tlačítka s názvy „Zapsat jednorázovou akci“ a „Začít/Ukončit déle trvající akci“. Při stisknutí poslední položky seznamu se zobrazí okno na úpravu textu. Při stisknutí tlačítka „Zapsat jednorázovou akci“ z dialogu Poznámky, se aplikace vrátí na obrazovku s mapou testu (obr. 2.5). Po stisknutí tlačítka „Začít/Ukončit déle trvající akci“ se zobrazí obrazovka déle trvajících akcí (obr. 2.8).

2.6.2.5 Obrazovka s novou událostí

Obrazovka (obr. 2.7) obsahuje 2 tlačítka.

- „Zapsat jednorázovou akci“
- „Začít/Ukončit déle trvající akci“



Obrázek 2.7: Obrazovka s novou událostí

Při stisknutí tlačítka „Zapsat jednorázovou akci“ se akce zapíše do databáze a aplikace se vrátí na obrazovku s mapou testu (obr. 2.5). Při stisknutí tlačítka „Začít/Ukončit déle trvající akci“ se zobrazí obrazovka dlouhotrvajících akcí (obr. 2.8).

2.6.2.6 Obrazovka dlouhotrvajících akcí



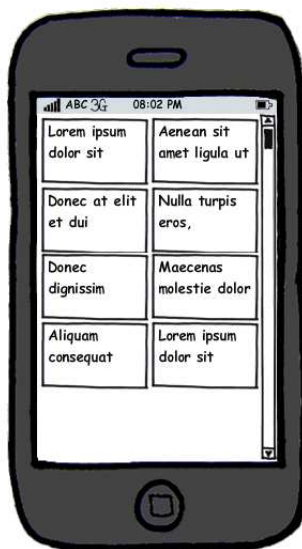
Obrázek 2.8: Obrazovka dlouhotrvajících akcí

Tato obrazovka (obr. 2.8) obsahuje 2 tlačítka:

- Začít novou
- Ukončit probíhající

Po stisknutí tlačítka „Začít novou“ se vybraná akce z předchozí obrazovky zapíše do databáze a aplikace se vrátí znovu na obrazovku s mapou (obr. 2.5). Tlačítko „Ukončit probíhající“ zobrazí obrazovku s právě probíhajícími událostmi (obr. 2.9).

2.6.2.7 Obrazovka s právě probíhajícími událostmi



Obrázek 2.9: Obrazovka s právě probíhajícími událostmi

V této obrazovce (obr. 2.9) se zobrazí tabulka se dvěma sloupci a řádky podle toho, kolik událostí právě probíhá. V každé buňce tabulky je název probíhající události. Po stisknutí nějaké události, se k této události připiše do databáze ukončení události a aplikace se vrátí zpět na obrazovku s mapou (obr. 2.5).

2.6.2.8 Okno na úpravu textu

Tato obrazovka (obr. 2.10) obsahuje textové pole a 2 tlačítka. Tlačítko „OK“ a tlačítko „Storno“. Obrazovka se bude využívat při zapisování událostí, které nebyly předem definovány v přípravě testu.

2.6.3 Datové struktury

V průběhu záznamu testu si aplikace musí pamatovat:

- počáteční nastavení testu,
- informace o zapisovaných událostech,



Obrázek 2.10: Okno na úpravu textu

- zapsané události, ať už jednorázové nebo déle trvající,
- právě probíhající déletrvající události.

Proto je nutné vytvořit odpovídající třídy, které budou reprezentovat tyto informace.

2.6.4 Ovládání

Návrh počítá s tím, že zařízení s Operačním systémem Google Android mají dotykové ovládání. Ve většině obrazovek v aplikaci jsou pouze tlačítka. Jsou ovládána standardně stiskem. Jediná obrazovka, která vyžaduje více druhů ovládání je Obrazovka zobrazující mapu testu. Jelikož se dá po obrazovce posouvat s obrázkem, mohlo by dojít ke komplikaci při rozlišení krátkého stisku a posouvání. Proto je zde rozlišen dlouhý stisk, kterým se bude přidávat nová poznámka.

