



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MOBILNÍ APLIKACE PRO POPTÁVÁNÍ SLUŽEB

MOBILE APPLICATION FOR A SERVICE REQUESTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ANDREJ BLIŽŇÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Bližňák Andrej**

Obor: Informační technologie

Téma: **Mobilní aplikace pro poptávání služeb**
Mobile Application for a Service Requests

Kategorie: Softwarové inženýrství

Pokyny:

1. Nastudujte základy implementace v Android OS. Identifikujte vhodný trh/službu a jeho požadavky na mobilní aplikaci pro nabídku a poptávku zvolené služby.
2. Navrhněte a implementujte aplikaci.
3. Aplikaci otestujte na vhodném vzorku beta-testerů.
4. Pokračujte v implementaci a změnách GUI pro co nejlepší UX. Získejte zpětnou vazbu od uživatelů. Zveřejněte aplikaci na Google Play.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Podle pokynů školitele

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 ze zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szóke Igor, Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

L.S.



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Táto práca sa zaoberá riešením problematiky vývoja aplikácií pre Android OS. Riešenie problematiky zahŕňa analýzu trhu potenciálnych užívateľov, analýzu trhu s mobilnými aplikáciami, návrh a implementáciu riešenia. V práci sú tiež zahrnuté základné znalosti potrebné pre implementáciu samotného užívateľského rozhrania, ale aj technológia slúžiaca backendu aplikácie. Priebežné výsledky práce sú potom publikované a testované na vzorke užívateľov.

Abstract

This thesis deals with the solution of the development of applications for OS Android. The solution contains analysis of market of potential users, applications market analysis, design and implementation of solution. Thesis also contains basic knowledge needed for implementation and it contains backend technology basics needed for application. Results of the work are then published and tested by group of users.

Kľúčové slová

Android, Google Firebase, užívateľské rozhranie, užívateľská skúsenosť

Keywords

Android, Google Firebase, user interface, user experience

Citácia

BLIŽŇÁK, Andrej. *Mobilní aplikace pro poptávání služeb*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Szóke Igor.

Mobilní aplikace pro poptávání služeb

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením doktora Igora Szókeho. Uviedol som všetky literárne pramene a publikácie z ktorých som čerpal.

.....
Andrej Bližňák
11. mája 2017

Podakovanie

Veľká vďaka patrí pánovi doktorovi Igorovi Szókemu za odborné vedenie práce a rady, ktoré dopomohli k tvorbe tejto práce. Rád by som sa tiež podakoval všetkým ochotným ľuďom, ktorí či už testovali samotnú aplikáciu, alebo sa aspoň zúčastnili prieskumov. Vďaka tiež patrí ľuďom, ktorí ma v práci podporovali hlavne v ťažších momentoch pri tvorbe práce a tolerovali vonkajšie prejavy mojej frustrácie a pomohli vrátiť mi motiváciu. Je to hlavne blízka rodina a kamaráti.

Obsah

1 Úvod	3
2 Analýza trhu a jeho požiadavky	4
2.1 Analýza užívateľov	4
2.2 Analýza trhu s mobilnými aplikáciami	6
3 Návrh aplikácie	8
3.1 Všeobecný postup užívateľa	9
3.2 Postup užívateľa typu <i>cukrár</i>	10
3.3 Postup užívateľa typu <i>zákazník</i>	11
3.4 Architektúra aplikácie	12
4 Základy implementácie pre operačný systém Android a mobilná platforma Google Firebase	14
4.1 Operačný systém <i>Android</i>	14
4.2 Vývojové prostredie <i>Android Studio</i>	14
4.3 Dôležité súbory	15
4.4 Mobilná platforma <i>Google Firebase</i>	16
4.4.1 Autentifikácia	17
4.4.2 Reálno-časová databáza	18
4.4.3 Úložisko	22
4.4.4 Cloudové správy	23
5 Ostatné časti architektúry	25
5.1 Vlastný server	25
5.1.1 Verzia <i>1.0</i>	25
5.1.2 Verzia <i>2.0</i>	26
5.2 <i>Offline</i> databáza	27
5.3 Ostatné služby	28
6 Implementácia užívateľského rozhrania	31
6.1 Aktivity a fragmenty	31
6.2 Vlastné adaptéry	32
6.3 Dialógové okná	32
6.4 Systémové a iné rozhrania	32
6.5 Jazykové verzie	33

7 Publikovanie a testovanie aplikácie	34
7.1 Publikovanie aplikácie	34
7.2 Testovanie	35
7.3 Budúcnosť aplikácie	37
8 Záver	38
Literatúra	40
Prílohy	41
Zoznam príloh	42
A Analýza užívateľov - dotazník s odpoveďami	43
B Pracovný postup	46
C Nárhy obrazoviek pre verziu 1.2	50
D Obsah CD	51

Kapitola 1

Úvod

Mobilné aplikácie v súčasnosti zažívajú značný rozmach týkajúci sa rôznych smerov a zameraní. Táto skutočnosť dáva vývojárom množstvo príležitostí k iniciatíve a bola by škoda nepreskúmať a nevyskúšať dnešné technológie k vytvoreniu praktickej aplikácie.

Cielom tejto práce je poukázať na riešenie spomínaných problémov spojených s implementáciou aplikácií na platformu *Android OS* s ohľadom na súčasné trendy. Z takýchto trendov je zaujímavé hlavne využitie tzv. *cloudových* služieb, ktoré sa začínajú uplatňovať v rôznych smeroch a využitíach. Pravdepodobne tiež je, že takéto služby budú v budúcnosti viac a viac rozšírené, na čo musia reagovať aj samotní vývojári nielen mobilných aplikácií.

Hlavnou myšlienkou samotnej aplikácie je poskytnutie platformy pre poskytovateľov služieb hlavne menšieho charakteru. Rôzni podnikatelia a predajcovia, hlavne strední a väčší, sa v dnešnej dobe začali sústreďovať na kontakt so zákazníkom práve prostredníctvom mobilných aplikácií, ktoré majú vyvinuté na mieru. Takéto riešenie je však pre malých podnikateľov alebo živnostníkov príliš nákladné a teda pre nich nerentabilné.

Potrebné je preto myslieť nie len na kvalitné prevedenie z technického hľadiska, ale predovšetkým je potrebné brať do úvahy pohľad ľudí, často laikov, ktorí budú s touto aplikáciou pracovať. Je preto pri vývoji potrebné dbať na výber rôznych technológií, tak ako aj na samotné užívateľské rozhranie so všetkým, čo k nemu patrí.

Prvotne teda bude potrebné analyzovať súčasný trh, čomu sa venuje kapitola 2. Následne sa kapitola 3 zaoberá návrhom riešenia požadovaných funkcií a užívateľského rozhrania. Tomuto návrhu je pomocou spomínaných moderných technológií zostavené pozadie aplikácie, hlavne pomocou technológie *Google Firebase*, čomu je venovaná kapitola 4 spolu so základmi implementácie pre vybranú platformu. V nadväznosti na túto technológiu sú použité aj rôzne iné služby popísané v kapitole 5, spolu s *cloudovou* variantom vlastného podporného servera. Kapitola 6 je venovaná niektorým špecifickým detailom implementácie užívateľského rozhrania.

Po zvládnutí týchto častí je potrebné venovať sa testovaniu implementovaného riešenia, v kapitole 7, ktorá hovorí aj o publikovaní aplikácie a o výsledkoch testovania s pohľadom na budúce možné rozšírenia.

Kapitola 2

Analýza trhu a jeho požiadavky

K zámeru vytvoriť aplikáciu využívajúcu najnovšie technológie je najprv potrebné nájsť oblasť trhu, na ktorý by sa dali tieto technológie aplikovať. Táto kapitola sa teda zaoberá konkretizovaním cieľového trhu, pre ktorý má byť aplikácia navrhnutá ako aj špecifikáciou požiadaviek daného trhu. Téma tejto práce má široký záber. Tento fakt by však mohol v tomto prípade začínajúcej aplikácie spôsobiť stav, keď je snaha o pokrytie čo najväčšieho priestoru trhu, čím by mohlo v systéme vzniknúť veľa druhov požiadaviek na službu a veľa druhov poskytovateľov služieb.

Problém by nastal v prípade, keby množiny poskytovaných služieb a ponúk mali spolu minimálny až žiadny prienik, teda by sa v systéme vyskytovali užívatelia ponúkajúci iné služby ako užívatelia požadujú.

Analýza trhu vyžaduje rôzny prístup. Je potrebné pozorovať, aké služby ľudia často vyhľadávajú, akým službám chýba platforma pre ich ponúkanie alebo aké služby sa ponúkajú spôsobom, ktorý prvotne nie je pre túto činnosť prispôbený.

Pozornosť je sústredená hlavne na rôznych poskytovateľov alebo predajcov, ktorí svoje aktivity sústredili na sociálnych sieťach, hlavne *Facebook*, kde pomocou stránok ponúkali svoje služby. Pomocou nich komunikovali s užívateľmi a poskytovali akýsi katalóg pre potenciálneho zákazníka. Najčastejšie takto pôsobiacimi službami sú rôzni cukrovinkári, cukrári a domáci pekári.

Pre týchto poskytovateľov služieb je *Facebook* často jediná platforma pre prezentovanie svojich výsledkov a kontaktu so zákazníkom. Poskytovatelia nevyužívajú inú formu kontaktu so zákazníkom, ako by mohli byť napríklad webové portfóliá, inzeráty atď.

Z pohľadu zákazníka sa tiež jedná o krkolomné riešenie, keďže zákazník častokrát nevie spoľahlivo nájsť poskytovateľa služby vo svojom okolí. Toto je zapríčinené rozličným spôsobom zaraďovania a nazývania stránok poskytovateľov služieb. Zákazníci teda najčastejšie prichádzajú do kontaktu s takýmito poskytovateľmi len na základe referencií od iných užívateľov sociálnych sietí alebo len jednoducho od svojich známych.

2.1 Analýza užívateľov

Aplikácia by mala slúžiť dvom typom užívateľov a to poskytovateľom služieb a ich potenciálnym zákazníkom. Aplikácia samotná by teda mala poskytovať dve sady služieb pre danú skupinu užívateľov.

Z dotazníkov (otázky s odpoveďami v prílohe A) a rozhovorov s 10 poskytovateľmi služieb, kontaktovaných hlavne prostredníctvom sociálnych sietí, vyplynuli rôzne fakty a to

napríklad fakt, že väčšina z nich má rozdielny prístup k organizovaniu objednávok a ku komunikácií so zákazníkom. Pre komunikáciu so zákazníkmi preferovali telefonickú komunikáciu, komunikáciu cez chat a komunikáciu e-mailom. Ďalšími dôležitými informáciami boli údaje, ktoré chcú poskytovatelia služieb zverejňovať.

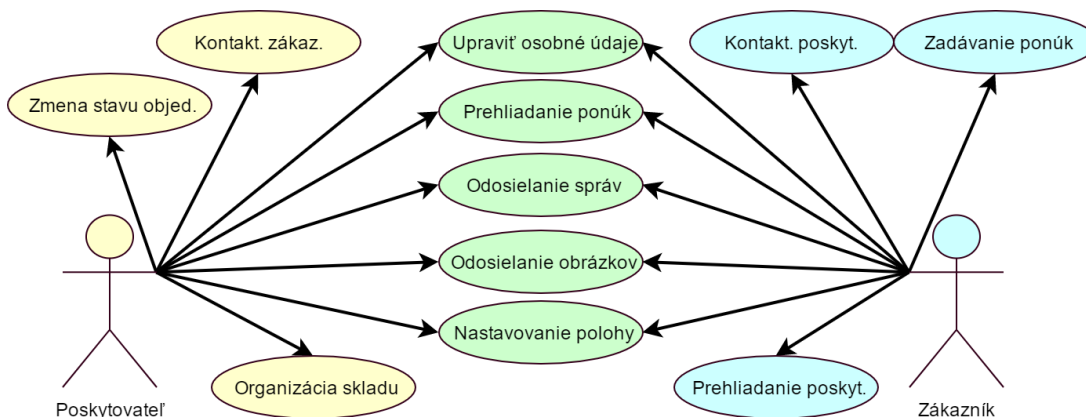
Potenciálny zákazník by mal o poskytovateľovi vedieť:

- Meno/Názov spoločnosti/Prezývka
- Vzdialenosť medzi ním a poskytovateľom
- Telefónne číslo
- E-mail

Poskytovatelia v dotazníkoch uvádzali aj iné možnosti ako odkaz na sociálne siete alebo recenzie od ostatných užívateľov. Tieto možnosti sú považované na základe počtu hlasov za menej prioritné, teda budú brané do úvahy neskôr, týkajúc sa vývoja samotnej aplikácie. Kontaktovaným poskytovateľom bola predstavená aj myšlienka interného chatu v rámci aplikácie, kde by túto možnosť uvítali v podstate všetci oslovení. Poskytovatelia tak isto naznačovali, že pre organizáciu ich objednávok nepoužívajú žiadny špecifický nástroj a aplikácia by mohla v rámci akejsi pridanej hodnoty obsahovať napríklad organizér objednávok, kalkulačku nákladov na výrobu alebo kalkulácie surovín potrebných pre nákup na dané množstvo výrobku.

