

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Eduard Füzesséry**

Studijní program: Softwarové technologie a management

Obor: Web a multimedia

Název tématu: **Mobilní aplikace pro sběr dat z testů použitelnosti v mobilním prostředí**

Pokyny pro vypracování:

Vytvořte mobilní aplikaci pro sledování a záznam mobilních testů použitelnosti v mobilním prostředí. Aplikace bude umět komunikovat pomocí WiFi a nebo Bluetooth s dalšími zařízeními použitými pro test použitelnosti, jako jsou kamery (např. GoPro kamera) nebo další mobilní telefony sbírající data z vestavěných senzorů (minimálně připojení jednoho telefonu). Dále aplikace umožní anotaci testu pomocí poznámek s časovou značkou a případně s odkazem na pozici v mapě, na GPS souřadnici, na obrázek z fotoaparátu nebo ručně psanou nebo hlasovou poznámku. Aplikace bude synchronizovat získaná data se serverovou databází Couchbase. Funkčnost a použitelnost celé aplikace otestujte v jednom testu použitelnosti s alespoň 6 uživateli. Pro implementaci použijte operační systém Android.

Seznam odborné literatury:

M. Jones, G. Marsden, Mobile Interaction Design, Wiley, 2006

R. Meier, Professional Android 4 Application Development, Wiley, 2012

<http://www.funf.org>

Vedoucí: Ing. Ivo Malý, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

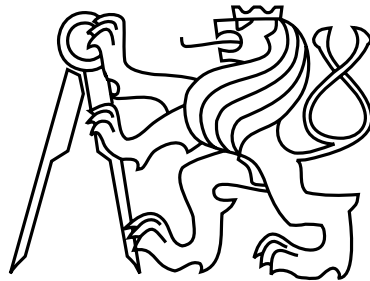



prof. Ing. Jiří Žára, CSc.
vedoucí katedry


prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 25. 3. 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Mobilní aplikace pro sběr dat z testů použitelnosti v mobilním prostředí

Eduard Fúzesséry

Vedoucí práce: Ing. Ivo Malý, Ph.D.

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

22. mája 2015

Pod'akovanie

Úvodom tejto práce by som sa chcel poďakovať za odbornú a technickú pomoc vedúcemu práce Ing. Ivovi Malému Ph.D. a Bc. Ondřejovi Křejířovi. Za psychickú pomoc a podporu najmä mojej rodine, ktorá mi pomohli dostať sa až k tejto práci a zvládnuť ju. Ďakujem vám.

Prehlásenie

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 21. 5. 2015

.....

Abstract

The aim of this thesis was the analysis of usability tests in the actual environment, identification and solution of detected issues. To detect these issues we defined our goals and specified drafts for fulfilling them basing on the analysis of these problems. Three applications were implemented based on the analysis: 2 for operating system Android, first of which is assigned for the moderator of the test (logger) and the second is for participant of the test (data collector). The third application is server which is used for synchronization of data with database common for mobile and desktop applications (IVE-next, created simultaneously with the thesis from Ondřej Krejčíř) . Android applications were checked with usability and functionality tests.

Abstrakt

Cieľom tejto práce bola analýza testov použiteľnosti v mobilnom prostredí. Identifikovať a poriešiť zistené problémy. Identifikácia prebiehala cez zdefinovanie si cieľov a na základe analýzy problému aj návrhov na dosiahnutie týchto cieľov. Na základe analýzy problému boli implementované tri aplikácie: dve na operačný systém Android, prvá pre moderátora (loger) a druhá pre účastníka testu (zbieranie dát), a tretia je serverová aplikácia, ktorá slúži na synchronizáciu dát s databázou spoločnou pre mobilné a desktopovú aplikáciu, IVE-next, ktorá vznikla pri súbežnej diplomovej práci. Android aplikácie boli otestované v rámci testu funkčnosti a použiteľnosti.

Obsah

1	Úvod	1
2	Popis problémov, špecifikácia cieľa	3
2.1	Užívateľské testovanie	3
2.1.1	Testovanie v laboratóriu použiteľnosti	4
2.1.2	Testovanie v mobilnom prostredí	5
2.2	Špecifikácia cieľa	7
3	Analýza a návrh riešení	9
3.1	Prípady použitia	9
3.2	Scenáre	11
3.2.1	Nastavenie testu	11
3.2.2	Prevedenie testu	12
3.2.3	Synchronizácia dát po teste	12
3.3	Vývoj grafického rozhrania	12
4	Realizácia	15
4.1	TestIT	15
4.1.1	Použité technológie	16
4.2	TestITClient	17
4.3	Server	17
4.3.1	Použité technológie	18
5	Testovanie	21
5.1	Popis testu funkčnosti	21
5.1.1	Výsledky testu funkčnosti	22
5.2	Popis testu použiteľnosti	22
5.2.1	Úlohy pre účastníka - moderátora	23
5.2.2	Výsledky testu použiteľnosti	23
5.2.2.1	Implementačné chyby	23
5.2.2.2	Chyby použiteľnosti	24
5.3	Obecný výsledok	24
6	Záver	25
	Literatúra	27

A	Zoznam použitých skratiek	29
B	Obsah priloženého CD	31

Zoznam obrázkov

2.1	Ulab	5
2.2	Rozlíšenie displejov	6
2.3	Ukážka ovešnosti účastníka	7
3.1	Mock-up o inštancie testu	13
3.2	Mock-up vytvorenia testu	14
3.3	Mock-up zoznamu úloh testu	14
4.1	Obrazovka účastníka testu	16
4.2	Aplikácia TestITClient	18

Kapitola 1

Úvod

Počas štúdií na Fakulte elektrotechnickej Českého vysokého učenia technického ma vždy zaujímal návrh „správneho“ užívateľského rozhrania, respektíve, ako sa dá zistiť, či je pohodlné a zrozumiteľné pre koncového užívateľa. Predmet *Testování uživatelského rozhraní* (TUR) mi poskytol solídny základ do tejto problematiky. Súčasne však predmet TUR poukázal na problémy s testovaním užívateľských rozhraní na mobilných zariadeniach, respektíve v prostredí bez usability labu.

Táto práca je rozdelená na 4 hlavné časti. V prvej časti (popis problému) práca objasňuje konkrétne dôvody, prečo je nutný špeciálny nástroj na testovanie mobilných aplikácií, respektíve aplikácií v mobilnom prostredí a prečo nie sú vhodné „klasické“ metódy používané pri testovaní klasických programov na dektopové zariadenia, prípadne pri testovaní „plných“ verzií webových stránok.

V druhej časti (analýza) sa v tejto práci zaoberám otázkou, čo by mal systém spĺňať, respektíve aké funkcie sú požadované. Tieto požiadavky potom roztriedim a presnejšie definujem v rámci prípadov použitia, ktoré sú rozdelené do jednotlivých logických skupín. Súčasne sa snažím presnejšie zdefinovať cieľ tejto práce. Okrem toho v tejto časti uvádzam príklad použitia aplikácie vyvíjanej pri tejto práci na reálnom teste použiteľnosti, ktorý už prebehol. Tieto príklady (scenáre) sú tri, a to: nastavenie, prevedenie testu, pre konkrétnu aplikáciu a konkrétneho účastníka a synchronizácia testu.

V tretej časti (realizácia) popisujem princípy aplikácií a technológie použité pri ich vývoji.

Posledná kapitola (testovanie) obsahuje základný záznam z testov prevedených počas bakalárskej práce. Tieto testy boli dva, prvý na overenie funkčnosti a druhý na overenie použiteľnosti implementovaného riešenia.

Kapitola 2

Popis problémov, špecifikácia cieľa

Táto práca má za cieľ vyriešiť problémy pri užívateľskom testovaní aplikácií/programov v externom prostredí, ktoré prebiehajú počas vývoja. Testovanie väčšinou prebieha v dvoch rovinách. Testovanie funkčnosti a testovanie použiteľnosti. Oba typy testovania sú dôležité z pohľadu user-experience. Programy/aplikácie majú byť implementované tak, aby zvládali všetky situácie, respektíve stavy, do ktorých sa s nimi užívateľ môže dostať a preto testovanie funkčnosti sa dá relatívne ľahko zrealizovať, či už pomocou unit testov¹ alebo pomocou človeka, takzvaného testera. Testovaním funkčnosť sa už bližšie zaoberať nebudem. Testovanie použiteľnosti avšak nie je také „jednoduché“. Testovanie môže prebiehať mnohými spôsobmi, či už metódami bez užívateľa (kognitívny priechod, heuristická evaluácia), alebo metódami s užívateľom. V tejto práci sa zaoberám metódou s užívateľom. Nato, aby bolo možné cieľ presne zašpecifikovať je nutné, aby čitateľ tejto práce vedel, ako prebieha test za normálnych okolností v laboratóriu a ako aktuálne prebiehajú testy v mobilnom prostredí, keďže práca chce vyriešiť problémy týkajúce sa testovania v mobilnom prostredí.