Kapitola 3

Realizace

Aplikace obsahuje 2 balíčky:

- mutt.logic
- mutt.GUI

3.1 mutt.logic

Tento balíček obsahuje třídy reprezentující datový model používaný v aplikaci.

3.1.1 XML parsování

O čtení dat z xml konfiguračního souboru se stará třída MyXmlHandler. Tato třída přečte xml soubor a vytvoří z něj instanci třídy TestOptions.

3.1.2 Datový model

Třída Note reprezentuje jednotlivou poznámku k zapsání. V této třídě je přepsána metoda toString(). Ta vytváří xml element note. Další metodou této třídy je relativeTime(). Tato metoda vrací instanci třídy Date. Vypočítá dobu uběhnutou od začátku testu.

Třída Notes uchovává informace o zapsaných krátkodobých a dlouhotrvajících událostech. Stejně tak obsahuje informace o právě probíhajících událostech.

Ve třídě TestOptions jsou uchována veškerá data o nastavení testu. Metoda getLocationByCoord() vrací seznam popisů míst odpovídajících předaným souřadnicím, pokud nenalezne žádné, vrací všechny místa. Tato třída se také stará o zapsání logu do souboru.

3.1.3 Datový přenos

O stáhnutí xml souboru a obrázku s mapou se stará třída OptionsCreator. Tato třída má metody generateOptionsFromXML(), getWebXml, getWebImg a changeFilePath.

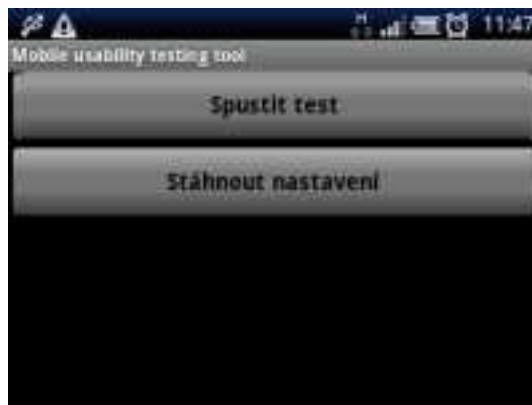
- generateOptionsFromXML() - tato metoda volá na předaný soubor xml handler

- `getWebXml` - z předané URL stáhne xml soubor na sd kartu
- `getWebImg` - z předané URL stáhne soubor s obrázkem na sd kartu
- `changeFilePath` - pomocná metoda, která získává z cesty ke xml souboru, cestu k souboru mapy

3.2 mutt.GUI

Tento balíček obsahuje třídy uživatelského rozhraní OS Google Android.

3.2.1 Hlavní obrazovka



Obrázek 3.1: Úvodní obrazovka

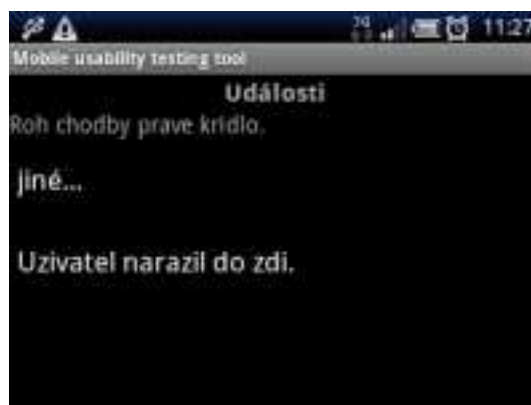
O zobrazení obrazovky po startu aplikace (obr. 3.1) se stará třída `Main`. V případě zavolání metody `btnRunTest.setOnClickListener()` se kontroluje, zda-li je již vytvořená statická proměnná `TestOptions.data`. Pokud ne, tak se zobrazí obrazovka s nastavením testu (obr. 3.8). Pokud ano, tak vytvoří novou instanci třídy `Notes` a nastaví jí jméno ze statické proměnné `TestOptions.data`. Poté tuto instanci uloží do statické proměnné `Notes.data`. V případě zavolání metody `btnDwnSettings.setOnClickListener()` se zobrazí obrazovka s nastavením testu (obr. 3.8).