Na druhej strane sú tu užívatelia, ktorým by mala aplikácia poskytnúť možnosť kontaktovať poskytovateľa rôznymi spôsobmi, zadávať požiadavky poskytovateľom a ukazovať informácie o aktuálnom stave nimi zadanej požiadavky.

Požiadavky užívateľov je potom možné zhrnúť do *diagramu prípadov použitia*¹, ktorý je v zjednodušenej podobe zobrazený na obrázku 2.1.



Obr. 2.1: Zjednodušený diagram prípadov použitia vyplývajúci z požiadaviek užívateľov typu *poskytovateľ* a *zákazník*.

¹use case diagram

2.2 Analýza trhu s mobilnými aplikáciami

Ďalším aspektom analýzy a prípravy na návrh aplikácie je určenie vhodnej platformy. V tomto prípade je vnímaný ako primárny trh pre nasadenie aplikácie Česká republika, prípadne Slovensko.

V takomto prípade pomôže prieskum trhu² ohľadom mobilných operačných systémov, vykonávaný českými mobilnými operátormi. Tento a podobné prieskumy slúžia ako vhodné ukazovatele toho, že operačný systém *Android* v týchto regionálnych podmienkach obsadzuje majoritu trhu a teda je predpoklad, že by mal byť najvhodnejším riešením pre trh Česka a Slovenska.

V segmente mobilných aplikácií existuje snáď len jedno priame konkurenčné riešenie zameriavajúce sa na malých podnikateľov, amatérskych výrobcov alebo remeselníkov.

Rukie

Táto aplikácia slúži na podobný účel ako navrhovaná aplikácia, no s tým rozdielom, že je cieleňá prednostne na absolventov stredných odborných učilíšť, keďže vznikla v rámci projektu podpory remesiel.

Aplikácia naplňa podobnú funkcionálnosť ako bola v tejto práci určená, ale v obmedzenom prevedení a to konkrétne poskytuje chat (v tomto prípade) s remeselníkom a pridávanie dopytu.

Neponúka žiadne vedľajšie funkcie podporujúce prácu poskytovateľa služby, pravdepodobne pre široký záber trhu poskytovateľov služieb.

Vyzdvihnúť však treba hlavne dizajn, ktorý je účelný. Užívateľ presne vie, čo má v danom kroku očakávať (ukážky na obrázku 2.2).

Foursquare

Konkurenciou, aj keď nepriamou, môže byť aj táto aplikácia. Hlavná myšlienka nástroja na vyhľadávanie služieb je však totožná. Aplikácia sa sústreďí hlavne na pomoc pri hľadaní stravovacích zariadení v okolí. Jej hlavná cieľová skupina sú turisti alebo ľudia na pracovných cestách.

Opäť je potrebné vyzdvihnúť dizajn aplikácie, ktorý je o niečo jednoduchší ako v prípade aplikácie *Rukie*, ale užívateľ vždy presne vie, čo prvky rozhrania znamenajú a čo je nimi možné dosiahnuť (ukážky na obrázku 2.3).

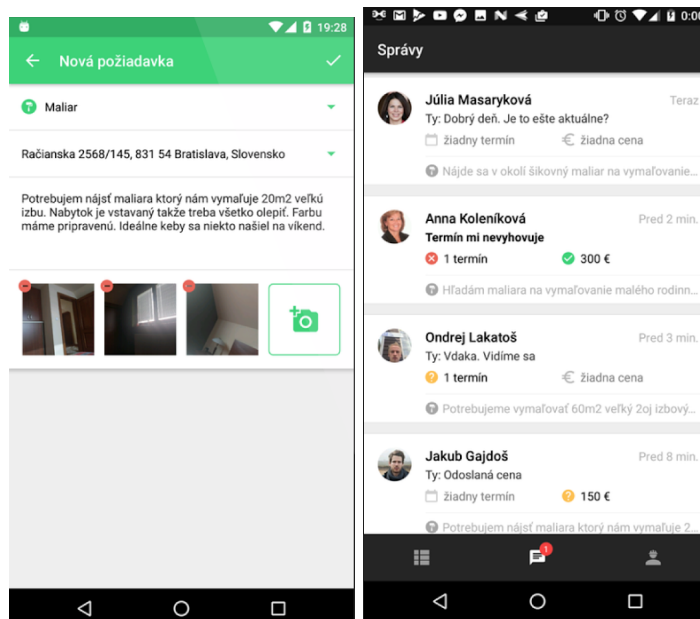
Ďalšie riešenia

Za ďalších konkurentov, ako vyplýva z prieskumu medzi samotnými poskytovateľmi, je možné považovať aj napríklad *Microsoft Excel*³ alebo praobyčajné pero a papier.

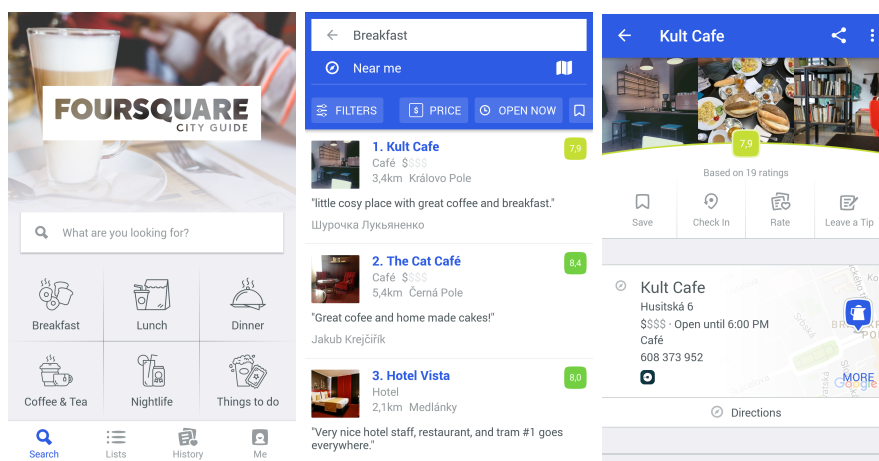
Poskytovatelia v takýchto prípadoch volia jednoduchú formu zápisu. Jedná sa väčšinou o poznamenávanie osobných údajov, dátumov a iných pre nich podstatných informácií. V prípade *Microsoft Excel* sa nejedná o využívanie žiadnych pokročilejších funkcií tohto nástroja.

²<https://mobilenet.cz/clanky/pravda-o-podilu-ios-a-windows-phone-v-cesku-statistiky-19696>

³<https://products.office.com/sk-sk/excel>



Obr. 2.2: Ukážky obrazoviek z aplikácie *Rukie*, obrázok vľavo znázorňuje obrazovku pridávania objednávok so slovným a obrázkovým popisom. Druhá obrazovka ukazuje prehľad požiadaviek v okolí poskytovateľa.



Obr. 2.3: Ukážky obrazoviek z aplikácie *Foursquare*, na obrázku vľavo je úvodná obrazovka poskytujúca možnosti rýchleho vyhľadávania. Obrázok v strede zobrazuje konkrétne výsledky vyhľadávania a obrazovka vpravo zobrazuje vybrané stravovacie zariadenie.

Kapitola 3

Návrh aplikácie

Táto kapitola bude hovoriť o návrhu, ktorý je vytvorený na základe požiadaviek trhu z predchádzajúcej kapitoly 2. Následne budú tieto požiadavky využité pre implementáciu opísanú v kapitole 6.

Pre lepšiu názornosť a jednoduchosť následnej implementácie sú vytvorené *mockupy*¹ pomocou nástroja *Balsamiq Mockups*², dostupného pre systémy *Windows*. Pri ich vytváraní bola snaha dbať na súčasný trend *material design*³. Tieto *mockupy* sú logicky zoradené podľa poradia a funkčného zaradenia a názorne je naznačené, aká akcia sa vykoná kliknutím na ktorý prvok. Ukážky návrhov na Obrázku 3.1.



Obr. 3.1: Ukážka niektorých z návrhov užívateľského rozhrania. Tieto návrhy slúžia ako návrh hlavných prvkov užívateľského rozhrania pre následnú implementáciu. Viac návrhov v prílohe B a doplnené pre verziu 1.2 v prílohe C.

¹návrh alebo model dizajnu určený na demonštráciu, schvaľovanie, propagáciu alebo na iné účely

²<https://balsamiq.com/download/>

³<https://material.io/guidelines/>

3.1 Všeobecný postup užívateľa

Užívateľom, bez ohľadu na ich typ, by sa po prvom spustení aplikácie mala zobrazíť obrazovka s výberom spôsobu prihlásenia a prípadnou možnosťou vytvorenia nového užívateľského účtu.

Ak užívateľ nemá vytvorený ani jeden druh účtu z ponúkaných, klikne na tlačidlo, ktoré ho preniesie na novú obrazovku, kde vyplní povinné polia ako sú napríklad meno, e-mail, vyplní kolónky na heslo a overenie hesla s patričnou dĺžkou hesla a klikne na tlačidlo *Registrovať*.

Po registrovaní alebo prihlásení bude užívateľom zobrazená úvodná obrazovka, kde sa vyskytujú rôzne údaje, ktoré majú pre daného užívateľa prioritný význam.

Pre užívateľa typu *cukrár* sú zobrazené položky z tzv. skladu surovín, ktorých hodnota sa nachádza pod nastavenou, kritickou hranicou a druhou kategóriou zobrazovaných dát sú tzv. čakajúce objednávky. Užívateľ typu *zákazník* vidí na úvodnej obrazovke len svoje aktuálne čakajúce objednávky.

Bočný panel

Na hornej lište si užívateľ bude môcť vybrať ponuku bočného panelu⁴ v ktorej sa budú nachádzať položky z ktorých sú pre oba druhy užívateľov zhodné položky *Môj profil*, *Chat*, položka pre zobrazenie pomoci a domovské tlačidlo. V bočnom menu sa v hornej časti budú nachádzať informácie ako meno a e-mail aktuálne prihláseného užívateľa.

Môj profil

Po výbere položky *Môj profil* sa zmení fragment aktivity a zobrazia sa textové polia, ktoré je možné zmeniť. Navyše sa zobrazí aj prepínač, ktorým užívateľ bude môcť určovať, či je potenciálny zákazník alebo sa jedná o užívateľa, ktorý chce ponúkať služby.

Na záver zobrazenia sa bude nachádzať tlačidlo, po ktorého aktivovaní sa užívateľovi zobrazí aktivita, kde na mape bude môcť zaznačiť svoju polohu, na základe ktorej sa bude potom vypočítavať vzdialenosť k ostatným užívateľom. Poskytovatelia budú mať možnosť zadania viacnásobnej polohy.

Chat

Ďalšou položkou z bočného menu je položka *Chat*. Po jej aktivovaní sa zmení fragment zobrazovanej aktivity na zoznam všetkých aktívnych chatov, teda chatov s užívateľmi, s ktorými si už užívateľ v rámci aplikácie písal.

Po kliknutí na konkrétny chat sa užívateľovi zobrazí aktivita súkromného chatu. V tejto aktivite bude vidieť v akýchsi bublinách rôznej farby pre prichádzajúce a odchádzajúce správy užívateľov danej konverzácie.

V spodnej časti je užívateľovi zobrazená trojica elementov pre vytvorenie textovej a obrazovej správy a tlačidlo pre odoslanie textovej správy. Pri kliknutí na tlačidlo vytvorenia obrazovej správy je užívateľovi ponúknuté dialógové okno pre výber fotografie z fotogalérie alebo možnosť pre nasnímanie fotografie.