Cieľom týchto testov je nájsť chyby v návrhu užívateľského rozhrania, pri použití v podmienkach, kde sa bude daný program reálne používať. Z tohto dôvodu je veľmi dôležitý kontext testu a z dôvodu získania relevantných výsledkov sa usporiadatelia testu snažia nasimulovať prostredie použitia čo najvernejšie. Tejto snahe sa hovorí ekologická validita testu².

Táto práca si kladie za cieľ poriešiť problémy vznikajúce pri testovaní aplikácií v mobilnom prostredí, teda pomôcť moderátorovi previesť účastníka testom, od ich prvého zvitania sa až k poslednému rozlúčeniu.

2.1 Užívateľské testovanie

Pred uvedením softwarového produktu na trh je zvykom vykonať testovanie použiteľnosti. Tieto testy slúžia na zistenie problémov v návrhu, respektíve takzvanom user experience. Takéto testovanie prebieha (väčšinou) v kontrolovaných podmienkach. Test použiteľnosti prebieha vo viacerých fázach.

¹<http://searchsoftwarequality.techtarget.com/definition/unit-testing>

²<http://study.com/academy/lesson/ecological-validity-in-psychology-definition-lesson-quiz.html>

Prvá fáza je výber účastníkov, základná snaha pri výbere účastníkov je ich mať náhodne vybraných z cieľovej skupiny.

Následne sa pošle vybraným uchádzačom dotazník, tzv screener a na základe toho sa vyberú uchádzači, ktorí spadajú do jednotlivých žiadaných expertných kategórií. Screener má dve časti, verejnú a neverejnú. Vo verejnej časti sú jednotlivé otázky na účastníkov, na určenie jednotlivých vlastností a úrovne znalostí. V neverejnej časti sú priradené jednotlivé percentá ku každej otázke. Tieto percentá určujú finálne zloženie znalostí účastníkov testu.

Po príchode vybraných účastníkov na test nastáva už samotný test použiteľnosti. Tento test sa začína brífingom, ktorý slúži na uvedenie užívateľa do kontextu testovaného programu a na určenie psychického stavu konkrétneho účastníka a teda, či moderátor testu bude pri účastníkovi, alebo môže objasňovať vzniklé situácie pri stakeholderoch. Súčasťou brífingu je zvyčajne aj pre-test dotazník, kde sa zisťujú ďalšie informácie o skúsenostiach a znalostiach účastníka. Okrem toho v tejto fázi prebieha potrebná byrokracia.

Samotný test prebieha vcelku jednoducho, participant na jeho začiatku dostane zoznam úloh, ktoré je potrebné splniť (avšak nič sa nestane, ak to nestihne) a moderátor ho požiada o to, aby myslel nahlas, aby bolo jasné nad čím a ako účastník uvažuje. Počas tejto fázy vzniká najväčšie množstvo dát, po ktorých analýze sa vyvodzujú výsledky a závery.

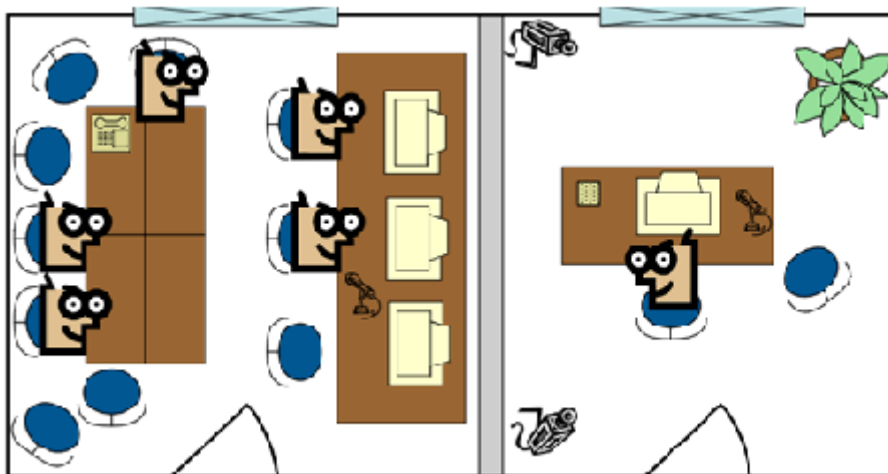
Po teste prichádza debrífing, aby participant, ak je náhodou v strese, sa dostal zase do psychickej pohody a moderátor sa ho v tejto fáze pýta na názory na testovaný program, či mu niečo nebolo jasné a podobné otázky. Tieto dotazníky väčšinou prebiehajú formou rozhovoru.

Táto kapitola vychádza zo zkladnej metodiky testovania použiteľnosti[5].

2.1.1 Testovanie v laboratóriu použiteľnosti

Testovanie programov/aplikácií za zvyčajne okolností prebieha v kontrolovanom prostredí laboratória použiteľnosti, alebo takzvaného Ulabu (usability lab). Ulab sú dve špeciálne zariadené miestnosti, ako je možné vidieť na obrázku 2.1, vhodné na testovanie desktopových programov. V ľavej miestnosti (na obrázku) sa nachádzajú stakeholderi, teda zainteresované osoby (vývojári, manažéri, návrhári užívateľského rozhrania, atď.), zatiaľ čo v pravej miestnosti sa nachádza účastník testu (s moderátorom testu). Moderátor sa nemusí nachádzať len pri účastníkovi, ale môže sa vyskytovať aj pri stakeholderoch. Rozhodnutie, kde moderátor bude, závisí od viacerých faktorov, počtu moderátorov, psychického stavu participanta, psychického stavu stakeholderov, apod). Moderátor je veľmi dôležitý pre test, on určuje čo sa kedy bude diať, on určuje samotný priebeh testu.

Pri testovaní v laboratóriu sa zvyčajne zbiera viac typov dát. Za normálnych okolností je väčšina dát vo video forme. Účastník je sledovaný z viacerých kamier, sleduje sa jeho mimika, jeho výraz, sleduje sa aj okolie a samozrejme sa sleduje interakcia s počítačom, najmä záznam z obrazovky, teda kam sa pozerá, kam kliká a podobné informácie. Okrem video záznamu sa ukladá aj audio, väčšinou separátne z dôvodu viacerých kamier a aby tento zvukový záznam bol, čo najkvalitnejší. Mimo multimediálnych súborov sa počas testu robí záznam (log) kľúčových udalostí počas testu. Tento záznam sa robí kvôli jednoduchšiemu hľadaniu tých udalostí pri analyzovaní dát.



Obr. 2.1: Ulab, FEL, ČVUT v Prahe, autor: Zdeněk Míkovec

Statické prostredie uLabu je vhodné na testovanie dekstopových aplikácií, prípadne webových stránok, pretože podobné laboratória sú navrhované tak, aby pripomínali buď domáce, alebo pracovné prostredie, aby sa zachovala validita testu.

2.1.2 Testovanie v mobilnom prostredí

Oproti testovaniu v Ulabe je testovanie v mobilnom prostredí o dosť komplikovanejšie, pretože kontext použitia mobilných aplikácií nemusí byť stály, respektíve konštantný. Preto nevieme definovať stále prostredie a prostredie testu teda závisí zásadne od testovanej aplikácie. S týmto súvisí nemožnosť získať dostatočné množstvo audio-vizuálnych dát na spracovanie. Z tohto dôvodu potrebujeme získavať dáta z iných zdrojov.