3.2.2 Seznam vyfiltrovaných událostí

Toto okno (obr. 3.2) zobrazuje třída `AccidentsMenu`. Ze statické proměnné `Notes.data` vezme souřadnice `x` a `y` dotyku na displeji zařízení. Poté získá data z další statické proměnné `TestOptions.data` metodou `getLocationByCoord` a tyto data zobrazí na obrazovku do listu.

3.2.3 Obrazovka s mapou testu

O zobrazení této obrazovky (obr. 3.3) se stará třída `ImgMenu`. Implementuje `OnGestureListener`. Přepisuje metodu `onLongPress()`. V této metodě přepočítá souřadnice dotyku displeje



Obrázek 3.2: Seznam vyfiltrovaných událostí



Obrázek 3.3: Obrazovka s mapou testu

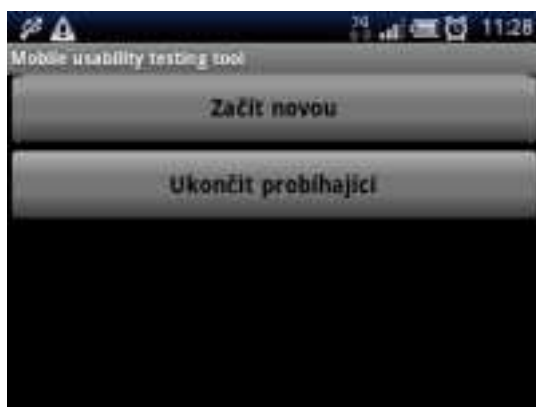
na skutečné souřadnice pixelů na obrázku. Tyto souřadnice uloží do statické proměnné `Notes.data`. Dále přepisuje metodu `onScroll()`. Zde nastavuje posunutí obrázku v `imageView`.

3.2.4 Obrazovka dlouhotrvajících akcí

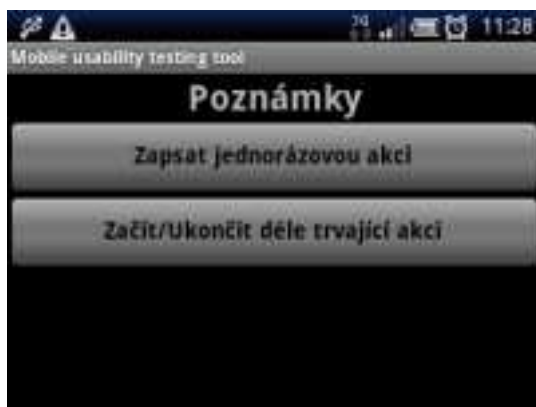
Třída `LongTakingNoteMenu` vytváří obrazovku dlouhotrvajících akcí (obr. 3.4). Nastavuje `clickListeners` na obě tlačítka. V případě zavolání metody `newNote.setOnClickListener()` vytvoří novou instanci třídy `Note`. Nastaví jí data, která vezme ze statické proměnné `Notes.data`. V případě zavolání metody `endRunning.setOnClickListener()` pouze zobrazí obrazovku s právě probíhajícími událostmi.

3.2.5 Obrazovka s novou událostí

O okno s novou událostí (obr. 3.5) se stará třída `NewNoteDialog`. Při zavolání metody `shortAction.setOnClickListener()` se vytvoří nová instance třídy `Note`. Nastaví se jí atributy získané ze statické proměnné `Notes.data`. Nakonec se tato instance přidá do `ArrayListu` statické proměnné `Notes.data.shortNotes`. Poté se obrazovka ukončí zavoláním `finish()`. V



Obrázek 3.4: Obrazovka dlouhotrvajících akcí



Obrázek 3.5: Obrazovka s novou událostí

případě zavolání metody `longAction.setOnClickListener()` se zobrazí obrazovka dlouhotrvajících akcí (obr. 3.4).