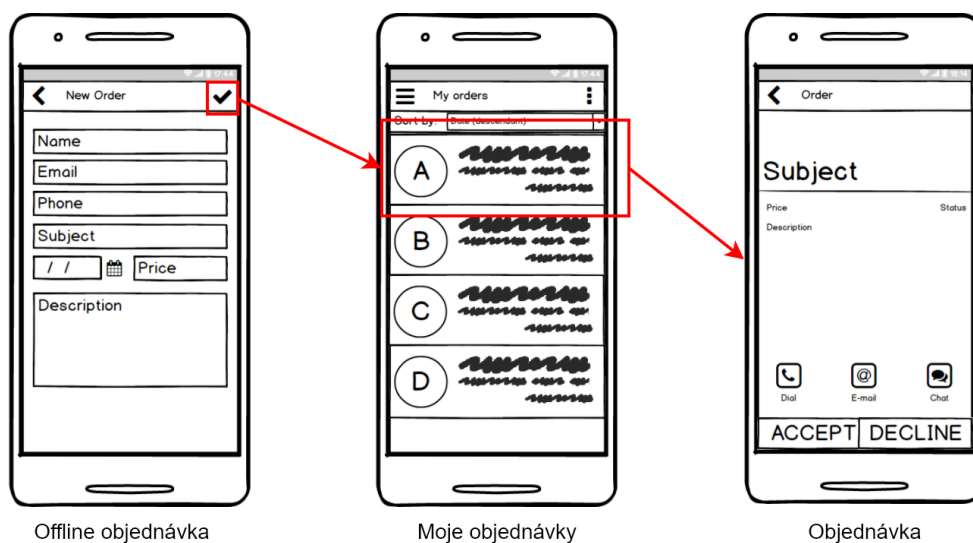
⁴angl. sidebar menu

Bočné menu

Po kliknutí na ikonu bočného menu je užívateľovi zobrazená ponuka položiek pre zobrazenie informácií o aplikácii a položka pre odhlásenie. Tieto položky sú oddelené od ostatných funkcií prihláseného užívateľa, keďže nesúvisia priamo s ostatnými aktivitami, ktoré môže užívateľ vykonávať.

3.2 Postup užívateľa typu *cukrár*

Užívateľia, ktorí si či už priamo v profile alebo pri registrácii vybrali, že budú v rámci aplikácie figurovať ako poskytovateľ služieb, budú mať prístup k iným funkciám ako užívateľia typu *zákazník*. Ukážky návrhov sú tiež na obrázku 3.2.



Obr. 3.2: Ukážky návrhov pre užívateľa typu *cukrár*. Užívateľ si môže do systému pridávať vlastné *offline objednávky*, ktoré sú zobrazené medzi všetkými objednávkami. Kliknutím na konkrétnu objednávku je užívateľovi zobrazená obrazovka s informáciami o nej. Tu môže tiež reagovať na objednávku zmenou statusu, prípadne môže kontaktovať zákazníka.

Offline objednávka

Poskytovatelia služieb budú mať možnosť pridávať do systému objednávky aj vtedy, ak im nebola pridaná iným užívateľom aplikácie, teda napríklad objednávky o ktorých sa dohodol so zákazníkom iným spôsobom ako v rámci aplikácie.

Takúto aktivitu pridávania *offline objednávky* môžu začať po kliknutí na tzv. plávajúce tlačidlo⁵ v spodnej časti obrazovky vpravo. Toto tlačidlo je zobrazené v rámci všetkých fragmentov hlavnej aktivity.

Po kliknutí na tlačidlo mu je zobrazený formulár, v ktorom sa nachádzajú položky pre vytvorenie objednávky ako meno objednávateľa, dohodnutá cena, dátum, popis objednávky a iné voliteľné položky ako e-mail alebo telefónne číslo.

⁵angl. floating button

Objednávky

V bočnom paneli má užívateľ tohto typu položku *Objednávky*, po ktorej výbere sa užívateľovi zobrazia všetky minulé a súčasné objednávky ale aj objednávky, ktoré čakajú na jeho schválenie. Poskytovateľ služieb si bude vedieť tento zoznam objednávok zoradovať podľa viacerých atribútov.

Po kliknutí na konkrétnu objednávku sa zobrazí objednávka s patričnými informáciami o nej. Užívateľ bude navyše môcť meniť stav objednávky tlačidlami na spodku obrazovky.

Pri novo vytvorenej objednávke bude mať na výber z dvoch možností pre akceptovanie alebo pre odmietnutie objednávky. Pri akceptovaní objednávky si bude môcť užívateľ importovať informácie o objednávke do interného kalendára zariadenia. Užívateľ bude mať možnosť na obrazovke objednávky kontaktovať užívateľa viacerými možnosťami. Po kliknutí na jednu z možností *telefonát* a *e-mail* bude spustená východisková systémová aplikácia pre telefonát alebo tvorbu e-mailu. Po kliknutí na tlačidlo chatu sa užívateľ dostane do chatového rozhrania spomínaného v predchádzajúcej sekcii [3.1](#).

Sklad surovín

Pre verziu aplikácie *1.2* je navrhnutá funkcia tzv. skladu surovín. Do takéhoto sa dostane kliknutím na danú položku v bočnom menu. Užívateľovi sa následne zobrazí zoznam aktuálne vedených surovín a ich aktuálny stav spolu s tlačidlami na rýchlu úpravu stavu suroviny o preddefinovanú hodnotu.

Jednotlivé záznamy bude možné samostatne meniť a upravovať ich nastavenia kritických hodnôt alebo hodnôt na ktoré budú reagovať tlačidlá pre rýchlu zmenu stavu surovín. Rovnako je možné zadať jednotky, v akých sa daná surovina eviduje.

3.3 Postup užívateľa typu *zákazník*

Užívateľ typu *zákazník* pochopiteľne nebude mať k dispozícii také funkcie ako má *cukrár*. V jeho kompetencii bude prezeranie *cukrárov*, zadávanie a kontrolovanie svojich objednávok. Ukážky návrhov na obrázku [3.3](#).

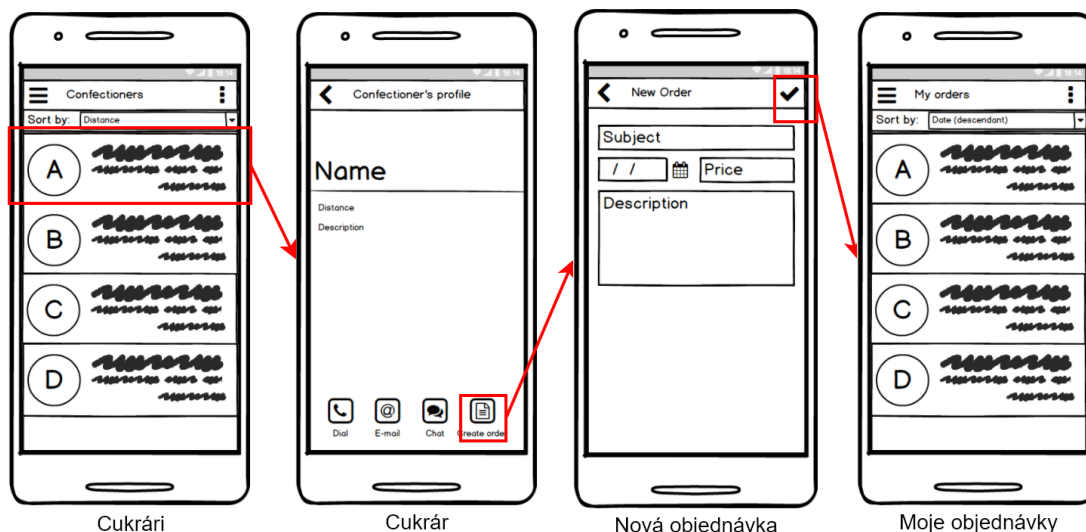
Cukrári

Po vybratí položky *cukrári* z bočného panelu sa užívateľovi zobrazí zoznam poskytovateľov služieb, ktorých je možné zoradovať podľa viacerých kritérií. Po výbere jedného zo zoznamu sa užívateľovi otvorí profil daného užívateľa, kde sa o ňom dozvie základné informácie.

Navyše bude môcť užívateľ ním vybraného užívateľa typu *cukrár* kontaktovať podobným spôsobom ako tomu je navrhnuté pri užívateľovi typu *cukrár* teda e-mailom, telefonicky alebo chatom v rámci aplikácie.

Užívateľ typu *zákazník* bude mať navyše kompetenciu pridávať požiadavky užívateľom typu *cukrár* a to kliknutím na danú ikonu v profile *cukrára*. Po takejto akcii sa užívateľovi zobrazí formulár s viacerými povinnými polami o informáciách k objednávke.

Po ich úspešnom vyplnení bude objednávka zaradená do systému a užívateľ bude môcť takúto objednávku vidieť v zozname svojich existujúcich objednávok.



Obr. 3.3: Ukážky návrhov pre užívateľa typu *zákazník*. Tento typ užívateľa má možnosť preliadať *cukrárov*, následne ich kontaktovať, prípadne vytvorí objednávku. Túto objednávku potom vidí medzi svojimi objednávkami.

Objednávky

Túto možnosť bude užívateľ môcť vyvolať kliknutím opäť na jednu z položiek bočného panelu. Po kliknutí sa užívateľovi zobrazí zoznam jeho objednávok aj so základnými informáciami o nej.

Po rozkliknutí vybranej objednávky sa užívateľ dozvie všetky informácie o nej a bude sa môcť podobne ako v predchádzajúcej časti, pomocou ikon kontaktovať s užívateľom, ktorý obdržal jeho objednávku.

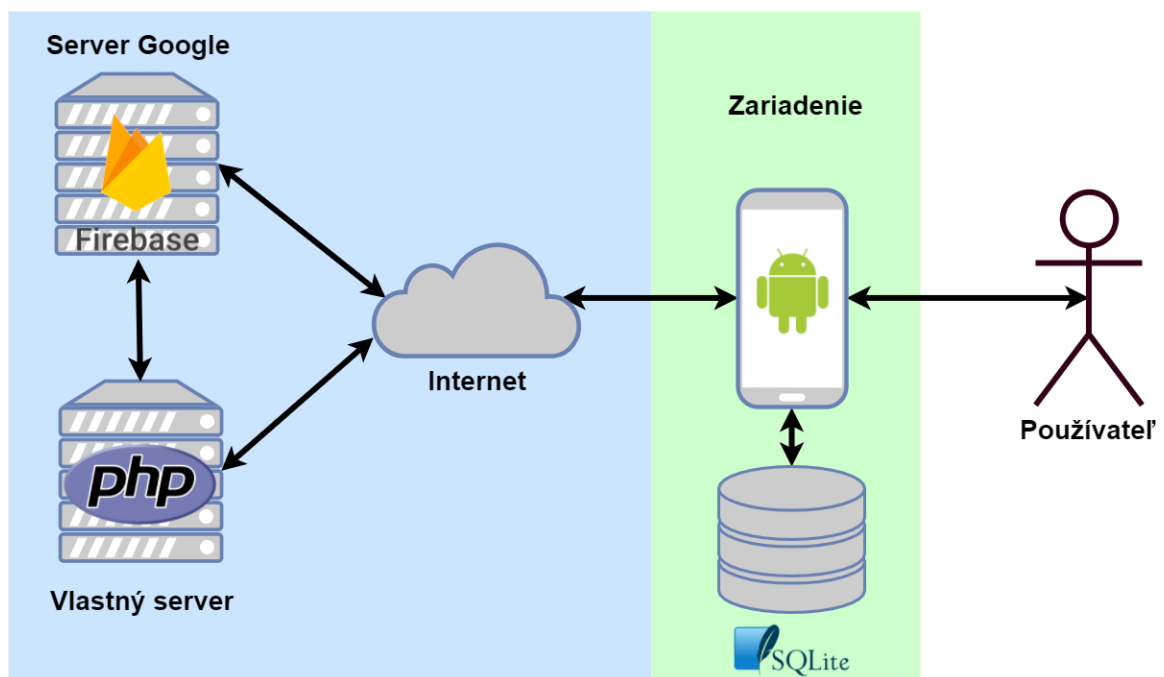
3.4 Architektúra aplikácie

Základom architektúry, ako bolo spomínané, je aplikácia pre platformu *Android*, s ktorou interaguje samotný užívateľ.

Táto aplikácia pracuje s dátami v *offline* databáze *SQLite*, bližšie popísanej v kapitole 5.2. Dáta, ku ktorým je potrebný prístup aj ostatných užívateľov sú ukladané na servery *Google Firebase*. Tieto servery ponúkajú aj rôzne iné služby z ktorých sú niektoré využívané. Bližšie je týmto službám venovaná kapitola 4.4.

Niektoré doplnkové služby je potrebné riešiť pomocou vlastného servera, na ktorý sa má zariadenie možnosť pripojiť cez internet. Tento server vytvára akýsi medzičlánok medzi zariadením a serverom *Firebase*, bližšie popísané v kapitole 5.1.

Názorne si je možné celú architektúru aplikácie predstaviť ako je na obrázku 3.4.



Obr. 3.4: Znázornenie komunikácie aplikácie a jednotlivých častí. Používateľ ovláda zariadenie s operačným systémom *Android*, ktoré v sebe uchováva *SQLite* databázu. Toto zariadenie je pripojené na internet, z kade má prístup na server *Firebase* a vlastný server. Tieto servery v prípade posielania cloudových správ spolu spolupracujú.