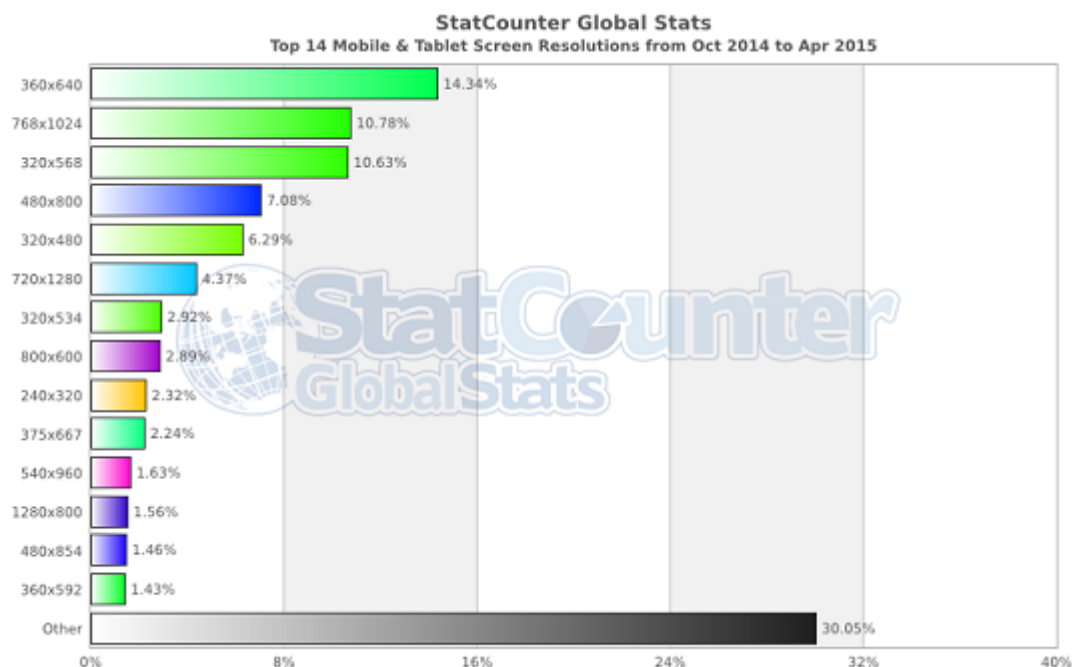
Okrem klasických vecí, čo sú v takom teste sledované (reakcie užívateľa, interakcia užívateľa s programom, komentáre a podobne) sa zbierajú aj iné dáta, ako napríklad GPS poloha, dáta z akcelerometra a podobne. Údaje GPS sa zbierajú najmä pre spätné určenie kontextu chyby, kedy a kde nastala. Údaje z akcelerometra sa napríklad využívajú na určenie typu dopravy účastníka testu. Pri testovaní v mobilnom prostredí je problematický aj zber klasických údajov moderátorom, teda log testu a o reakciách užívateľa ani nehovoriac.

Vďaka nestálemu prostrediu sa dajú efektívne testovať aplikácie závisiace od napríklad GPS súradníc, alebo iných dát, ktoré sa v laboratóriu ťažko simulujú, respektíve sa simulujú nedostatočne.

Treba mať stále na mysli aj isté limitácie samotných mobilných zariadení, ako napríklad slabší výkon týchto zariadení, keďže ešte stále je tam výkonnostný rozdiel a preto použitie niektorých typov aplikácií by nebolo praktické, či už kvôli zlej užívateľskej skúsenosti (napríklad „lagovanie“ aplikácie), alebo aj procesorovo, respektíve graficky náročné aplikácie spotrebúvajú oveľa viac elektrickej energie. V súčasnosti sa snažia výrobcovia mobilných zariadení riešiť tento problém, ale vyzerá to, ako by ho riešili najmä Moorovým zákonom³.

³http://www.webopedia.com/TERM/M/Moores_Law.html

Ďalšie, na čo sa nemôže zabudnúť, sú fyzické rozmery zariadenia, najmä malá veľkosť displeja (v porovnaní s klasickým PC, respektíve notebookmi), čo môže významnou mierou ovplyvniť použiteľnosť zariadenia, a teda aplikácií na ňom zobrazovaných. Testovanie si vďaka týmto rozmerom vyžaduje rôzne nezvyčajné prístupy, ako napríklad <http://www.mrtappy.com/> a podobne. Blízko s týmto súvisí nízke rozlíšenie displejov. Normálne podporujú mobilné zariadenia nižšie rozlíšenia ako desktopové zariadenia. Najpoužívanejšie veľkosti rozlíšení sú uvedené na obrázku 2.2. Štatistika bola robená za obdobie október 2014 - apríl 2015 celosvetovo a skúmali sa rozlíšenia mobilných telefónov a tabletov. Z obrázku je zrejmé, že ešte aj v máji 2015 je najrozšírenejšie rozlíšenie 360x640 pixelov s podielom 14,34%, pričom HD (1280x720) má len 4,37%. Vzhľadom na zobrazenú rôznorodosť rozlíšení je potrebné testovať aplikácie na viacerých rozlíšeníach. Nízke rozlíšenie môže znížiť kvalitu multimediálnych informácií na rôznych zariadeniach.



Obr. 2.2: Rozlíšenie displejov, podľa <http://gs.statcounter.com>

Aktuálne testy prebiehajú za účasti jedného participanta a jedného moderátora. Oba účastníci testu sú v podstate „ovešani“ sledovacou technikou. Moderátor má zariadenie na logovanie participanta, kameru, aby bolo možné sledovať akcie a pohyb participanta a ďalšie vybavenie. Participant (obrázok 2.3)⁴ využíva zariadenie, na ktorom beží aplikácia, ďalšie zariadenie, ktoré meria ďalšie parametre, ako polohu, otrasy a podobne. Tieto popisy vychádzajú z testu prevedeného na Katedre počítačovej grafiky a interakcie - Fakulta elektrotechnická. Čiastočné riešenie tejto situácie je prizvať viac moderátorov do tohto testu, ktorí by pomohli s natáčaním, respektíve logovaním, ale toto riešenie nie je ideálne.

⁴http://forte.fh-hagenberg.at/Project-Homepages/Blindenhund/conferences/granada/papers/VANDENBREDE/VandenBreede-latestpapers_files/BrailleNoteGPS.jpg



Obr. 2.3: Ukážka ovešnosti účastníka

2.2 Špecifikácia cieľa

Práve túto zložitú situáciu ohľadom testovania mimo Ulabu má táto práca zoptimalizovať a uľahčiť život aj participantovi aj moderátorovi testu. Existuje už niekoľko spôsobov ako vzniklú situáciu riešiť, najmä rôznymi stojanmi na kamery, ktoré sledujú obrazovku a užívateľa. Toto riešenie však nerieši problém „ovešnosti“ participanta a moderátora elektronikou, ale naopak, práve ho zhoršuje o tie kamery a stojany. Preto toto riešenie nie je ideálne na mobilné prostredie.

Cieľom práce je vytvoriť nástroje pre zber dát v mobilnom prostredí, zjednodušiť a zefektívniť zber väčšieho množstva dát vznikajúcich pri testoch použiteľnosti, zlepšiť previazanie medzi desktopovými nástrojmi na analýzu a mobilnými nástrojmi na zber dát. Posledný bonusový cieľ je pomôcť moderátorovi s prevedením participanta testom a udržiavať participantov a výsledky testov zosynchronizované naprieč zariadeniami.

Cieľ je takto zameraný najmä na prácu s dátami, pretože počas testu vzniká mnoho dát, ako sme si ukázali v podkapitolách 2.1.1 a 2.1.2, počas samotného testu, ale aj pred, respektíve po konkrétnom teste s užívateľmi. Existuje viac druhov zoznamov, ktoré sú nutné vytvoriť ešte predtým, ako sa pozvú účastníci. Tieto zoznamy sú: úlohy pre účastníkov,

účastníci a možné problémy. Ďalšie potrebné dokumenty sú screener, pre- a post-test dotazníky. Tento typ údajov sa najlepšie vytvára na desktopovom PC, ale zase najideálnejšie využitie je práve na mobilnom zariadení, pretože sú to údaje potrebné, respektíve minimálne pomocné pri testovaní užívateľských rozhraní aplikácií. Preto je potrebný prostredník medzi dektopovými a mobilnými aplikáciami.