3.2.6 Okno na úpravu textu

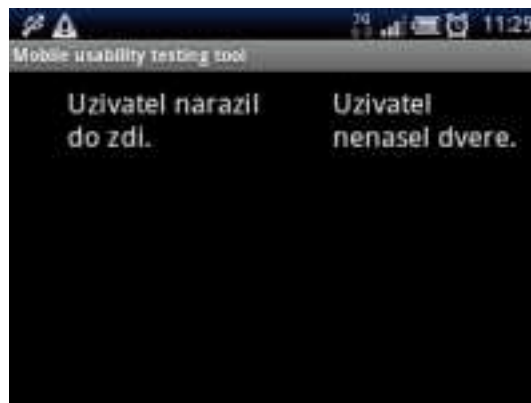
Toto okno (obr. 3.6) zobrazuje třída `NewNoteMenu`. Po zavolání metody `okbtn.setOnClickListener()` nastavuje získaný text do statické proměnné `Notes.data.description`.

3.2.7 Obrazovka s právě probíhajícími událostmi

Třída `RunningNotesMenu` zajišťuje zobrazení této obrazovky (obr. 3.7). Zobrazuje se popis události získaný ze statické proměnné `Notes.data`. Tyto popisy se plní do jednotlivých políček Android GUI elementu `Grid`. Při zavolání metody `grid.setOnItemClickListener()` se zavolá metoda `Notes.data.getAndRemoveRunningLongTakingNote()`, která získá instanci třídy `Note` a zároveň ji odebere z `ArrayListu` právě probíhajících událostí ze statické proměnné `Notes.data`. Této instanci nastaví aktuální čas, jako čas ukončení a přidá tuto instanci do `ArrayListu` dlouhotrvajících událostí statické proměnné `Notes.data`. Poté obrazovku ukončí.

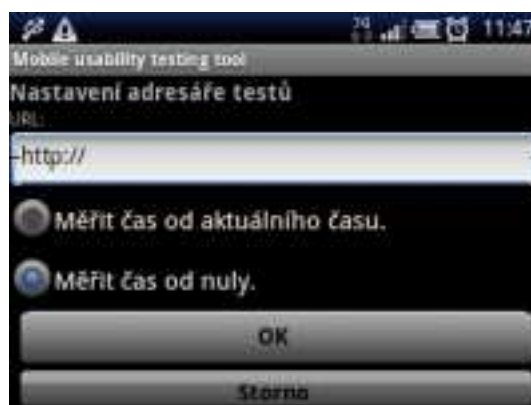


Obrázek 3.6: Okno na úpravu textu



Obrázek 3.7: Obrazovka s právě probíhajícími událostmi

3.2.8 Obrazovka s nastavením testu



Obrázek 3.8: Obrazovka s nastavením testu

Třída `SetupTestFolderMenu` vytváří obrazovku s nastavením testu (obr. 3.8). Obsahuje

InputText na zadání url konfiguračního xml souboru, 2 radio Buttony na nastavení zda-li se bude měřit čas jako absolutní, nebo relativní od začátku testu. Dále pak 2 tlačítka Ok a Storno. V případě zavolání metody `ok.setOnClickListener()` se vytvoří instance třídy `OptionsCreator`. Na této instanci se zavolá metoda `generateOptionsFromXML()`. Dále pak stáhne soubor s mapou definovaný v konfiguračním souboru do telefonu. V případě jakékoli výjimky se zpráva výjimky vypíše na obrazovku v podobě Toast zprávy.

Kapitola 4

Testování

4.1 Zkouška funkčnosti

Funkčnost aplikace jsem vyzkoušel při testech použitelnosti navigačního software v interiéru pro nevidomé. Testování proběhlo v budově Elektrotechnické fakulty na Karlově náměstí. Nevidomý uživatel byl zaveden do čtvrtého patra. Měl za úkol se odtud dostat do Media Labu ve třetím patře.