Kapitola 4

Základy implementácie pre operačný systém Android a mobilná platforma *Google Firebase*

Pred samotnou implementáciou navrhnutého riešenia je potrebné spoznať cieľovú platformu *Android* a určiť potrebné technológie k dosiahnutiu požadovanej funkcionality.

4.1 Operačný systém *Android*

Tento operačný systém, vyvíjaný spoločnosťou *Google*, je založený na jadre *Linuxu*, jedná sa teda o *open source* projekt.

Android sa od verzie 1.5 začal označovať aj pomocou rôznych sladkých pochutín vyplývajúcich z písmen abecedy po poradi, v tomto prípade to bol *Cupcake* [4]. Dnes sa nachádza na najnovšej verzii 7.0 - 7.1.2, označovanej aj ako *Nougat*. Mimo iného sa spomínaný operačný systém prestal aplikovať len v oblasti mobilných telefónov ale jeho pôsobnosť sa rozšírila do oblasti tabletov, inteligentných hodínok, televízorov a mnohých iných druhov zariadení s rozličnou veľkosťou a určením.

Momentálna verzia *API* sa nachádza vo verzii 25 no aplikáciu nemôžeme zacieliť len na túto - najnovšiu verziu. Kvôli tejto skutočnosti bude používaná verzia *API 16*, ktorá je dostupná od verzie operačného systému *Android 4.1.x*. Týmto by sa podľa zdrojov¹ malo pokryť približne 97% používaných zariadení.

4.2 Vývojové prostredie *Android Studio*

Pre vývoj aplikácií pre vybraný operačný systém je možné si vybrať z dvoch najčastejšie používaných vývojových prostredí *Eclipse* a *Android Studio*. V tomto prípade je vybrané práve *Android Studio*.

Jedná sa o oficiálne vývojové prostredie založené na prostredí *IntelliJ IDEA*² podporujúce viacero operačných systémov pre inštaláciu - v tomto prípade operačný systém *Windows 10*.

¹<https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/>

²JetBrains IntelliJ IDEA - <https://www.jetbrains.com/idea/>

Samotné vývojové prostredie ponúka množstvo pre vývoj dôležitých nástrojov ako napríklad sú:

- *Gradle* - nástroj pre automatické zostavovanie programov
- *Android Virtual Device manager* (alebo skratkou *AVD manager*) - nástroj pre správu a používanie emulátorov zariadení s operačným systémom *Android*
- *Android software development kit* (alebo skratkou *Android SDK*) - súbor vývojových nástrojov ako napr.: knižnice, debugger, dokumentácia atď.

Naviac bol doinštalovaný plugin³ pre verzovací systém *git* pre službu *Bitbucket*⁴. Týmto nástrojom je možné spravovať zdrojové súbory a ich verzie.

4.3 Dôležité súbory

Pred samotným vývojom bolo potrebné spoznať aké možnosti poskytuje vývoj pre *Android* a čo poskytujú rôzne rozhrania.

Súbor manifestu aplikácie

Takýto súbor⁵ obsahuje podstatné informácie o aplikácii pre operačný systém. Súbor s názvom `AndroidManifest.xml` sa nachádza v koreňovom adresári.

Do neho sú zapisované povolenia k rôznym zdrojom operačného systému, čo je napríklad prístup k pamäti zariadenia, kalendáru operačného systému atď.⁶ Takisto sa tu vyskytujú informácie o jednotlivých aktivitách, základné informácie o tom ako majú byť zobrazované atď.

Súbor *Gradle*

Pre správu vydaní aplikácie je potrebný práve takýto súbor. V ňom je možné upravovať nastavenia vydania, ako aj nastavenie minimálnej verzie podporovaného *API* alebo aj číselné označenie verzie. Podstatnou časťou tohto súboru sú závislosti, ktoré sú kompilované do vydaní. Sem sú pridávané napríklad závislosti rôznych podporných knižníc či rozhraní.

Zdrojové súbory

Zdrojové súbory je možné rozdeliť do dvoch kategórií. Kategória súborov s kódom v jazyku *Java*, slúžiace pre programové ovládanie fragmentov, aktivít alebo pre vytvorenie objektov databázy, podporné triedy atď. Takéto súbory sa nachádzajú v zložke `src/`.

Druhou kategóriou sú súbory v zložke `res/`. Tieto súbory je možné rozdeliť do viacerých kategórií, ktorým patrí samostatná zložka. Podstatnou kategóriou sú napríklad *XML* súbory zložky `layout` ktoré definujú užívateľské rozhrania. Tieto súbory je možné editovať aj pomocou grafického editora v prostredí *Android Studio*.

Ďalšie často využívané súbory sa nachádzajú v zložke `values`. Nachádzajú sa tu napríklad definície používaných farieb alebo jazykových reťazcov. Pre rôzne jazykové lokalizácie

³ plugin - prídavný program rozširujúci funkcie iného programu

⁴ <https://bitbucket.org>

⁵ <https://developer.android.com/guide/topics/manifest/manifest-intro.html>

⁶ toto však od verzie *API 23* nie je dostatočné

je potrebné vytvoriť zložku `values` s patričnou príponou, napríklad pre českú lokalizáciu je to zložka `values-cs`.

4.4 Mobilná platforma *Google Firebase*

Z analýzy riešenia mobilnej aplikácie pre tento účel vyplýva, že bude potrebné zabezpečiť, aby užívatelia medzi sebou vedeli interagovať (prostredníctvom internetu). Ďalej bude potreba zabezpečiť ukladanie dát a rôzne iné, s týmto súvisiace činnosti.

Kvôli týmto požiadavkám je vhodné využiť platformu od spoločnosti *Google* pre vývoj mobilných aplikácií, integrujúcu množstvo potrebných služieb. Služba *Firebase* poskytuje svoje služby nielen pre aplikácie vyvíjané pre systém *Android*, ale aj pre *iOS* a pre aplikácie navrhnuté pre web, čo je možné brať ako pozitívum k prípadnému budúcemu rozširovaniu navrhnutého systému[3]. Najnovšie bola pridaná podpora služieb *Firebase* aj pre programy v *C++* alebo pre aplikácie v hernom engine *Unity*⁷.

Služby *Firebase* sú dostupné pre rôzne platformy v rôznych podobách a v rôznom množstve. Pre vybranú platformu *Android* sú dostupné služby rôznych kategórií:

- Analytické
- Vývojárske
- Podpora rastu
- Zárobkové

Primárne použité v aplikácii sú nástroje pre autentifikáciu 4.4.1, reálno-časová databáza 4.4.2, úložisko pre dáta 4.4.3 a služby tzv. cloudových správ 4.4.4. Najnižšia požadovaná verzia operačného systému *Android* je verzia 4.0, čo vyhovuje návrhu aplikácie.

Vo využívanej bezplatnej verzii služieb, tzv. *Spark*⁸, určenej pre začínajúce aplikácie sú isté obmedzenia, z ktorých najpodstatnejšie sú:

- maximálne 100 simultánne pripojených zariadení
- maximálne 1GB dát v reálno-časovej databázy
- maximálne 5GB dát v dátovom úložisku

Kompletný zoznam obmedzení a podmienok je uvedený na stránkach poskytovateľa⁹.

Prechod na inú verziu je možné riešiť plynulo, bez výraznejšieho zásahu do kódu aplikácie a bez vedomosti užívateľov.

Inicializácia

Pred samotným používaním tejto mobilnej služby je potrebné službu inicializovať.

Po vytvorení užívateľského účtu je potrebné pridať vytvorený projekt a vyplniť patričné informácie o ňom. Následne je vygenerovaný konfiguračný súbor, ktorý je vložený k zdrojovým súborom do zložky `app/`. Po tomto kroku je potrebné upraviť *Gradle* súbory a to nasledovne.

Pre súbor `build.gradle` v koreňovom adresári:

⁷<https://unity3d.com/>

⁸<https://firebase.google.com/pricing/>

⁹<https://firebase.google.com/pricing/>

```
buildscript {
    dependencies {
        // Pridat nasledovny riadok
        classpath 'com.google.gms:google-services:3.0.0'
    }
}
```

Následne pre súbor `build.gradle` v zložke `app/build.gradle`:

```
dependencies {
    // Pridat nasledovny riadok + ine zavislosti ktore chceme pridat
    compile 'com.google.firebase:firebase-core:10.0.1'
}
// Na koniec suboru je potrebne pridat tento riadok
apply plugin: 'com.google.gms.google-services'
```

Po tejto relatívne jednoduchej inicializácii je možné začať používať požadované súčasti *Firebase*.

4.4.1 Autentifikácia

Služba *Firebase* ponúka hneď niekoľko možností autentifikácie, konkrétne pomocou účtu *Google*, *Facebook*, *Twitter*, *GitHub* a pomocou vlastnej registrácie užívateľa.

Na základe prvotného prieskumu medzi potenciálnymi užívateľmi sú vybrané dve možnosti autentifikácie a to konkrétne pomocou účtu *Facebook* a pomocou vlastnej registrácie. Neskôr však bude možné relatívne jednoducho pridať ďalšiu požadovanú možnosť.

Účet *Facebook*

K používaniu účtov sociálnej siete *Facebook* je najprv nutné pridať potrebnú závislosť do *Gradle* súboru aplikácie.

Následne je nutné vytvoriť si vývojársky účet na stránkach *Facebooku*. Po jeho úspešnom vytvorení a inicializovaní sú podľa inštrukcií pridané do *Gradle* súboru aplikácie potrebné závislosti.

Vlastná registrácia

Ďalšia možnosť, ktorá integrovaná je do aplikácie, je možnosť autentifikácie pomocou vlastného účtu.

Na úvod, ako je pravidlom, je potrebné pridať závislosť požadovanej funkcionality do *Gradle* súboru aplikácie. Tu sa však zhodnosť s inými riešeniami rozchádza, keďže je potrebné riešiť ako prihlasovanie do aplikácie, tak aj registráciu nových užívateľov.

Overovanie *e-mailov*

Táto funkcia je opäť z veľkej časti v kompetencii služieb *Firebase*, ktoré poskytujú možnosť odosielania overovacích *e-mailov*. Formát týchto správ je možné zmeniť vo webovom rozhraní služieb¹⁰.

¹⁰<https://console.firebase.google.com/>

Obnovovanie hesiel

Obnovovanie hesiel užívateľov, ktorí sa z nejakých dôvodov nevedia prihlásiť do aplikácie prebieha podobne ako overovanie *e-mailov*. To znamená že z rozhrania aplikácie je odoslaná požiadavka na servery *Firebase* pre odoslanie overovacieho *e-mailu* na zadanú adresu.

Formát obnovovacích správ je opäť možné zmeniť cez webové rozhranie služieb *Firebase*.

Využitie

Pre určenie validných užívateľov, teda užívateľov, ktorým bude sprístupnená plná funkcionality, je použitá možnosť autentifikácie¹¹.

Rozhranie *Firebase* po úspešnom prihlásení vracia údaje o autentifikovanom užívateľovi, ako sú meno, e-mail a unikátny identifikátor. Tieto údaje sú pri prvom prihlásení vkladané do databázy. V kombinácii s *Facebook API* je možné získať napríklad aj adresu profilového obrázka (viď ukážka 4.1). Aplikácia následne pracuje s dátami v databáze.

Ukážka 4.1: Zdrojový kód, ktorý spracúva prístupový *token* získaný prihlásením cez *Facebook API*. Údaje sú použité pre zostavenie *URL* adresy profilového obrázka.

```
private void handleFacebookAccessToken(final AccessToken token) {
    AuthCredential credential =
        FacebookAuthProvider.getCredential(token.getToken());
    mAuth.signInWithCredential(credential)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (!task.isSuccessful()) {
                    Log.w(TAG, "signInWithCredential", task.getException());
                    Toast.makeText(LoginActivity.this, "Authentication
                        failed.", Toast.LENGTH_SHORT).show();
                } else {
                    updatedPhotoUrl = "https://graph.facebook.com/" +
                        token.getUserId() + "/picture?type=large";
                }
            }
        });
}
```

Overovanie *e-mailov* bolo potrebné zaviesť po istej testovacej iterácii, ktorá ukázala, že bolo možné registrovať užívateľov s vymyslenými adresami a nebolo možné ich nijako filtrovať.

4.4.2 Reálno-časová databáza

Služba *Firebase* ponúka nevšedné riešenie týkajúce sa správy dát. Na rozdiel od väčšinou používaných *SQL*¹² databáz sa jedná o *cloudovú*¹³ databázu so záznamami databázy vo formáte *JSON*.

¹¹mechanizmus identifikácie a overenia identity

¹²dopytovací jazyk pre relačné databázy

¹³služba alebo program uložená na serveroch na internete

Reálno-časovosť

Namiesto klasických *HTTP požiadaviek*, kde sa klienti serveru starajú o aktuálnosť dát ako v databáze, tak dát ich samotných, je využívaný iný mechanizmus.