Na splnenie vyššie definovaného problému musí riešenie spĺňať isté požiadavky. Samozrejme najzákladnejšou funkciou je, aby boli aplikácie schopné logovať test. Tento log sa má ukladať do telefónu a potom má byť synchronizovateľný so serverom. Ďalšia funkcionality bude umožňovať vytvoriť a spravovať (upravovať) testy, aj na mobilnom zariadení (teda nie len cez server a na to vhodnejšie nástroje). Ďalej riešenie bude umožňovať pridanie multimedialných poznámok, či už zvukových, alebo aj obrazových. Ďalšia časť problému je získavanie dát z účastníkového mobilu. Z tohto dôvodu je potrebné, aby riešenie zbieralo dáta od užívateľa, ako napr GPS a akcelerometer. Ako jeden z posledných problémových bodov vidím synchronizáciu dát so serverom. Podobný problém je aj synchronizácia dát medzi účastníkom testu a moderátorom testu. Vzhľadom na použitie mobilného zariadenia, musí grafický návrh aplikácií spĺňať kritéria zadané v guidelines k systému, používanom v tomto zariadení.

Obece sa požiadavky delia na dve hlavné skupiny:

1. funkčné - definujú, čo aplikácia/program má spĺňať a konkrétne aká funkcionality je od aplikácie požadovaná zadávateľom problému, ktorý treba riešiť
2. obecné - respektíve kvalitatívne, často nazývané nesprávne nefunkčné, ktoré definujú okolnosti a prostredie, v ktorom bude aplikácia používaná

Kapitola 3

Analýza a návrh riešení

Vzhľadom na vyššie spomenuté problémy testovania aplikácií na mobilné zariadenia som sa rozhodol vytvoriť aplikáciu na zjednodušenie procesu testovania, najmä od prvého kontaktu s účastníkom testu.

Najrozumnejšie riešenie je podľa mňa v prepojení dvoch aplikácií na mobilné zariadenia. Jednu aplikáciu bude využívať moderátor testu na správu testov, účastníkov a úloh ku konkrétnym testom a najmä na logovanie konkrétneho testu s jedným užívateľom. Druhá aplikácia bude len zbierať dáta podľa nastavenia testu, tieto dáta budú buď GPS, alebo dáta z akcelerometra, respektíve oba typy. Ďalej bude potrebný server, ktorý zabezpečí synchronizáciu dát medzi desktopovými aplikáciami a aplikáciami vyvíjanými pri tejto práci.

Komunikácia aplikácií navzájom bude prebiehať viacerými smermi. Od aplikácie účastníka pôjdu logy, resp nazbierané GPS/akcelerometrické dáta do zariadenia moderátora, teda zariadenie účastníka nepotrebuje pripojenie k internetu.

Aplikácia pre moderátora hrá v tomto riešení hlavnú úlohu, ona udržiava všetky dáta a jediná ona sa pripája k serveru, a teda ona je zodpovedná za všetku synchronizáciu. Pri zapnutí aplikácie sa aplikácia spýta na všetky projekty zo servera. Po vybratí konkrétneho testu sa pustí synchronizácia a stiahnu sa potrebné dáta zo servera. Po prevedení testu sa zase dáta získané z tohto testu, z tejto aplikácie a aj z aplikácie účastníka, pošlú na server. Pri opúšťaní kontextu testu sa zase pošlú zoznamy účastníkov, úloh a eventov, ak sa nejak zmenili.

Server bude spravovať jednotlivé volania moderátorovej aplikácie a ukladať ich do permanentného úložiska na serveri.

3.1 Prípady použitia

Sú priamym pokračovaním procesu zdefinovania požiadaviek. Prípady použitia, alebo tiež USE-CASES (UC), majú odzrkadľovať funkčné požiadavky. UC pridelujú jednotlivým roliam konkrétne úkony. Vzhľadom na náš problém sú role systému definované už používanými aplikáciami a to moderátor a účastník testu. Všetky tu spomenuté entity, alebo skupiny prípadov použitia majú zdefinované CRUD-ové operácie (Create, Read, Update, Delete) a súčasne aj zobrazenie zoznamu všetkých príslušiacich objektov.

Nástroje pre test - tieto nástroje popisujú netriviálne možnosti, aké môže s testom, ako entitou, moderátor urobiť.

1. Vytvorenie - prípad použitia zaznamenáva postup vytvorenia testu a vyplnenie/nastavenie základných údajov o konkrétnom teste. Potrebné informácie sú: názov testu, testovaná aplikácia a určenie potreby pre-/post-interview.
2. Spustenie - ak je už test vytvorený a má pridané všetky potrebné typy udalostí, úlohy pre účastníkov a aj účastníkov, je možné spustiť test. Jediná vec, čo nebude nastavená sú externé zariadenia. To sa deje po vybratí konkrétneho účastníka zo zoznamu.
3. Stiahnutie - test sa stiahne s nastavenými účastníkmi, úlohami a udalosťami, uloží sa do mobilného zariadenia a sprístupní sa užívateľovi. Následne sa môže akákoľvek informácia doplniť. Po stiahnutí testu do zariadenia je potrebné prídanie zariadení, ktoré môže užívateľ ovládať.
4. Ďalšie prípady použitia sú už len operácie CRUD a zobrazenie všetkých užívateľov.

Nástroje pre užívateľa - okrem operácií spomenutých vyššie (CRUD) obsahuje tento bod ešte pár prípadov použitia a tie sú:

1. Spustenie inštancie testu
2. Synchronizácia dát medzi zariadením užívateľa a moderátora
3. Synchronizácia inštancie testu medzi moderátorovým zariadením a databázou na serveri

Všetky tieto úkony prebiehajú v kontexte užívateľa, najmä z dôvodu, aby sa jednoznačne určil zdroj dát, pri zápise na server.

Nástroje pre typ udalosti (marker) - sú nástroje, pomocou ktorých tvoríme, upravujeme markre/eventy/anotácie a pomocou ktorých ich aj zadávame do logu konkrétneho testu a konkrétneho užívateľa.

1. Zaznamenanie - potom, čo je spustený test, je užívateľovi umožnené zaznamenávať udalosti do logu. K udalosti sa doplní, kedy udalosť nastala, aký typ udalostí nastal a v prípade, že konkrétny typ je nastavený s potrebou fotografie sa aplikácia spýta užívateľa, či chce odfoťiť daný stav. Samotné zaznamenanie je prevedené stlačením tlačidla príslušnej udalosti.
2. Načrtnutie - špeciálny typ udalosti zaznamenaný do logu, obrázok sa uloží do foto-galérie ako fotografia, aby ho bolo jednoducho nájsť a bol ľahko exportovateľný. Do úložiska aplikácie sa uloží čas uloženia obrázka a cesta k nemu.
3. Zvuková poznámka - špeciálny typ udalosti zaznamenaný do logu. Využiteľné skôr pri klasickom testovaní v oddelenej miestnosti. Do logu sa doplní čas začatia nahrávania.
4. Textová poznámka - špeciálny typ udalosti zaznamenaný do logu. V aplikácií sa nachádza text field určený na custom poznámky.

Nástroje pre úlohu - upravujú zmenu flow konkrétnej inštancie testu, teda najmä:

1. Znova pustiť test od istej úlohy - dostupné po dlhom pridržíaní úlohy v profile konkrétneho účastníka
2. Zmena aktuálnej úlohy - pomocou tlačidla na obrazovke testu. Ak sú nastavené pred-, respektíve po-úlohové dotazníky zobrazia sa v tomto momente.

Nástroje pre zariadenie účastníka testu - sú možnosti, ktoré musí spĺňať zariadenie účastníka testu pre funkčnosť celého systému. Bez zariadenia pre účastníka testu, by bolo nepraktické a neefektívne dostávať údaje tohto typu počas, respektíve po teste od účastníka. Zariadenie si nemusí pamätať viac ako jednu sadu dát, z tohto dôvodu bude nutné po každom užívateľskom teste previesť synchronizáciu dát so zariadením moderátora.

1. Synchronizácia údajov so zariadením
2. Záznam GPS polohy
3. Záznam údajov akcelerometra

3.2 Scenáre

V nasledujúcej kapitole popíšem príklad použitia aplikácie vyvíjanej pri tejto práci, respektíve, aký prínos by aplikácia Test IT mala pri testovaní NaviTerier.

Test prebiehal na aplikáciu NaviTerier, respektíve schopnosť nevidiacich navigovať sa navzájom pomocou mobilného telefonu. Situácia bola nasledujúca. Nevidiaci A, ktorý pozná správnu trasu a prešiel si ju, sa snaží navigovať druhú osobu do konkrétneho cieľa pomocou pokynov cez mobilný telefón. Nevidiaci mal dosiahnuť tento cieľ podľa vopred naplánovanej trasy. Hlavný cieľ moderátora testu bolo zaznamenať test a získať potrebné dáta počas testovania.