4.2 Příprava testu

Před testem jsem měl k dispozici mapy obou pater, kde test probíhal a xml soubor s orientačními místy používaný v navigační aplikaci. Podle toho jsem si připravil dva soubory s připravenými místy a uložil je společně se soubory s mapou na server. V příloze uvádím soubor s popisem 4.patra (viz. [C.1](#)).

4.3 Průběh testu

Při testování jsem chodil se skupinou testující uživatele a zapisoval nastalé události. Během testu jsem si zapsal několik událostí. Zde uvádím xml log, který aplikace vytvořila.

```
<log>
  <note>
    <startTime>0h 3m 25s</startTime>
  <description>Uživatel narazil do zdi.</description>
  </note>
  <note>
    <startTime>0h 5m 22s</startTime>
  <description>nerozumi qr kodum</description>
  </note>
  <note>
    <startTime>0h 6m 54s</startTime>
```

```
<description>Uzivatel nenasel dveře</description>
</note>
<note>
  <startTime>0h 3m 31s</startTime>
  <description>tapa</description>
  <endTime></endTime>
</note>
</log>
```

V xml souboru je vidět chyba aplikace. V posledním elementu note je element endTime prázdný. Tuto chybu jsem ve finální verzi opravil.

4.4 Vyhodnocení zkoušky funkčnosti

Aplikace jako taková zafungovala. Objevily se ale nějaké problémy, se kterými původní návrh nepočítal. Když se testovalo v rámci dvou pater, tak se musely připravit 2 různé soubory s nastavením. Tím pádem také vzniknou 2 log soubory s poznámkami, které ale patří jen jednomu testu. Tento problém by se dal vyřešit způsobem výměny obrázku s mapou během testu. V konfiguračním xml souboru, by se muselo vyřešit, jak přiřadit jednotlivá místa ke konkrétní mapě.

Pokud se při zadání události ke konkrétnímu místu nenašla žádná předdefinovaná událost, zobrazí se všechny události zapsané v souboru. Toto řešení se ukázalo jako nešťastné, protože se zobrazovaly stejné popisy vícekrát. A to je dost nepřehledné. Nejlepší by bylo seznam vyfiltrovat a zobrazit stejné popisy jen jednou.

Také způsob načtení konfiguračního souboru není zvolen ideálně. Pro každé načtení xml souboru je dost zdoluhavé vypisování URL. Chybí našeptávání. Tzn. při zadávání adresy by se měly zobrazit již jednou načtené URL souborů. Chybí zde také možnost načíst xml soubor lokálně.

Kapitola 5

Závěr

Zadání je vytvořit aplikaci pro záznam dat o uživatelské činnosti na mobilní zařízení s platformou Google Android se zaměřením na rychlý vstup dat.

Aplikace byla vytvořena na Standard Android platform 2.1-update1. Zadání je splněno z větší části. Jelikož jsem s Android začínal, tak jsem se potýkal s celou řadou problémů, které mi zabraly hodně času vyřešit. Proto jsem již nestihl dodělat část, kdy se do obrázku s mapou zakreslí grafická značka vzniklé události. Zaměření na rychlý vstup dat je splněno s podmínkou, že je dostatečně připravený xml soubor s nastavením. Pokud budou existovat na místech reálného vzniku problému poznámky k tomuto místu, tak je rychlost zapsání poznámky dostatečná. V opačném případě je rychlost stejná, jako v případě psaní poznámek do obyčejného textového souboru. Ale oproti tomu se k této poznámce zapíše i informace o čase a místě vzniku, které se nemusí zapisovat manuálně.

5.1 Pokračování práce

Dalším možným pokračováním práce je možnost dodělení grafického znázornění místa vzniku události. Nabízí se také předělat způsob ukládání zapsaných poznámek do paměti. Zatímco ve stávajícím řešení je ukládání řešeno jen v rámci paměti do ArrayList, výhodnější by bylo použít vestavěnou SQLite databázi.

Dále je možnost vzít v potaz více různých pater v budově, jak jsem již psal v předchozí kapitole.