Tento mechanizmus spočíva v princípe, že v momente, ako sa zmenia dáta databázy sú touto skutočnosťou oboznámené všetky pripojené zariadenia a môžu vykonať požadovanú činnosť. Iným spôsobom je čítanie dát jednorazovo, a teda ak nastane zmena v danom uzle, tak v aplikácii sa tieto zmeny neprejavia. Takéto správanie je vhodné napríklad v prípade, že dáta budú statické.

Ukážka 4.2: Zdrojový kód jednorazovo získavajúci dáta objektu o používateľovi (*sender*) v uzly *users*. V prípade priebehu bez chyby budú dáta (v *datasnapshote*) ďalej spracované. Napríklad sú prevedené na konkrétny objekt typu *User*. Podobným spôsobom sú spracované aj dáta v prípade trvalého pripojenia na uzol databázy namiesto jednorazového.

```
mDatabase.child("users").child(sender).addListenerForSingleValueEvent(new
    ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        doSomeStuff(dataSnapshot);
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        handleError(databaseError);
    }
});
```

Formát JSON

Ak sa hovorí o dátovom formáte *JSON*¹⁴ alebo neskrátene *JavaScript Object Notation*, jedná sa o formát zapisovaných dát vo forme, ktorá je čitateľná pre človeka.

Tento formát bol priamo navrhnutý pre servery operujúce v reálnom čase a snaží sa o nahradenie iného masívne používaného formátu dát *XML*¹⁵. Formát dát v databáze potom môže vyzeráť ako v nasledujúcej ukážke 4.3.

Je možné si všimnúť, že dáta sa do seba postupne vnárajú, teda tzv. uzol *users* obsahuje ďalší uzol obsahujúci ďalšie uzly. Takto je možné do seba vnárať rôzne štruktúry, ktorých vytváranie dnes už často podporujú rôzne programovacie jazyky.

V prípade využitia v aplikácii sa jedná o transformáciu objektov programu priamo na formát *JSON*. O túto transformáciu sa starajú knižnice *Google Firebase*.

¹⁴<https://en.wikipedia.org/wiki/JSON>

¹⁵v preklade rozšíriteľný značkovací jazyk - <https://en.wikipedia.org/wiki/XML>

Ukážka 4.3: Formát JSON, hlavný uzol v tomto prípade je uzol *users*, do ktorého je vnorený uzol *userID* s poduzlami *name* a *email* s príslušnými hodnotami.

```
1 {
2   "users" : {
3     "userID" : {
4       "name" : "valueName",
5       "email": "valueEmail"
6     }
7   }
8 }
```

V tejto štruktúre sa môžu vyskytovať len také objekty, ktoré je možné reprezentovať vo formáte *JSON*, ktorý podporuje nasledovné dátové typy¹⁶:

- *String*
- *Long*
- *Double*
- *Boolean*
- *Map<String, Object>*
- *List<Object>*

Dáta v režime *offline*

Potrebnou súčasťou práce s databázou je zabezpečiť, aby dáta, ktoré z nejakého dôvodu nie je možné čítať alebo zapisovať v súčasnosti bolo možné dodatočne synchronizovať s databázou.

Takáto situácia tzv. režimu *offline*¹⁷ môže byť zapríčinená rôznymi dôvodmi, ako je napríklad nekvalitné internetové pripojenie alebo problém siete.

Skutočnosť, že došlo k problému s pripojením k serverom, kde sa dáta nachádzajú, sa dá pomocou *Firebase* vyriešiť tým, že v zdrojovom kóde je určené, ktoré dáta majú byť synchronizované aj v čase, keď nastane problém s pripojením. Tieto dáta sú následne po obnovení internetového pripojenia synchronizované s databázou.

Celý princíp zjednodušene funguje tak, že dáta sa pri nahrávaní daným zariadením ukládajú ako do *cloudového* úložiska, tak sa vytvára aj lokálna kópia daných dát v konkrétnom zariadení.

Využitie

Informácie o užívateľoch, ale aj rôzne iné dáta potrebné pre komunikáciu medzi užívateľmi sú ukladané do databázy.

Tieto dáta sú potom podľa vhodnosti užívateľmi načítavané či už priebežne, alebo jednorazovo. Napríklad pri prihlasovaní užívateľov sú dáta čítané jednorazovo, kým pri chatovej aktivite sú užívatelia pripojení na uzol databázy trvalo a dáta chatových správ sú prijímané priebežne.

¹⁶<https://firebase.google.com/docs/database/android/read-and-write>

¹⁷stav absencie internetového pripojenia

Návrh databázy

Objekty aplikácie v reprezentácii vo formáte *JSON* sú ukladané do rôznych uzlov databázy. Užívatelia potom tzv. načítavajú jednotlivým uzlom, čím je im sprístupnený obsah daného uzlu a všetkých vnorených uzlov.

Je pre to potrebné navrhnuť databázu tak, aby sa jednalo o tzv. plochú štruktúru. Výsledná databáza má potom v prípade navrhovanej aplikácie štruktúru aká je naznačená v ukážke 4.4.

Táto štruktúra obsahuje niekoľko základných uzlov a to pre:

- Chat aplikácie - poduzly sú zložené zo zoradených identifikátorov účastníkov chatu. Poduzly potom obsahujú chatové správy označené a zoradené podľa koordinovaného svetového času. Ďalšími poduzlami sú potom napr. identifikátor odosielateľa, telo správy atď.
- Dáta *geofire* - poduzly sú zložené z identifikátoru prideleného danému užívateľovi. Jedná sa o upravenú formu unikátneho identifikačného reťazca užívateľov, prispôbeného o možnosť viacnásobnej uloženej polohy. Poduzly potom obsahujú tzv. *geohash* a doplnkové informácie o polohe užívateľa v podobe presných zemepisných súradníc.
- *Offline* kontakty - poduzly sa skladajú z unikátneho identifikátora a ukladajú informácie o užívateľoch, ktorí nie sú priamo registrovaní v aplikácii, ale poskytovatelia si takéto kontakty uložili do systému pri vytváraní tzv. *offline objednávky*.
- Požiadavky (alebo objednávky) - poduzly sú zložené z unikátneho identifikátora a obsahujú dáta objednávok systému, ako sú napr. identifikátor zadávateľa, identifikátor príjemcu, predmet objednávky atď.
- Používatelia - poduzly obsahujú jednoznačný identifikátor a skladajú sa z informácií o daných užívateľoch. Takéto informácie sú napr. celé meno, e-mailová adresa, telefónne číslo, *URL adresa* k profilovej fotografii atď.
- Notifikácie - poduzly obsahujú dáta pre server 2.0 komunikujúci so serverom *Firebase* pre *cloudové* správy. Tieto poduzly sú po úspešnom vytvorení takejto správy vymazané, čím často zaniká aj uzol pre notifikácie. Je možné o ňom teda hovoriť, že nie je trvalým uzlom databázy.

Ukážka 4.4: Jednoduchá ukážka databázy a jej hlavných uzlov s poduzlami.

```
1 {
2   {
3     "chat" :{
4       "chatID1" :{
5         ...
6       }, ...
7     },
8     "geofire" :{
9       "userID1_1" :{
10        ...
11      }, ...
12    },
13    "offlineContacts" :{
14      "offlineUser-ID1" :{
15        ...
16      }, ...
17    },
18    "requests" :{
19      "requestID" :{
20        ...
21      }, ...
22    },
23    "users" :{
24      "userID1" :{
25        ...
26      }, ...
27    }
28 }
```

Skutočnosť obmedzení dátových typov je potrebné brať do úvahy pri samotnom návrhu objektov v rámci aplikácie (viď obrázok 4.1).

4.4.3 Úložisko

Rozhranie poskytuje možnosť ako nahrávať, tak sťahovať súbory do úložiska s možnosťou sledovania vývoja. Vo využívanej verzii služieb *Firebase 9.6.1* však táto obsahuje nedostatočné monitorovanie vývoja nahrávania či sťahovania súboru. Informuje len o žiadnom alebo plne úspešnom priebehu, nie teda o časti, aká je spracovaná.

Vo využívanej verzii služieb *Firebase* je možnosť pracovať s obmedzeným množstvom dátového priestoru, konkrétne *5GB* dát uložených na servery.

Využitie

Úložisko predstavuje dôležitú súčasť hlavne pre implementáciu posielania fotografií cez chat v aplikácii. Kvôli skutočnosti, že je k dispozícii *5GB* dát, je využitý mechanizmus zdieľania súborov, ktorý uchováva na serveroch len súbory, ktoré neboli prevzaté adresátom. Je možné tomu rozumieť napríklad tak, že používateľ *A* odošle súbor užívateľovi *B* a pokiaľ si užívateľ



Obr. 4.1: Príklad navrhnutého objektu - *užívateľ*

B tento súbor z úložiska nestiahne, bude na ňom tento súbor uchovávaný. Po vytvorení lokálnej kópie tohoto súboru užívateľom *B* sa súbor z úložiska zmaže a je dostupný len lokálne na zariadeniach užívateľov.

4.4.4 Cloudové správy

Pomocou tzv. *cloudových správ Firebase*¹⁸ je možné jednotlivých užívateľov upovedomiť notifikačnými správami.

Správa, ktorá má fungovať ako notifikácia, musí byť doručená vo formáte *JSON* na server *Firebase* vo formáte aký je zobrazený v ukážke 4.5.

Ukážka 4.5: Telo odosielanej správy musí obsahovať: identifikátor zariadenia a aplikácie generovaný službami *Firebase*; čas, po ktorý sa servery *Firebase* pokúšajú notifikáciu doručiť; samotný obsah notifikácie. Obsahom notifikácie je v prípade chatovej správy identifikátor odosielateľa, identifikátor chatu (ktorý potom slúži ako parameter pre spustenie danej chatovej aktivity) a ikona.

```

1 {
2   "to" : "userID",
3   "time_to_live" : "2419200"
4   "notification" : {
5     "title" : "userName",
6     "body" : "messageText",
7     "icon" : "ic_launcher"
8   }
9 }
```

¹⁸jedná sa o odnož častejšie používaného systému *GCM (Google Cloud messaging)*

Využitie

Takéto správy sú v rámci aplikácie posielané v prípade odosielania chatovej správy a vytvorenia novej požiadavky pre užívateľa.

V prípade chatových správ sa potom po kliknutí na danú notifikáciu spúšťa aktivita daného chatu a v prípade požiadavky sa užívateľ dostáva do hlavného rozhrania aplikácie.

Kapitola 5

Ostatné časti architektúry

Okrem služieb *Firebase* sú pre architektúru nemenej potrebné aj časti, ktoré či už podporujú funkcie služieb *Firebase*, alebo majú v rámci aplikácie samostatný význam pre implementáciu navrhnutých funkcií.

5.1 Vlastný server

Ako bolo v prechádzajúcej kapitole 4.4.4 spomínané, pre obsluhu tzv. *cloudových* správ je potrebné hlavne z bezpečnostných dôvodov implementovať vlastný server pre obsluhu správ. Pre prípad tejto aplikácie sa jedná o správy, ktoré užívatelia vnímajú ako notifikácie a sú doručované aj v prípade, že aplikácia nie je spustená.

5.1.1 Verzia 1.0

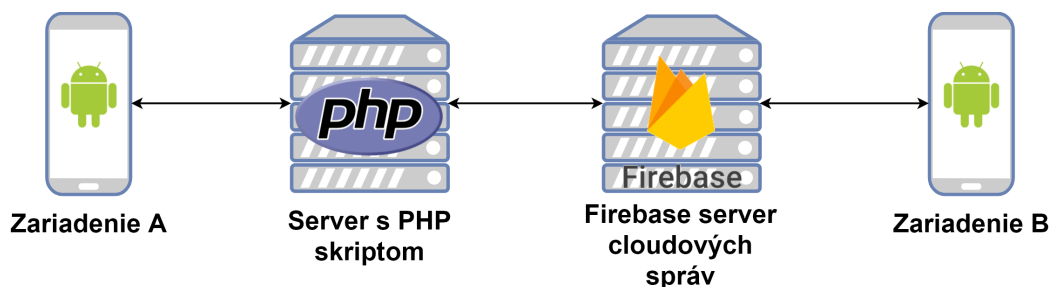
Na serveri pripojenom na internet je umiestnený *PHP skript*, ktorý ako vstupné parametre prijíma:

- Registračný token - reťazec znakov generovaný servermi poskytovateľa slúžiaci k identifikácii zariadenia a aplikácie pre príjem správy
- Nadpis - titulok k notifikácii nesúci meno odosielateľa správy
- Telo - telo správy nesúce napr. obsah chatovej správy v aplikácií

Celú komunikáciu (zobrazenú na obrázku 5.1) začína zariadenie *A*, ktoré pri danej situácii vytvorí *HTTP požiadavku*¹, korešpondujúcu so vstupnými parametrami *PHP skriptu*, na server aplikácie. Ten požiadavku vyhodnotí a vytvorí *HTTP požiadavku*, ktorej telo tvoria dáta vo formáte, aký bol spomenutý v sekcii 4.4.4. Do hlavičky požiadavky sú nastavované parametre typu správy (v tomto prípade *JSON*) a autorizačný kľúč získaný zo služieb *Firebase*, slúžiaci k overeniu pravosti požiadavky. Tento kľúč je uchovávaný neverejne na serveri.