Na základe tejto situácie boli vytvorené nasledujúce scenáre, vychádzajúce z vyššie popísaného reálneho testu použiteľnosti. Reálny test prebehol na Katedre počítačovej grafiky a interakcie - Fakulta elektrotechnická, Českého vysokého učení technického v Prahe.

3.2.1 Nastavenie testu

Test sa nastaví v desktopovej aplikácii, ktorá vznikla pri súběžnej diplomovej práci, do aplikácie TestIT sa po prihlásení stiahnu automaticky všetky potrebné údaje ako sú zoznamy účastníkov, úloh a špecifických markerov. Súčasne sa stiahnu iné nastavenia testu (pre- a post-test dotazníky) a automaticky sa aplikujú. Po stiahnutí dát by si mal moderátor testu ešte prekontrolovať kompletnosť všetkých údajov.

3.2.2 Prevedenie testu

Pri príchode účastníka moderátor vyberie test zo zoznamu na začiatku, následne sa zobrazí zoznam užívateľov, z ktorého vyberie konkrétnu osobu. Nasledujúca obrazovka je informačná obrazovka o participantovi a je možné z nej rýchlo spustiť test. Ešte je potrebné, aby moderátor zapol logovanie na zariadení pre účastníka a zvolil, ktoré z dostupných typov dát sú potrebné pre správnu analýzu tohto testu, teda v našom prípade logovanie GPS súradníc.

Pred spustením testu sa zobrazí pre-test dotazník s predvyplnenými otázkami a moderátor následne dopíše účastníkové odpovede. Po potvrdení sa zobrazí obrazovka samostatného logovania, teda mriežka eventov (markerov). Táto obrazovka obsahuje ešte tlačidlo na posunutie do ďalšej úlohy a možnosti pridania textových, obrazových a zvukových záznamov. Po teste sa zase zobrazí obrazovka s post-test dotazníkom a predvyplnenými otázkami. Dotazník sa môže odkliknúť ako vybavený. Ak post-test dotazník prebieha niekde inde, aplikáciu sa môže ukončiť a k dotazníku sa neskôr vrátiť vybraním účastníka a upravením jeho profilu a teda aj dotazníkov.

Počas testu jedného účastníka sa vyskytla situácia, že na naplánovanej trase pribudla značka čistenia ulice, ktorú bolo nutné zaznamenať, v tomto prípade sa mohla využiť funkcia fotenia. Ďalšie efektívne využitie aplikácie mohlo byť pri prechodoch účastníka cez cestu na nesprávnom mieste, moderátor by mohol zakresliť, kde presne účastník prešiel, respektíve kedy sa vrátil, teda mohol by nakresliť záznam cesty. A samozrejme sú využiteľné jednotlivé markre, respektíve eventy, či už očakávané (prednastavené), alebo aj neočakávané udalosti, ktoré sú zaznamenateľné pomocou poľa medzi jednotlivými časťami obrazovky.

3.2.3 Synchronizácia dát po teste

Po ukončení testu s účastníkom má moderátor ešte pár povinností. Z dôvodu návrhu aplikácie pre užívateľovo zariadenie, najmä fakt, že nie je nutné, aby si pamätala viac ako jednu sadu dát, musí previesť synchronizáciu dát s aplikáciou pre moderátora, ktorá vyšle výsledky na server, v prípade, že sa neprenesú dáta z moderátorovej aplikácie na server, je možnosť túto synchronizáciu vyvolať opätovne. Okrem tejto synchronizácie (zariadenie-zariadenie) musí prebehnúť aj poslanie dát nazbieraných pomocou aplikácie moderátora.

3.3 Vývoj grafického rozhrania

Po zadefinovaní funkčných a obecných požiadaviek v kapitole 2.2 a braní ohľadov na Android guidelines¹ vznikol low-fidelity prototyp. Prototyp v prvej verzii vznikol pre telefón s uhlopriečkou 4", avšak hneď v druhej iterácii prototypu sa plánované zariadenie zmenilo na 7" tablet, z dôvodu zásadnej zmeny grafického návrhu hlavnej obrazovky testu (obrázok 3.1) a jemných zmenách prípadov použitia. Nasledujúce obrázky sú z poslednej iterácie papierových mock-upov. Okrem tu zobrazenýchh návrhov existujú aj papierové návrhy zbytku užívateľského rozhrania. Všetky tu zobrazené mock-upy boli vytvorené vo webovom nástroji Lucidchart.com.

¹<https://developer.android.com/design/index.html>



Obr. 3.1: Mock-up o inštancie testu

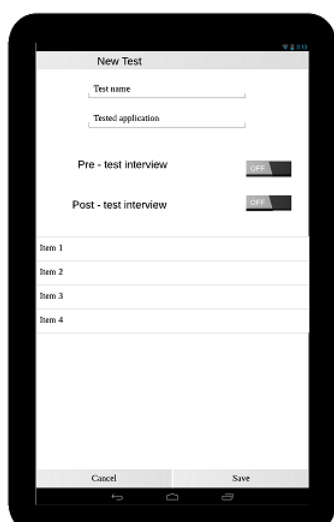
Na obrázku 3.1 vidíme mock-up obrazovky prebiehajúceho testu, pričom grid tlačidiel v spodnej časti obrazovky sú jednotlivé markre, respektíve rýchle odkazy na ne a vo vrchnej časti je miesto na načrtnutie nejakej situácie, pričom jej uloženie prebehne cez kontextové menu.

Na obrázku 3.2 vidíme obrazovku pridania nového testu, kde sú dve položky editovateľné s menom aplikácie. Prvá položka je konkrétna inštancia testu aplikácie uvedenej v druhom editačnom poli. Následne sa na tejto obrazovke nastavujú pre-, respektíve post-test dotazníky a z kontextového menu sa dajú pridať účastníci, úlohy, markre, alebo zariadenia. Všetky tieto údaje sú po pridaní viditeľné v expandable zozname.

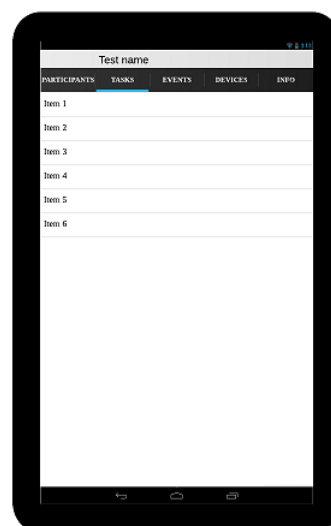
Obrázok 3.3 zobrazuje tri zoznamy a jednu podobrazovku o nastaveniach testu. V troch zoznamoch sa nachádzajú postupne všetci účastníci testu, úlohy pre účastníkov, všetky markre (najprv špeciálne pre aplikáciu a následne sada všeobecných). Štvrtá a posledná podobrazovka obsahuje „statické“ informácie o teste plus možnosť zmeniť ktorúkoľvek informáciu.

Počas vývoja aplikácie nastali ďalšie zmeny grafického návrhu, na základe jednoduchého užívateľského testovania. Vďaka týmto zmenám sa návrh vrátil aj na mobilné telefóny, teda prestal byť orientovaný na tablety. Najpodstatnejšia zmena nastala v presunutí elementu, na ktorý bolo možné kresliť na samostatnú obrazovku.

Ďalšia zmena nastala len v zmene funkčnosti tlačidla na spodnej lište, kde namiesto zoznamu zariadení, získalo miesto tlačidlo, ktoré vyvolá aktivitu na nahranie (a prehranie poslednej) zvukovej poznámky, táto obrazovka je z pohľadu návrhu jednoduchá, keďže obsahuje len dve okrúhle tlačidlá (Nahraj záznam a Prehraj záznam). Tým pádom na obrazovke testu ostal len zoznam markerov/eventov a odkaz na obrazovú, respektíve zvukovú poznámku.