Literatura

- [1] GREGOR, P. Gartner: Celosvětový prodej mobilů v roce 2010.
<http://www.reselleronline.cz/gartner-celosvetovy-prodej-mobilu-v-roce-2010>,
stav z 14. 02. 2011.
- [2] web:androidDevel. The Developer's Guide — Android Developers.
<http://developer.android.com/guide/index.html/>, stav ze 20. 5. 2011.
- [3] web:appleDevel. iOS Technology Overview - Apple Developer.
<http://developer.apple.com/technologies/ios/>, stav ze 20. 5. 2011.
- [4] web:blackBerryDevel. BlackBerry - Manuals and Guides for Developersp.
<http://docs.blackberry.com/en/developers/?userType=21>, stav ze 20. 5. 2011.
- [5] web:infobp. K336 Info — pokyny pro psaní bakalářských prací.
<https://info336.felk.cvut.cz/?clanek=504>, stav ze 20. 5. 2011.
- [6] web:latexdocweb. — online manuál.
<http://www.cstug.cz/latex/lm/frames.html>, stav ze 20. 5. 2011.
- [7] web:nokiaDevel. Forum Nokia - Develop.
<http://www.forum.nokia.com/Develop/>, stav ze 20. 5. 2011.
- [8] web:usabTestLab. Remote Sensing Using Satellites.
<http://it.coe.uga.edu/~treeves/RSUSeval/>, stav z 1. 10. 1999.

Příloha A

Seznam použitých zkratek

DTD Document Type Definition

GUI Graphical User Interface

HW Hardware

J2ME Java 2 Micro Edition

OS Operační systém

RIM Research In Motion

SQL Structured Query Language

SW Software

URL Uniform Resource Locator

XML Extensible Markup Language

Příloha B

Instalační a uživatelská příručka

B.1 Příprava telefonu

Před instalací aplikace do telefonu je nutné v menu telefonu povolit instalace aplikací, které nepocházejí ze služby market. To provedeme v menu telefonu: Nastavení — Aplikace — Neznámé zdroje. Zde zaškrtneme tuto volbu.

B.2 Instalace

Aplikace OS Android se distribuuje pomocí instalačního souboru s příponou .apk. Zkopírujeme soubor Mutt.apk do úložiště v mobilním zařízení. Odtud tento soubor spustíme souborovým manažerem, čímž se soubor otevře v aplikačním manažeru. Zde spustíme instalaci tlačítkem install. Nyní je aplikace připravena ke spuštění. V menu telefonu se vytvořila spouštěcí ikona Mobile usability testing tool.

B.3 Příprava nastavení testu

Před tím, než začneme aplikaci používat pro zápis poznámek při testování, musíme si připravit data. Na webový server nahrajeme DTD soubor test.dtd. Do stejného adresáře zkopírujeme soubor obrázku s mapou. Poté již vytvoříme xml soubor jako v příloze C.1, kde do elementu fileMap uvedeme název souboru s mapou. Tento soubor také nakopírujeme do příslušného adresáře na serveru.

B.4 Průběh testu

Po spuštění aplikace načteme nastavení testu stisknutím tlačítka Stáhnout nastavení. Do uvedeného políčka zadáme URL adresu xml souboru na serveru. Poté zvolíme jakou možnost zapisování času chceme a stiskneme OK. Pokud bychom toto neudělali a chtěli spustit rovnou test, aplikace nás stejně přeměruje do této obrazovky. Bez počátečního nastavení není možné spustit test.

B.4.1 Přidání jednorázové akce

Dlouhým stiskem stiskneme místo na mapě, kde chceme přidat událost. Objeví se obrazovka (obr. 3.2) se seznamem událostí k místu, pokud nejsou, zobrazí se všechny zapsané události v xml souboru. Máme 2 možnosti:

1. Nastala událost, která je v seznamu
2. Nastala jiná událost

V případě první možnosti stiskneme položku s událostí, kterou chceme zapsat. Objeví se obrazovka (obr. 3.5) a zde stiskneme tlačítko Zapsat jednorázovou akci.