Následne je kontaktovaný server *cloudových* správ *Firebase*, ktorý mu vráti odpoveď o úspešnosti požiadavky a táto odpoveď je následne preposlaná späť na zariadenie odosielateľa. Jeden z parametrov nastavovaných pri odosielaní požiadavky na notifikačnú správu serveru *Firebase* je dĺžka platnosti správy. Tento parameter je nastavený na najvyššiu prípustnú hodnotu, čo zaručuje, že ak je aj zariadenie *B* nedostupné dlhšiu dobu, notifikačná správa bude doručená po jeho pripojení na internet.

¹k čomu je využitá knižnica *OkHttp* - <http://square.github.io/okhttp/>



Obr. 5.1: Znázornenie komunikácie pri posielaní *cloudových* správ. Zariadenie A komunikuje so serverom s *PHP skriptom*, ktorý komunikuje so serverom *Firebase*. Ten vracia odpoveď pre zariadenie A cez server s *PHP skriptom* a doručuje správu zariadeniu B.

5.1.2 Verzia 2.0

Z dôvodu bezpečnostných nedostatkov pôvodného riešenia je potrebné vytvoriť také riešenie, ktoré buď bude podporovať bezpečnú komunikáciu medzi zariadením a serverom, alebo takáto komunikácia nebude priamo vôbec prebiehať.

Na server je nasadená internetová aplikácia *Node.js*, používajúca najrozšírenejší programovací jazyk *Javascript*^[1]. Táto aplikácia nekomunikuje so zariadením *Android* pomocou *HTTP* požiadaviek, ale prostredníctvom databázy serveru *Firebase*.

Komunikácia funguje tak, že zariadenie *Android* vloží do konkrétneho uzla² v základnej štruktúre databázy položku nesúcu všetky potrebné dáta, aké boli posielané na server vo verzii 1.0. Tu zariadenie *Android* svoju činnosť končí.

Na serveri spustená *Node.js* aplikácia načúva na rovnakom uzle databázy, do ktorého zariadenia vkladajú požiadavky. V prípade, že sa v uzle vyskytnú požadované dáta, sú spracované a preposlané na server *Firebase* na doručenia ako notifikácia. Po úspešnom doručení sú dáta o notifikácii z databázy vlastným serverom vymazané. Hlavná časť programu vyzerá ako v ukážke 5.1.

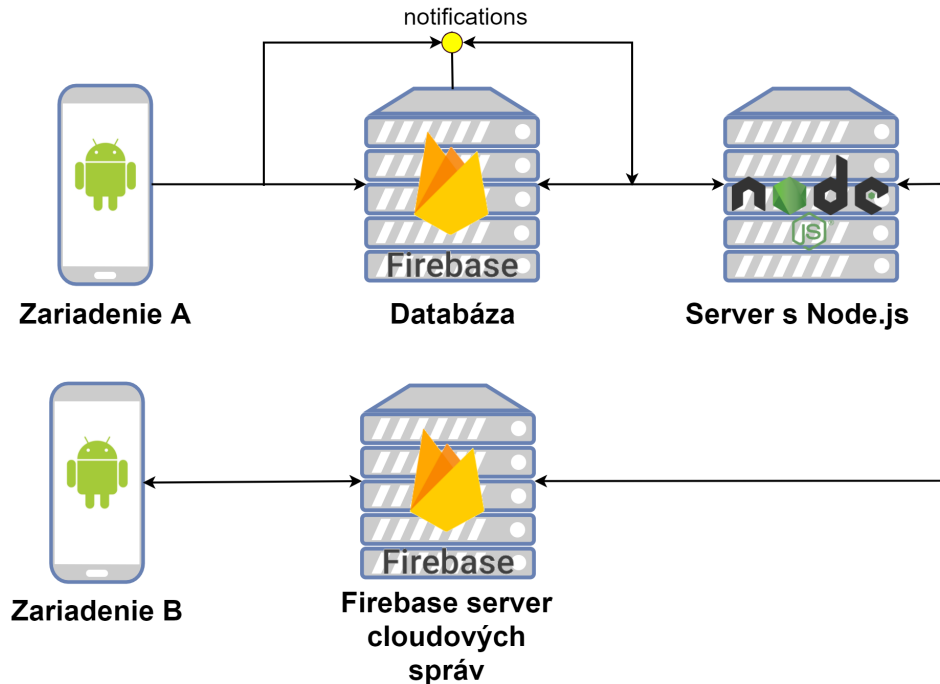
Ukážka 5.1: Časť zdrojového kódu na servery *Node.js*. Táto aplikácia načúva na uzly *notifications* do ktorého zapisujú zariadenia dáta. Server potom tieto dáta ďalej spracuje a prepošle na servery *FCM*.

```

1  var requests = ref.child('notifications');
2  requests.on('child_added', function(requestSnapshot) {
3    var request = requestSnapshot.val();
4    sendNotificationToUser( request.username, request.message,
5      request.displayName,
6      function() {
7        requestSnapshot.ref.remove();
8      }
9  }, function(error) {
10   console.error(error);
11 });

```

²notifications



Obr. 5.2: Znáznornenie komunikácie pri posielaní *cloudových správ*. Zariadenie A vkladá dáta do uzlu databázy, na ktorom načúva server s *Node.js*. Tento server spracuje dané dáta a ďalej ich pošle serveru *Firebase* pre vytvorenie notifikácie a čaká na odpoveď. Pri úspešnom odoslaní zariadeniu B sú dáta z databázy vymazané.

5.2 Offline databáza

Pri vývoji na platforme *Android* je viacero možností, ako uchovávať *offline* dáta. Pri menšom množstve jednoduchých informácií je vhodné použiť tzv. zdieľané nastavenia³. Využiť sa dajú napríklad pre ukladanie nastavení aplikácie, teda hodnôt, kde je predpoklad, že pre danú premennú je potrebné poznať len jednu hodnotu.

Inou možnosťou je ukladanie dát v súboroch⁴. Takáto možnosť je vhodná pre ukladanie dát do pamäti, či už externej alebo internej, ktorých je veľké množstvo. Jedná sa zväčša o multimediálne súbory alebo súbory prijaté cez sieť. V prípade tejto aplikácie môžeme hovoriť napríklad o ukladaní obrázkov z chatových konverzácií.

Pre ukladanie dát, ktoré nie sú dôležité pre ostatných užívateľov je použitá *SQLite* databáza. Takýto druh databázy funguje ako tzv. vstavaná databáza, čo znamená, že namiesto databázy ako samostatného procesu existuje databáza v rámci používanej aplikácie[2].

V tomto prípade aplikácia komunikuje s databázou v jazyku *Java* priamo pomocou dodávanej knižnice. Databáza uchováva dáta, ku ktorým by ostatní užívatelia nemali mať prístup a užívateľ by sa k nim mal dostať aj mimo pripojenia na internet.

³z angl. shared preferences - <https://developer.android.com/training/basics/data-storage/shared-preferences.html>

⁴<https://developer.android.com/training/basics/data-storage/files.html>

Využitie

V navrhutej databáze sa vo verzii 1.2 nachádza jedna tabuľka (Obrázok 5.3) nesúca informácie o položkách tzv. skladu a to sú:

- Názov položky
- Nnožstvo danej položky
- Kritické množstvo danej položky, ktoré znamená kritickú hodnotu, podľa ktorej bude užívateľovi zobrazované upozornenie o nedostatku tovaru
- Veľkosť kroku po akom sa zväčšuje alebo znižuje množstvo položky v sklade pomocou tlačidiel
- Jednotke danej položky

Do budúcnosti nie je vylúčené, že položky bude možné synchronizovať s *online* databázou pre potreby viacerých užívateľov. V momentálnej verzii sa však jedná len o *offline* dáta, ku ktorým sa ostatní užívatelia nevedia dostať. Je tomu tak aj z dôvodu ochrany poskytovateľov a ich pracovných postupov.

Takto ukladané dáta sú však náchylné na *dáta aplikácie*. S ich prípadnou stratou musí užívateľ počítať.

Item
id : integer (PK)
name : text
count : real
critical : real
step : real
unit : text

Obr. 5.3: Tabuľka položiek *skladu*

5.3 Ostatné služby

V aplikácii sú využívané aj ďalšie služby a voľne dostupné knižnice, ktoré často dopĺňujú alebo uľahčujú implementáciu niektorých častí.

Facebook

V momentálnej verzii vyvíjanej aplikácie 1.2 je *Facebook API*⁵ využívané v spolupráci so službami *Firebase* pre autentifikáciu užívateľov, pričom je z jednotlivých účtov získavaná *e-mailová* adresa, meno a *URL adresa* profilového obrázku. V budúcnosti by sa dali pomocou tejto knižnice získať údaje napr. pre účely umiestňovania reklamy, ako sú napr. pohlavie, dátum narodenia atď.

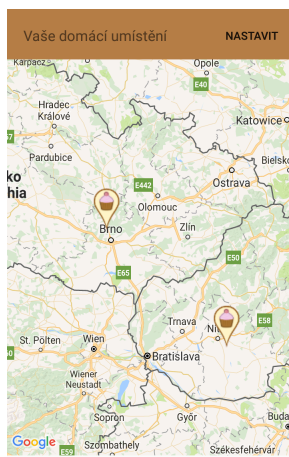
Bude tiež možné využiť napr. funkciu zdieľania na *Facebooku*, rôzne analýzy, posielanie pozvánok do aplikácie alebo samotné umiestňovanie reklamy.

⁵<https://developers.facebook.com/>

Google Maps

Hlavným účelom služieb *Google Maps*⁶ je zobrazovanie máp a práca s nimi, prípadne ich prispôsobenie aplikácii napr. farebnou schémou.

Rozhranie *Google Maps* je zobrazované v rámci aktivity na ukladanie polohy užívateľa (Obrázok 5.4). Z rozhrania sú tiež využívané reakcie na dlhý alebo krátky dotyk užívateľa na danom geografickom mieste, kde krátky dotyk na značku v mape znamená odstránenie značky a dlhý dotyk znamená pridanie značky do mapy na dané miesto. Spracovanie práve takýchto užívateľských vstupov má na starosti toto rozhranie.



Obr. 5.4: Aktivita so zobrazenými mapami *Google Maps*

Geofire

Táto knižnica⁷ v nadväznosti na predchádzajúcu sekciu a na sekciu o reálno-časovej databáze služieb *Firebase* ukladá polohy užívateľov do vybraného uzla databázy a pomocou tzv. *geohashu* vie určiť polohu užívateľov. Pôvodný systém ukladania užívateľských polôh pomocou unikátnej kombinácie 28 znakov užívateľa bolo potrebné rozšíriť z dôvodu možnosti viacnásobnej polohy užívateľov o poradové číslo danej polohy. Bolo preto nutné upraviť algoritmus pridávania a mazania uložených polôh. Takisto je možné pomocou tejto podpornej knižnice zistiť užívateľov v zadanom rádiuse, čo je možné využiť v prípade vyhľadávacej podmienky pre poskytovateľov služieb v okolí užívateľa (ukážka zdrojového kódu 5.2).

⁶<https://developers.google.com/maps/>

⁷<https://github.com/firebase/geofire> - licencia: *The MIT License (MIT)*

Ukážka 5.2: Zdrojový kód ukazujúci použitie *Geofire* v rámci aplikácie. Tento kód je volaný v prípade, ak sa používateľ rozhodne hľadať poskytovateľa v rádiuse.

```
1 GeoQuery geoQuery = geoFire.queryAtLocation(new
    GeoLocation(currentUser.getLatitude(), currentUser.getLongitude()),
    distance);
2 geoQuery.addGeoQueryEventListener(new GeoQueryEventListener() {
3     @Override
4     public void onKeyEntered(String key, GeoLocation location) {
5         addUserToList(key.substring(0,28));
6     }
7     ...
8 });
9 }
```

OkHttp

Pre vytváranie a odosielanie *HTTP* požiadaviek je použitá knižnica *OkHttp*⁸. Táto knižnica je využívaná na komunikáciu s vlastným serverom⁹. Vytvára hlavičku a telo požiadavky s formulárom na spracovanie pre server. Je tiež schopná prijímať odpovede od servera na následné spracovanie.