Obr. 3.2: Mock-up vytvorenia testu



Obr. 3.3: Mock-up zoznamu úloh testu

Kapitola 4

Realizácia

Android ako jedna zo základných technológií je operačný systém, na ktorom je postavená táto aplikácia. Tento operačný systém vznikol v roku 2008 ako projekt technologického giganta firmy Google a za ten čas doiteroval do verzie 5.0 Lollipop (lízatko). Využívajú ho nielen mobilné telefóny, ale aj tablety a v poslednom období sa rozmáha Android aj ako Smart TV technológia do televízorov a do inteligentných hodínok. Google priniesol s verziou 5.0 dosť veľké zmeny nielen čo sa dizajnových prvkov týka (za všetko spomeňme zmenu dizajnovej filozofie s názvom Material), ale aj čo sa implementácie týka (zase len jeden príklad zrušenie „dalvika“).

Aplikácie budú naprogramované pre operačný systém Android. Jedna aplikácia bude určená pre moderátora a druhá pre účastníka testu. Programovanie pre systém Android má svoje implementačné špecifiká, či už ako pracuje s životným cyklom aplikácie, alebo aj ako sa používajú jednotlivé funkčné a grafické prvky tohto frameworku (postavené na jazyku Java).

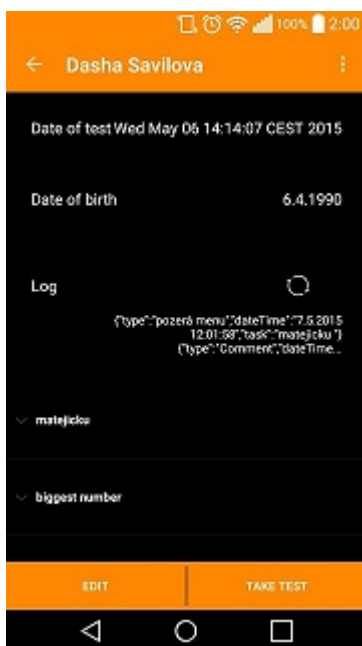
Aktuálny trend vývoja aplikácií pre Android je vývoj pre verziu 4.0.3 (API verzia 15) a novšie (aktuálna verzia API je 22). Z dôvodu použitej technológie, ktorá bude popísaná nižšie, sa muselo obmedziť toto rozpätie pre API aspoň verzie 16, teda Android verzia 4.1. Aktuálne realizovaná aplikácia je teda podporovaná 88,7%¹ všetkých zariadení.

4.1 TestIT

Aplikácia má naimplementované všetky základné obrazovky, najmä čo sa týka CRUD-ových (Create, Read, Update, Delete) úkonov, pre všetky dostupné entity. Aplikácia vie zosynchronizovať údaje obsiahnuté v telefóne s údajmi na serverovej časti CB. Táto synchronizácia prebieha pri výbere konkrétneho testu zo zoznamu všetkých dostupných testov. Test sa stane dostupným po zadaní názvu Bucketu² a jeho hesla, týmto úkonom sa stiahnu príslušné údaje zo servera a test sa uloží do databázy aplikácie. Ďalšia interakcia so serverom prebieha pri ukončení testu, kedy sa všetky zozbierané údaje, najmä log testu a multimedialne súbory pošlú na server, spolu s identifikáciou testu a účastníka, aby server vedel, kam má zaradiť konkrétne údaje. Následne v obrazovke účastníka testu sa môžu údaje zase zosynchronizovať kliknutím na príslušné tlačidlo, ako je vidno na obrázku 4.1.

¹https://developer.android.com/about/dashboards/index.html?utm_source=suzunone

²<http://docs.couchbase.com/admin/admin/REST/rest-bucket-intro.html>



Obr. 4.1: Obrazovka účastníka testu

Aktivita `TestInstance` aktuálne log testu zapisuje do súboru názov `Testu_menoÚčastníka.json`. Každý záznam má aktuálny čas a súčasne čas od začatia testu, aká udalosť sa stala a v ktorej úlohe aktuálneho testu sa to stalo. Log sa ukladá do formátu JSON, formát JSON som zvolil aj kvôli databáze CouchBase, ktorá je postavená na JSON súboroch. Táto aktivita je rovnaká pre každú úlohu a pre každú úlohu sa reštartuje s inými parametrami.

Vývoj prebiehal počas celého školského roka 2014/2015 vo viacerých fázach. Po rôznych papierových a elektronických mock-upoch nastala prvá fáza vývoja. V tejto fáze bolo vytvorenie základného GUI a databázového úložiska aplikácie TestIT. Táto fáza prebiehala počas zimného semestra. Možnosti ukladania permanentných dát na platforme Android je niekoľko (uloženie do pamäte, SQLite databáza, externý databázový systém). Na začiatok som zvolil možnosť uloženia dát do natívnej SQLite databázi, najmä z časových dôvodov. Vzhľadom na použitie CB databázového systému v diplomovej práci Bc. Ondřeja Krejčíře, bola plánovaná zmena na nenatívny NoSQL systém CB.

4.1.1 Použité technológie

Aplikácia TestIT si vyžiadala nasledujúce externé balíky, respektíve ne-vlastný kód.

CouchBase Od aplikácie je vyžadovaná spolupráca so serverom, na ktorom beží NoSQL databáza CouchBase³. Po preštudovaní dokumentácie tohto databázového systému bolo zistené, že najlepšia voľba pre synchronizáciu s mobilným zariadením bude naimplementovať túto databázu (v mobilnej verzii) aj do aplikácie Test IT. Táto verzia databázi je postavená

³<http://couchbase.com/>

na základnej Android SQLite databáze. Táto informácia bola zistená nepríjemne, pri aktualizácii zariadenia, na ktorom sa vyvíja, na Android 5.0, keďže Google zmenil jednu definíciu stĺpca.

SlidingTab Aplikácia obsahuje balíčky stiahnuté z internetu, ktoré som upravil pre konkrétny prípad. Konkrétne sú to balíčky slidingtab (stiahnuté zo stránok developer.android.com, kód bol použitý a uvedený na GoogleIO 2013), balíček je používaný ako grafická nadstavba takzvaných tabov v triede: TestActivity.

WiFi P2P Balík services je použitý na peer to peer komunikáciu medzi dvoma android zariadeniami (najmä medzi aplikáciami TestIT a TestITClient) na synchronizáciu súborov od užívateľa najprv so zariadením moderátora a následne so serverom. Kód bol prevzatý a následne upravený z <https://android.googlesource.com/platform/development/+master/samples/WiFiDirectServiceDiscovery>

GSON Balík slúžiaci na prevod JSON objektov do objektov v kóde a naopak.

DrawingActivity Posledný externý balík je z webovej stránky <http://code.tutsplus.com/> a využíva sa na podporu DrawingActivity, teda aktivita, ktorá umožňuje kresliť a ukladať obrázky moderátorom. V tomto prípade som nasledoval tutoriál.

4.2 TestITClient

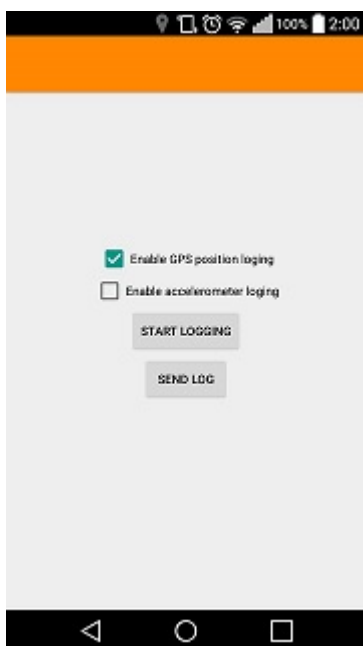
Po vyvinutí aplikácie pre účastníka, ktorá loguje GPS dáta a dáta z akcelerometra, bolo nutné naimplementovať vzájomnú komunikáciu. Z tohto dôvodu je v oboch aplikáciach pridaný komunikačný balík. Tento balík vyžaduje pripojenie aplikácií do rovnakej WiFi siete, ale po vybratí zariadenia (je jedno na ktorom zariadení) a potvrdení spojenia na druhom zariadení sa dáta odošlú takmer automaticky. Moderátorovo zariadenie okamžite po získaní dát pošle dáta na server. Zariadenie účastníka testu si dáta drží od začiatku testu, až kým nie je súbor prepísaný.

Riešenie obsahuje aj klientskú verziu aplikácie. Prvotné myšlienky boli ohľadom využitia frameworku Funf. Klientská aplikácia má vedieť logovať GPS dáta a dáta akcelerometru. Vzhľadom na jednoduchosť zberu tohto typu dát a najmä jednoduchosť implementácie, padlo rozhodnutie nepoužiť tento framework, ako zbytočne obsiahly nástroj na takto jednoduché úkony. Screenshot reálnej aplikácie vidíme na obrázku 4.2.