V případě druhé možnosti stiskneme tlačítko jiné... Objeví se obrazovka (obr. 3.6). Napíšeme popis, který chceme zapsat a stiskneme tlačítko OK. Objeví se obrazovka (obr. 3.5) a zde stiskneme tlačítko Zapsat jednorázovou akci.

B.4.2 Přidat déle trvající akci

Pokračujeme stejným způsobem jako při přidání jednorázové akce až po obrazovku (obr. 3.5). Zde stiskneme tlačítko Začít/Ukončit déle trvající akci. Objeví se obrazovka (obr. 3.4). Poté stiskneme tlačítko Začít novou.

B.4.3 Ukončit déle trvající akci

Pokračujeme stejným způsobem jako při přidání déle trvající akce až po obrazovku (obr. 3.4). Nyní stiskneme tlačítko Ukončit probíhající. Zobrazí se obrazovka s probíhajícími akcemi (obr. 3.7). Nyní stiskneme akci, kterou chceme ukončit.

B.4.4 Ukončit test, zapsat poznámky do logu

Pro ukončení testu vyvoláme menu v obrazovce s mapou (obr. 3.3) funkčním HW tlačítkem pro vyvolání menu. Zde stiskneme jedinou ikonu stop. Tímto dojde k vytvoření log souboru na sd kartě mobilního zařízení.

Příloha C

Xml soubory

C.1 Soubor nastavení testu pro zkoušku funkčnosti aplikace

```
<?xml version="1.0"?>
<!DOCTYPE test SYSTEM "test.dtd">
<test>
  <fileMap>4np.jpg</fileMap>
  <location>
    <name>Dvere výtahu. 3.patro</name>
    <description>Uživatel nenasel dveře</description>
    <coord>
      <x>360</x>
      <y>990</y>
    </coord>
  </location>
  <location>
    <name>Dvere na chodbu ke schodisti. 3.patro</name>
    <description>Uživatel nenasel dveře</description>
    <coord>
      <x>1390</x>
      <y>1200</y>
    </coord>
  </location>
  <location>
    <name>Dvere do leveho kridla od schodiste. 3.patro</name>
    <description>Uživatel nenasel dveře</description>
    <coord>
      <x>1434</x>
      <y>1212</y>
    </coord>
  </location>
  <location>
    <name>Dvere do praveho kridla od schodiste. 3. patro</name>
```

```
<description>Uzivatel nenasel dveře</description>
<coord>
  <x>1554</x>
  <y>1206</y>
</coord>
</location>
<location>
  <name>Dveře z praveho kridla ke schodisti. 3. patro.</name>
  <description>Uzivatel nenasel dveře</description>
  <coord>
    <x>1600</x>
    <y>1218</y>
  </coord>
</location>
<location>
  <name>Roh chodby prave kridlo.</name>
  <description>Uzivatel narazil do zdi.</description>
  <coord>
    <x>2400</x>
    <y>1248</y>
  </coord>
</location>
<location>
  <name>Media Lab 318.</name>
  <description>Uzivatel nenasel mistnost.</description>
  <description>Uzivatel nenasel qr kod.</description>
  <description>Problem s nacitanim qr kodu.</description>
  <coord>
    <x>2280</x>
    <y>1107</y>
  </coord>
</location>
<location>
  <name>Zachody prave kridlo.</name>
  <description>Uzivatel nenasel mistnost.</description>
  <coord>
    <x>2293</x>
    <y>985</y>
  </coord>
</location>
<location>
  <name>Kancelar pocitacove katedry 319.</name>
  <description>Uzivatel nenasel mistnost.</description>
  <coord>
    <x>2244</x>
    <y>1398</y>
```

```
    </coord></location>
<location>
  <name>Kancelar 321.</name>
  <description>Uzivatel nenasel mistnost.</description>
  <coord>
    <x>2025</x>
    <y>1398</y>
  </coord>
</location>
</test>
```