Picasso

Táto knižnica¹⁰ slúži primárne na asynchrónne sťahovanie a cashovanie obrázkov. Nie je pre to nutné v rámci aplikácie zakaždým sťahovať dáta zo serverov, ale dané obrázky sú *cashované* podľa *URL adresy* a sú sprístupňované. Knižnica sa tiež stará o transformáciu obrázkov na požadovanú veľkosť s cieľom o minimalizáciu použitej pamäti.

Knižnica je využívaná vo verzii aplikácie 1.2 pre ukladanie profilových obrázkov užívateľov prihlásených pomocou *Facebooku*.

ShortcutBadger

Kvôli zjednoteniu pridávania a odoberania notifikačných čísiel pri ikone aplikácie je využívaná táto knižnica¹¹. Takéto notifikačné čísla sú pridávané napr. pri prijatí chatových správ.

Knižnica sa snaží o zjednotenie spôsobu pridávania notifikačných čísiel u rozličných výrobcov modifikácií operačného systému *Android*.

⁸<https://github.com/square/okhttp> - licencia: *Apache License 2.0*

⁹Pre verziu so serverovou časťou s PHP skriptom

¹⁰<https://github.com/square/picasso> - licencia: *Apache License 2.0*

¹¹<https://github.com/leolin310148/ShortcutBadger> - licencia: *Apache License 2.0*

Kapitola 6

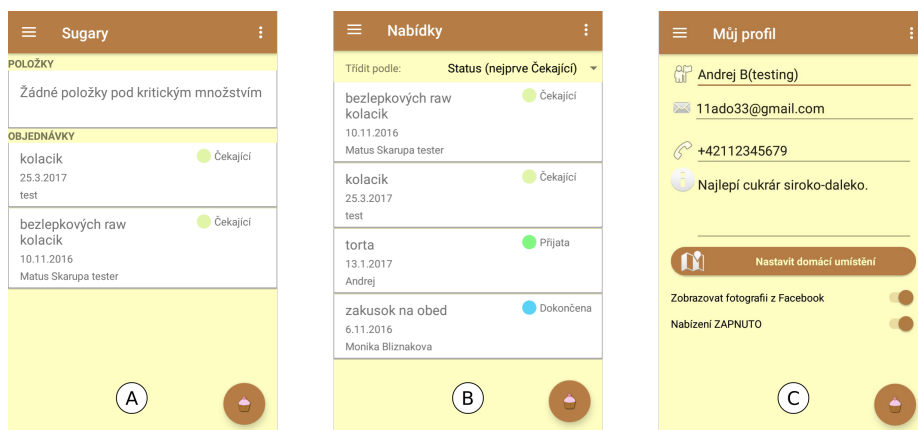
Implementácia užívateľského rozhrania

Užívateľské rozhranie aplikácie predstavuje nemenej podstatnú časť od ostatných súčastí aplikácie. Jeho implementácia vyplýva z návrhu užívateľského prostredia popísaného v kapitole 2. Nasledujúce sekcie sú venované hlavným prvkom, ktoré tvoria užívateľské rozhranie aplikácie.

6.1 Aktivity a fragmenty

Hlavnými zobrazovacími prvkami užívateľského rozhrania sú *aktivity* a *fragmenty*. Aktivitu je možné chápať ako celoobrazovkové okno zobrazované užívateľovi. Toto okno môže zobrazovať rôzne komponenty užívateľského rozhrania alebo môže pri zachovaní hlavných prvkov rozhrania zobrazovať fragmenty.

Tieto hlavné prvky rozhrania zobrazujú často *textové polia* s informáciami alebo iné grafické prvky a zoznamy. Ukážka hlavnej aktivity s fragmentami je demonštrovaná na obrázku 6.1.

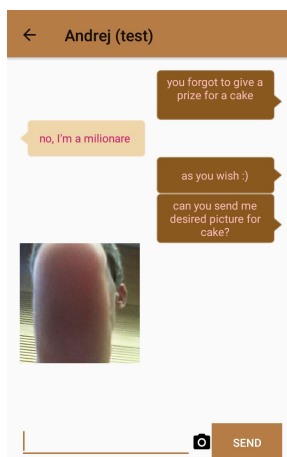


Obr. 6.1: Aktivita v tomto prípade zobrazuje hlavné ovládacie prvky, ako sú horná lišta s tlačidlami a *plávajúce tlačidlo* v pravej časti obrazovky. Fragmenty sa zobrazujú na hlavnej časti obrazovky v prostriedku. Konkrétne sa jedná o fragment A s úvodnou obrazovkou, B fragment s ponukami pre poskytovateľa a profilový fragment C s možnosťami úpravy osobných údajov. Toto konkrétne zobrazenie je dostupné pre užívateľov typu *poskytovateľ*.

6.2 Vlastné adaptéry

Adaptéry¹ môžu vo všeobecnosti slúžiť ako spojovací článok medzi prvkom užívateľského rozhrania a dátami. Často sa napríklad jedná o zobrazovanie dát v zoznamoch atď.

Pôvodné adaptéry však neposkytujú požadovaný vizuálny výsledok. Preto je potrebné navrhnúť vlastné adaptéry pre rôzne druhy využitia. Rozmanitosť adaptérov najlepšie ilustruje obrázok 6.2.



Obr. 6.2: Aktivita pre chat je asi najlepšia pre demonštráciu rozmanitosti použitých adaptérov. Adaptéry sa v tejto časti skladajú z obrazovej a textovej časti, ktorých výber záleží na prezentovaných dátach. Taktiež je v tomto prípade použitá technológia tzv. *9-patch* obrázkov³, ktorá prispôbuje veľkosti bublín zobrazujúce chatové správy od užívateľov pri zachovaní požadovaných tvarov. Rôzne adaptéry sú použité aj v ukážke na Obrázku 6.1.A, kde sú použité viaceré adaptéry v zoznamoch s dynamicky meniacimi výškami podľa veľkosti obsahujúcich adaptérov.

6.3 Dialógové okná

Dialógové okná je možné považovať za akýsi doplnok hlavného rozhrania, teda aktivít a fragmentov. Je tomu tak z dôvodu, že sú zobrazované zväčša nad týmito hlavnými prvkami užívateľského rozhrania a slúžia na zisťovanie rôznych doplnkových informácií alebo ako potvrdzovacie okná pre rôzne činnosti (viď obrázok 6.3).

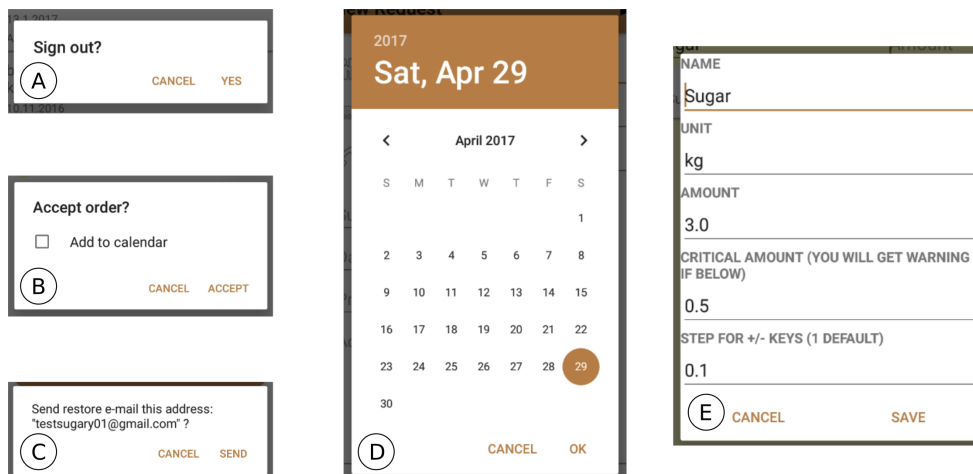
6.4 Systémové a iné rozhrania

Aplikácia tiež využíva rozhrania daného operačného systému. Takéto rozhrania sú prístupované pomocou tzv. systémových požiadaviek⁴, ktoré sú volané s patričnými parametrami.

Medzi takéto rozhrania patrí napríklad rozhranie systému pre vytáčanie telefónnych čísiel, rozhranie predvolenej aplikácie pre posielanie *e-mailov* alebo rozhranie pre ukladanie termínov do systémového kalendára. Takisto je použité rozhranie východiskového prehliadača obrázkov pre prezeranie obrázkov z chatu aplikácie v skutočnej veľkosti obrázka.

¹<https://developer.android.com/reference/android/widget/Adapter.html>

⁴<https://developer.android.com/guide/components/intents-common.html>



Obr. 6.3: Dialógové okná A až C predstavujú ukážky dialógových okien slúžiacich pre potvrdzovanie rôznych činností, výraznejšie ovplyvňujúcich činnosť užívateľa. Dialógové okno B je doplnené aj o možnosť, ktorou je užívateľovi uložená daná objednávka do systémového kalendára. Dialógové okno D slúži ako pomôcka pre vyberanie dátumu, čím sa dosiahne aj požadovaný formát dátumu pre kontext aplikácie. Vo verzii aplikácie 1.2 pribudlo dialógové okno E s vlastným návrhom. Slúži na editáciu položiek zoznamu skladu surovín.

V aplikácii je tiež využívané rozhranie poskytované službou *Google Maps*. Toto rozhranie je zobrazované v samostatnej aktivite určenej pre zadávanie polohy užívateľov. Bližšie popísané v kapitole 5.3.

6.5 Jazykové verzie

Z návrhu vyplýva, že aplikácia má byť spočiatku cielená pre český a slovenský trh. Je preto nutné prispôbiť užívateľské rozhrania pre týchto užívateľov a to hlavne lokalizáciou zobrazovaných textov. Tomu slúžia zdrojové súbory textových reťazcov pre každý jazyk. V momentálnej verzii sa teda nachádzajú 3 jazykové lokalizácie a to česká, slovenská a ako vývojársky jazyk - anglická.

Kapitola 7

Publikovanie a testovanie aplikácie

Aplikácia je v momentálnej verzii 1.2.x publikovaná v obchode *Google Play*¹.

7.1 Publikovanie aplikácie

Z dôvodu publikácie, ale aj pre marketingové účely bolo vytvorené video, znázorňujúce hlavnú myšlienku aplikácie. Toto video je dostupné zo stránok *YouTube*² a bolo vytvorené pomocou programov *Adobe Premiere Pro*³ a *Adobe After Effects*⁴. Niektoré grafické prvky videa boli prebraté zo stránky *Freepik*⁵.

K publikovaniu aplikácie bolo potrebné generovať podpísaný súbor `apk`, ktorý bol následne nahratý cez webové rozhranie do vývojárskej konzoly *Google Play*. Po istom čase spracovania sa aplikácia ocitla priamo v obchode *Google Play*, avšak ako je upozornené názvom a grafikou pri aplikácii jedná sa o *beta testovací*⁶ režim. Dáta v aplikácii sú teda tiež testovacie a užívatelia sú s týmto faktom oboznámení. Na konkrétny odkaz obchodu *Google Play* je možné sa preniesť po naskenovaní QR kódu na obrázku 7.1.



Obr. 7.1: QR kód s odkazom na *Google Play* s logom aplikácie.

¹<https://play.google.com>

²<https://www.youtube.com/watch?v=ECcS5pXqYXU>

³<http://www.adobe.com/sk/products/premiere.html>

⁴<http://www.adobe.com/sk/products/aftereffects.html>

⁵<http://www.freepik.com/>

⁶fáza testovania, kde je prizvaná verejnosť

7.2 Testovanie

Testovanie je neoddeliteľná súčasť tvorby užívateľského rozhrania, ale aj všeobecne vývoja akéhokoľvek produktu. Preto je potrebné zvážiť, akým spôsobom bude testovaná vytvorená aplikácia.

Je preto vhodné, ak sa bude testovať produkt v dvoch fázach.

Testovanie počas vývoja

Výsledky vývoja je potrebné testovať priebežne a to pred vydaním aplikácie iným užívateľom. Toto testovanie je možné rozdeliť do dvoch fáz:

- Testovanie pozadia aplikácie a dát
- Testovanie užívateľského rozhrania a zobrazovania dát

Tento proces častokrát prebieha súčasne a sú testované oba faktory aplikácie zároveň.

Pozadie aplikácie a dát

Takéto testovanie prebieha väčšinou za pomoci vývojového nástroja *Android Studio*, konkrétne štúdiou vypisovaných informácií do konzoly spustenej aplikácie.

Sem sú konštantne posielané niektoré informácie, ktoré môžu v budúcnosti napomôcť k skoršiemu odhaleniu chýb týkajúcich sa dát, ale niekedy sú vypisované dáta aj priebežne, hlavne pri detailnejšom pátraní po chybách.