Použité technológie nie sú o nič iné oproti knižniciam a technológiám použitým v aplikácií TestIT. Využívajú sa avšak len 2 GSON a balíček pre WiFi P2P pripojenie.

4.3 Server

Súčasťou riešenia je aj napojenie aplikácie moderátora na server, na ktorom beží databáza CouchBase (ďalej len CB). Pôvodná idea bola využitie CB SDK pre operačný systém



Obr. 4.2: Aplikácia TestITClient

Android, avšak od tej sa opustilo a preto vznikla potreba vyvinúť server. Od implementácie tohto riešenia sa upustilo najmä z dôvodu už fungujúceho databázového systému a efektívnejšej práce s novým (Java) serverom, ako s (javascriptovým) riešením v podobe SyncGate od CB. Server využíva REST API a integračnú vrstvu z už spomínanej diplomovej práce Ondřeja Krejčíře, ktorá sa napája na server, kde beží CB databáza. Integračná vrstva je importovaná do serverovej časti tejto bakalárskej práce. Server v tomto prípade spĺňa účel jednoduchej brány, ktorá získava dotazy od mobilnej aplikácie a posíla ich upravené do databázi CB.

Server slúži ako brána pre mobilné zariadenia a napája sa na verejnú databázu CB, ktorá sídli na adrese „ive.felk.cvut.cz“. Server je napísaný v jazyku Java a využíva knižnice Grizzly a Jersey pre zadenovanie REST API. Klasický scénar použitia tohto servera je nasledovný. Po zapnutí mobilnej aplikácie pre moderátora sa zariadenie spýta na zoznam testov (projektov), v tomto prípade server pošle požiadavku cez integračnú vrstvu, na CB server a jeho odpoveď pošle ako odpoveď mobilnému zariadeniu ako JSON súbor. Po vybratí testu sa dáta stiahnu zo servera do moderátorovho telefónu a tým pádom prebehne synchronizácia. Po konkrétnom teste s užívateľom sa pošle aktuálny log na server. Týmto sa zabezpečí synchronizácia dát.

4.3.1 Použité technológie

Na rozdiel od Androidu sa v tomto prípade začínalo s relatívne čistou Javou. Na nápravu toho sa použila technológia Maven a cez ňu sa doplnili ostatné potrebné technológie.

Maven - technológia slúžiaca k správe závislostí projektov. Definuje sa pomocou XML súboru pom.xml.

org.simple.JSON - technológia pre prácu s JSON objektami, keďže nie je to natívny dátový typ Javy.

Jersey + Grizzly - technológie zabezpečujúce beh serverovej aplikácie. Grizzly je server, na ktorý je možné sa dotazovať a služba Jersey nám zaistuje práve to REST API.

Integračná vrstva - posledný serverový balíček je integračná vrstva z diplomovej práce Ondřeja Krejčíře, ktorá zabezpečuje pripojenie na databázu CouchBase, bežiacu na tom istom serveri, ale na inom porte.

Kapitola 5

Testovanie

Práca bola v priebehu vypracovania testovaná, tieto testy boli robené bez užívateľov, najmä heuristickou evaluáciou, týmto štýlom sa iterovalo medzi rôznymi grafickými návrhmi. Na konci implementačnej časti bol prevedený test funkčnosti a použiteľnosti aplikácií.

Prvý test bol namodelovaný na základe v minulosti prevedeného užívateľského testovania. Účastníci testu prechádzali rovnakou, respektíve podobnou trasou ako participanti reálneho testu, avšak s odlišnými úlohami, keďže reálny test bol uskutočnený so slepými účastníkmi. Tento test bol test funkčnosti aplikácií. Hlavná úloha testu bola určiť a nájsť chyby programu, respektíve nájsť situácie, kedy by mohli aplikácie padnúť.

Druhý test bol klasický test použiteľnosti v laboratóriu určenom na testovanie použiteľnosti. Tento test mal za cieľ nájsť chyby použiteľnosti aplikácií. Všetci účastníci tohto testu boli absolventi predmetu TUR (testování uživatelského rozhraní) na Fakulte elektrotechnickej, Českého vysokého učení technického. Účastníci testu boli zvolení absolventi tohto predmetu najmä z dôvodu absolvovania užívateľského testovania ako moderátori.

5.1 Popis testu funkčnosti

Účastníci mali splniť jednoduché úlohy, ktoré sa týkali prostredia a konkrétne ulíc: Dittrichova, Záhořanského, Na Zderaze, Na Zbořenci. Týmto testom sa testovala najmä technická funkčnosť aplikácií. Užívateľské rozhranie aplikácie pre moderátora bolo testované s jedným užívateľom v ten istý deň.

Úlohy, ktoré mali účastníci testu plniť

1. Dôjdite k reštaurácii „U Matejíčku“ a zistite cenu denného menu 1.
2. Pri najvyššom čísle domu Dittrichovej ulice urobte 1 drep.
3. Každú lampu obehnite dookola.
4. Dôjdite na roh ulíc Záhořanského a Na Zderaze a zistite materiál a jedno/dvojkřídlovost najbližších dverí na ulici Na zderaze.

5. Na sever od vašej aktuálnej pozície je vaša posledná úloha.
6. Pomocou ulice Na Zbořenci sa dostanete k soche zavesenej na dáždniku, urobte si s ňou fotku.

5.1.1 Výsledky testu funkčnosti

Test funkčnosti bol urobený s troma užívateľmi, pričom účastníci testu sa pohybovali na už uvedených uliciach a na telefóne HTC Desire 500 bežala aplikácia TestITClient a bola nastavená na logovanie GPS súradníc. Na telefóne moderátora bežala aplikácia TestIT, kde boli prednastavené údaje o užívateľoch a úlohách, ktoré majú plniť.

Aplikácie fungovali podľa očakávania a, tým sa overila použiteľnosť navrhovaného riešenia v praxi. To znamená, že boli vytvorené zoznamy účastníkov, zariadení, markerov, aplikácia TestITClient úspešne zozbierala dáta z GPS a potom sa tieto dáta synchronizovali do aplikácie TestIT. Aplikácia pre moderátora fungovala a robila anotácie tak, ako bolo očakávané. Napriek tomu počas testu nastalo pár problémov, ktoré boli po analýze opravené.

Napriek zameraniu testu bolo zistených viac problémov použiteľnosti, pri zachovaní funkčnosti. Bola zistená jedna chyba funkčnosti a tá bola po teste odstránená okamžite. Konkrétne to bola chyba, že po poslednej úlohe aplikácia spadla, údaje sa avšak zachovali, keďže sa zapisujú okamžite, ako sú zaznamenané. Aplikácia neposlala údaje na server po skončení testu, ale tie sa zosynchronizovali na konci celého testu.

Problémy použiteľnosti boli avšak problematickejšie a podľa môjho názoru aj dôležitejšie. Vzhľadom na zameranie aplikácie na mobilné a teda väčšinou externé prostredie je dôležité zachovávať vysoký kontrast všetkých textov, na tento fakt som pri vývoji zabudol a počas testu s používateľmi bol tento usability problém asi najcitelnejší, keďže bol slnečný deň a ani s maximálnym jasom nebola čitateľnosť ideálna. Aplikácia je obecné tmavá, čo nevádi, ale z tohto dôvodu musia byť svetlé/biele písmená aby sa zabezpečila čitateľnosť na slnku.

Vzhľadom na zistenú potrebu zadávania veľkého počtu nepredpripravených poznámok sa vyskytla ďalšia chyba použiteľnosti a to, že po úspešnom zadaní poznámky nedostal moderátor žiadnu odozvu systému a navyše ostával text v grafickom prvku EditText, a to zásadným spôsobom obmedzovalo použiteľnosť aplikácie.

Posledná zásadná chyba nastala zase pri zadávaní nepredpripravenej poznámky. Tlačidlá „Next task“ a „Submit“ boli pri vysunutej klávesnici blízko seba, a teda nastala situácia, kde namiesto uloženia textovej poznámky sa posunulo na ďalšiu úlohu.

Počas testovania som si uvedomil nepotrebnosť mena, respektíve identifikátoru účastníka v Toolbar-e/ActionBar-e.

Po každom účastníkovom teste prebehla synchronizácia so zariadením moderátora, ktorý po ukončení všetkých testov dáta úspešne zosynchronizoval so serverom. Z tohto dôvodu vieme povedať, že test bol z funkčného hľadiska úspešný, ale boli zistené relatívne zásadné problémy použiteľnosti, ktoré boli opravené.

5.2 Popis testu použiteľnosti

Prebiehal s dvoma účastníkmi, uvedomujem si, že dvaja účastníci nie sú ideálna situácia, ale aj títo dvaja užívatelia odhalili relatívne dosť chýb použiteľnosti. Okrem tohto

obaja účastníci sa dajú brať ako experti, keďže už poznajú procesy neverejnej časti testu. Všetci účastníci boli absolventi/študenti fakulty FEL, ČVUT a najmä absolvovali predmet TUR (Testování uživatelského rozhraní), a teda už boli v úlohe moderátora testu a poznajú testovanie použiteľnosti aj z tejto neverejnej strany. Jeden účastník bol užívateľ telefónu iPhone, čiže operačného systému iOS a preto užívateľ potreboval pomoc pri interakcii so štandardnými grafickými prvkami Androidu, ako napríklad DatePicker.

5.2.1 Úlohy pre účastníka - moderátora

1. Vytvorte test aplikácie NaviTerier verzia 1.0, pričom nechcete ani pre-test ani post-test interview.
2. Pridajte účastníka testu Jan Čáp, narodeného 15.4.1992, s poznámkou, že to je študent FELu.
3. Pridajte úlohu „Nainštalujte aplikáciu“ bez popisu.
4. Začnite test s vami vytvoreným užívateľom
5. Napíšte správu: „mam sa skvelo“.
6. Nahrajte zvukovú poznámku podľa vašeho výberu.
7. Pridajte event „Give up“.
8. Ukončíte test

5.2.2 Výsledky testu použiteľnosti

Účastníci sa v aplikácií nestrácali a vedeli sa v nej orientovať, väčšina ovládacích prvkov bola na očakávateľných miestach a teda test prebehol úspešne. Napriek úspešnosti testu boli odhalené aj chyby použiteľnosti a implementácie. Táto kapitola je rozdelená na dve podkapitoly, a to chyby implementačné a chybu použiteľnosti.

5.2.2.1 Implementačné chyby

Prvá chyba sa vyskytla už pri pridávaní participanta, keďže po úspešnom pridaní sa nezobrazil v zozname účastníkov. Analogická chyba bola pri pridaní úlohy/markeru.

Podobná chyba je aj po ukončení testu, keďže sa v obrazovke účastníka zobrazuje výsledný log a k nemu synchronizovacie tlačidlo sa nezobrazilo po ukončení testu.

Ďalšia implementačná chyba nastala pri pridávaní úlohy, vzhľadom na celkovú zmenu práce s hlavnou obrazovkou testu. Nie je potrebné pridávať k úlohe obrázky, z dôvodu zmeny návrhu grafického rozhrania obrazovky. Pôvodný plán bol ohľadom zobrazenia obrázku úlohy/mapy vo vrchnej časti obrazovky testu ale zmenili sa návrhy, ale neprebehla zmena kódu.

5.2.2.2 Chyby použiteľnosti

Účastníkom chýbalo potvrdenie o zalogovaní akcie, či už písaná poznámka alebo aj poznámka z mriežky markerov/eventov. Naplánované riešenie tejto chyby je zobrazovať Toast pri pridaní anotácie. Užívateľ bude môcť vypnúť túto funkcionality z nastavení aplikácie, keď už bude aplikácii dostatočne dôverovať.

Aktuálna verzia nie je konzistentná, čo sa pridávania multimediálnych súborov týka, resp ich layoutov. Ďalší problém videl účastník v rozdielne udávanej dôležitosti týchto akcií, keďže jedna akcia má len položku options menu (kreslenie) a druhá má samostatné a stále viditeľné tlačidlo. Obecné sa účastníkom nepáčilo spodný pruh dvoch tlačidiel, keďže pri vysunutej klávesnici zaberajú pomerne dosť miesta. Navrhované riešenie je vymazanie spodného pruhu tlačidiel a presunutie príslušných eventov do kontextového menu.

Na aktivite participanta sa účastníkom zdali blízko tlačidlá synchronizácie testu a vymazanie účastníka. Aktuálne nie je implementované potvrdzovacie okno vymazania, na čom sme sa zhodli, že by implementované byť malo.

5.3 Obecný výsledok

Z týchto testov vyplýva, že riešenie navrhnuté v kapitole 2 je použiteľné a realizovateľné, ale je potrebné ešte dopracovať problémy, ktoré sa týkajú užívateľského rozhrania a súčasne najmä prepojenia jednotlivých aplikácií (desktop-mobil).

Kapitola 6

Záver

Táto práca mala za cieľ vytvoriť IT riešenie na pomoc pri testovaní aplikácií, respektíve webových stránok využívajúcich mobilný kontext. Aktuálne sú vytvorené dve Android aplikácie „logger“ a „collector“ a serverové riešenie, ktoré slúži ako synchronizačná brána medzi mobilnými a desktopovými aplikáciami. V prvom rade prebehla analýza problémov, kde boli zadefinované problémy pri testovaní v mobilnom prostredí.

Následne som tieto problémy rozobral a vyšpecifikoval cieľ (teda vytvorenie troch aplikácií). Po zanalyzovaní problémov zadefinovaní cieľa som sa pustil do vývoja grafického rozhrania, za ktorým tesne nasledovala implementácia aplikácií.

V poslednej časti tejto práce je uvedený záznam dvoch testov použiteľnosti a funkčnosti, ktorými som overil, že navrhnuté riešenie principiálne funguje, ale z dôvodu existujúcich nedostatkov nie je ešte ideálne.

Vzhľadom na môj záujem o túto problematiku by som chcel do budúcnosti pokračovať v tejto práci, a teda odladiť aplikácie na Android, aby boli užívateľsky prívetivé. A mojím ďalším cieľom je zlepšenie prepojenia aplikácií, ktoré bežia na desktope a pri tejto práci vyvinutých aplikácií.

Literatúra

- [1] KAIKKONEN, Anne, Aki KEKÄLÄINEN, Mihael CANKAR, KALLIO a KANKAINEN. *Usability testing of mobile applications: a comparison between laboratory and field testing*. [online]. 2005, č. 1, s. 4-16 [cit. 2015-01-04]. Dostupné z: <http://lib.tkk.fi/Diss/2009/isbn9789522481900/article6.pdf>
- [2] DEY, Anind, Gregory ABOWD a Daniel SALBER. *A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*. *Human-Computer Interaction*. 2001-12-1, vol. 16, issue 2, s. 97-166. Dostupné z: <http://informaworld.com/openurl?genre=article>
- [3] ZHANG, Dongsong a Boonlit ADIPAT. *Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications*. *International Journal of Human-Computer Interaction*. 2005, vol. 18, issue 3, s. 293-308. Dostupné z: http://tandfonline.com/doi/abs/10.1207/s15327590ijhc1803_3
- [4] *Learn REST* Dostupné z: <http://rest.elkstein.org>
- [5] RUBIN, Jeffrey a Dana CHISNELL. *Handbook of usability testing: how to plan, design, and conduct effective tests*. 2nd ed. Indianapolis: Wiley Publishing, c2008, xxxvi, 348 s. ISBN 9780470185483.

Dodatok A

Zoznam použitých skratiek

API Application Programm Interface

CB CouchBase

CRUD Create, Read, Update, Delete












JSON JavaScript Object Notation

REST Representational State Transfer

Dodatok B

Obsah priloženého CD

 /

-  `project` – zložka s projektom
 -  `Install` – zložka s inštalačnými súbormi
 -  `TestIT` – zložka s Android aplikáciou pre moderátora
 -  `res` – zložka s layout súbormi
 -  `src` – zložka so zdrojovými kódmi
 -  `TestITClient` – zložka s Android aplikáciou pre účastníka testu
 -  `res` – zložka s layout súbormi
 -  `src` – zložka so zdrojovými kódmi
 -  `TestITRest` – zložka so serverovou aplikačnou bránou
 -  `src` – zložka so zdrojovými kódmi
-  `.pdf` – PDF verzia tohto textu písaná v *LaTeX*