Užívateľské rozhranie a zobrazovanie dát

Pri tomto testovaní je primárne zisťované, či sa podarilo dosiahnuť navrhnuté užívateľské rozhranie s požadovaným obsahom. Ako nástroj slúži súčasť *Android Studia* - *Android Virtual Device*.

Pomocou neho sú simulované rôzne zariadenia s rôznymi parametrami, napríklad s často rozličnými rozlíšeniami displejov zariadení. Pri tomto testovaní je často odhalené nesprávne navrhnuté užívateľské rozhranie, či už z pohľadu neprehľadnosti výsledného zobrazenia, alebo aj z pohľadu nelogickosti návrhu.

V rámci testovania bolo potrebné po prvých vydaniach pridať aj zariadenia s nižšími rozlíšeniami ako *HD*.

Užívateľské testovanie

Tento spôsob testovania predstavuje ďalšiu dôležitú časť vývoja produktu. Pri takomto testovaní je zisťované na istej vzorke užívateľov či navrhnuté alebo implementované riešenie spĺňa požiadavky užívateľov.

K testovaniu sú potrebné dva typy užívateľov: *cukrárov* a *potenciálnych zákazníkov*. Prvoradá pri testovaní je komunikácia s užívateľom, kde je užívateľovi nejakým spôsobom prezentovaný návrh riešenia a užívateľ spätne posielal svoj názor na testované riešenie.

Takáto komunikácia prebieha väčšinou pomocou elektronickej komunikácie, hlavne v prípade užívateľov typu *cukrár*. Pri druhom type užívateľov sa často jedná o kombináciu elektronickej komunikácie s užívateľom a osobného stretnutia.

Užívatelia mávajú za úlohu otestovať konkrétne časti aplikácie alebo návrh v istom časovom intervale. Následne je od nich vyžadovaná spätná väzba v podobe rôznych otázok

týkajúcich sa aktuálne riešeného problému. Otázky sa môžu týkať aj budúcej funkcionality a názoru užívateľov na ňu.

Testovania sa zúčastnilo 5 užívateľov pre každú testovaciu skupinu, teda poskytovateľov a potenciálnych zákazníkov.

Príklad úlohy:

Úloha: *Spustíte aplikáciu a pridajte si offline objednávku s vymyslenými údajmi. Následne danú objednávku prijmite s pridaním udalosti do kalendára.* Od užívateľov potom bolo zisťované, či sa im úlohu podarilo splniť a vidia v rámci systému objednávku. Takisto či vidia v systémovej kalendári udalosť.

Výsledok: Z piatich užívateľov sa požadovaný výsledok podarilo dosiahnuť len jednému, keďže užívatelia nevedeli o tlačidlo, ktoré na vytvorenie objednávky slúži. Následne bol na úvodnú obrazovku pridaný návod⁷.

Takéto informácie sú od užívateľov po intervale testovania zozbierané, analyzované a je navrhnuté následné riešenie, ktoré je implementované a opätovne testované.

Užívateľské testy často odhalia triviálne chyby, ktoré boli prehliadané pri implementácii. Zvykne sa však stať, že užívateľ odhalí chybu, ktorá je vážna a vo veľkej miere ovplyvňuje prácu s navrhnutou aplikáciou. Chyby nemusia byť len charakteru týkajúceho sa užívateľského rozhrania.

Problémy odhalené užívateľským testovaním

Medzi najčastejšie nájdené problémy sa radili problémy so samotným zobrazovaním informácií.

Často sa jednalo o rôzne preklepy alebo chyby v preklade pre českú lokalizáciu. Ne-zriedkavé chyby predstavovalo násobné zobrazovanie tých istých informácií nesprávnym ošetrením preberania dát z databázy.

Niekedy však boli samotní užívatelia nespokojní s navrhnutým rozhraním. Išlo o chyby návrhu alebo sa jednalo len o fakt, že užívatelia neboli na rozhranie zvyknutí. Takéto ne-dorozumenia by bolo možné v budúcnosti riešiť napr. interaktívnym sprievodcom v rámci aplikácie.

Zo začiatku nastali väčšie komplikácie pri nasadzovaní aplikácie hlavne z dôvodu nevedomosti nutnosti použitia rôznych bezpečnostných nastavení⁸ pre *debug* a *release* verziu aplikácie. Taktiež sa pri prvých vydaniach objavili problémy s rôznymi rozlíšeniami displejov zariadení z dôvodu nesprávneho návrhu rozhrania. Údaje sa tak často vôbec nezobrazovali alebo v neusporiadanej štruktúre. Bolo tomu hlavne v prípadoch, keď užívatelia používali zariadenia s nižším ako *HD* rozlíšením displeja.

Niektoré chyby, priamo netýkajúce sa užívateľského rozhrania, sa v aplikácii nachádzali v dôsledku zlého návrhu aj počas niekoľkých mesiacov. Dôsledky sa však začali prejavovať až v určitých, špecifických situáciách. Napríklad pri zobrazovaní viacerých obrázkov v chate bola z dôvodu optimalizácie pamäte zatváraná aktivita na pozadí, čo spôsobilo po návrate z aktivity chatu pád aplikácie, keďže aktivita bola reštartovaná bez vstupných parametrov.

Takéto chyby bolo často možné odstrániť jednoduchým nastavením alebo úpravou. Dôvodom pravdepodobne je začiatočnícka neznalosť niektorých vecí.

⁷ten sa v súčasnej verzii nachádza ako samostatný fragment a je sa k nemu možné dostať cez menu aplikácie.

⁸rôzne *API* kľúče pre *release* verziu aplikácie atď.

7.3 Budúcnosť aplikácie

Komunikácia s užívateľmi bola prínosná nielen z pohľadu súčasných verzií aplikácie, ale často sami užívatelia prinášali vlastné nápady pre zlepšenie aplikácie v budúcnosti. Často sa jedná o funkcionality, ktorá by mimo komunikácie so zákazníkmi, primárneho cieľa aplikácie, ponúkla poskytovateľovi viac dôvodov takúto aplikáciu používať.

Z toho vyplýva aj smer, akým by sa mala aplikácia uberať, keďže zo strany potenciálnych zákazníkov sa skoro v každom prípade jednalo o pozitívne reakcie na myšlienku aplikácie. V prípade poskytovateľov boli vo veľa prípadoch reakcie odmietavé, čo obhajovali dostatočnosťou z ich strany používaných nástrojov.

Preto bude potrebné dopĺňať funkcionality podporujúcu ich činnosti. Stalo sa tak už pridaním napríklad *offline skladu*, ktorého funkcie sa však môžu rozšíriť napríklad o spravovanie expirácie používaného tovaru alebo automatického generovania nákupných zoznamov s možnosťou posielania vo forme *SMS* alebo *e-mailom*.

Z komunikácie s poskytovateľmi tiež vyplýva návrh na funkciu adresára, ktorá sa tiež už v dnešnej verzii nachádza v obmedzenej podobe s využitím systémového kalendára.

Návrhy na zmeny dávali aj užívatelia v roly potenciálneho zákazníka, ktorí by uvítali širšie možnosti pri zadávaní ponúk pre poskytovateľov. K rozšíreniam patrí vytvorenie kolónky pre pridanie druhého dátumu, čím by bolo možné zadať interval, kedy má byť tovar doručený. Ďalšou zaujímavou požadovanou vlastnosťou je možnosť filtrovať poskytovateľov podľa nejakých základných kritérií, ako môže byť napríklad kategória požadovaných výrobkov. Samotní poskytovatelia by sa tak mali zaradiť do kategórie výrobkov, ktoré sú pravdepodobne schopní vyrobiť, čím by sa zjednodušil proces výberu poskytovateľa služieb. Návrhy sa taktiež týkali zjednodušenia niektorých činností, ako je vyhľadávanie poskytovateľov v priestore a to bez nutnosti zadania statickej polohy zákazníka. Zákazníková poloha by mohla byť získavaná automaticky, napr. zo signálu *GPS*.

Ďalším aspektom rozširovania funkcionality aplikácie je nutnosť prepájať súčasne používané spôsoby poskytovateľov (hlavne komunikácia cez *Facebook*) s aplikáciou. Bolo by to možné doceliť pomocou prelinkovania aplikácie na sociálne siete, prípadne možnosťou poskytovateľov vytvárať si unikátny identifikátor, pomocou ktorého by ho zákazníci vedeli kontaktovať z prostredia aplikácie.

Ak by sa aplikácia dostala do povedomia viacerých poskytovateľov, bolo by možné zahrnúť aj funkciu, pomocou ktorej by bolo možné z pohľadu zákazníka vložiť akúsi nikomu nepriradenú objednávku. O tú by mohli daní poskytovatelia súťažiť. Ak by zákazníkovi nezáležalo na cene, rozhodujúca by bola rýchlosť obsluhy objednávky.

Pri úspechu takejto aplikácie by bolo možné uvažovať nad rozširovaním záberu trhu s poskytovateľmi služieb nielen na cukrárov, ale aj napr. kvetinárov, stolárov, krajčírov atď. V takom prípade by bolo najvhodnejšie aplikáciu rozdeliť na viacero druhov, kde by zákazník pracoval s jednou, centrálnou aplikáciou poskytujúcou možnosť kontaktovania poskytovateľov rôznych služieb. Poskytovatelia by mali k dispozícii špecifickú, pre nich určenú aplikáciu s podpornými funkciami.

V takomto úspešnom prípade je však nutné zaviesť opatrenia, kde bude potrebné verifikovať hlavne poskytovateľov služieb, čo je možné dosiahnuť aj prostredníctvom monetizácie, teda spoplatnenia funkcií aplikácie pre poskytovateľov.

Veľký priestor ponúka možnosť rozširovať podporu operačných systémov aplikácie, hlavne *iOS* alebo *webová* verzia pre počítače. Možnosťou pre rozšírenie je aj implementácia viacerých jazykových lokalizácií. To však rozhodne nie je práca pre jedného človeka.

Kapitola 8

Záver

Cieľom práce bolo primárne spoznať technológie vhodné pre vývoj mobilných aplikácií pre platformu *Android* s čím súvisí technológia *Firebase*. Tá sa javila ako zaujímavé riešenie pre začínajúce mobilné aplikácie, hlavne z dôvodu možnosti bezplatného využitia v obmedzenom množstve.

Hlavnou výhodou tejto technológie je určite fakt, že programátor je odbremený od väčšiny zabezpečovania a implementácie serverovej časti aplikácie. Po pár mesiacoch skúseností je možné tvrdiť, že konfigurácia novej aplikácie pre komunikáciu so servermi *Firebase* by skúsenému užívateľovi nezabrala viac ako hodinu práce. Konfigurácia je relatívne jednoduchá, s dobrou dokumentáciou od poskytovateľa služieb.

Niektoré zvyklosti v rámci tejto technológie sa však od konvenčných riešení líšia. Na prácu s databázou si bolo potrebné zvyknúť, respektíve bolo potrebné zmeniť prístup k návrhu štruktúr databázy v porovnaní s databázami *SQL*. S týmito odlišnosťami súvisí aj nevýhoda takejto databázy. V momentálnej verzii rozhrania je totiž komplikované dotazovať sa na objekty s podmienkou na viac ako jeden atribút. V takýchto situáciách je potrebné vytvárať vlastné indexy tvorené kombináciou požadovaných atribútov, z čoho vzniká redundancia.

V rámci tejto aplikácie by tento problém mohol nastať hlavne v prípade určovania polohy podľa dát z databázy, kde by do úvahy pripadalo viacero atribútov. Tento problém však šikovne rieši podporná knižnica *Geofire* pracujúca tiež s formou indexovania - *geohashu*.

Zároveň so zoznamovaním sa s technológiami, bolo potrebné nájsť oblasť, kde by aplikácia mohla nájsť uplatnenie. Prebiehala teda analýza trhu, kde bolo potrebné v prvom rade vnímať okolitý svet a trh, ktorý je značne komplikovaný. Analýzou bola následne vybraná oblasť trhu pre vývoj aplikácie. Táto oblasť bola zo začiatku pomerne obsiahla a to podpora činnosti malých alebo domácich výrobcov respektíve remeselníkov. Vývoj aplikácie bol teda smerovaný na podporu cukrárov. Takáto podpora mala spočívať vo vytvorení platformy pre komunikáciu medzi výrobcom a jeho zákazníkom. Pre zákazníka to mal byť nástroj pre vyhľadanie výrobcu v jeho okolí. Táto hlavná myšlienka bola potom transformovaná do návrhu konkrétneho riešenia. Návrhu sa zúčastnili samotní poskytovatelia v podobe vyplnenia dotazníkov.

Poskytovatelia, ako aj potenciálni zákazníci, boli účastní aj samotného testovania aplikácie, čím bolo mimo interného testovania odhalených množstvo chýb, týkajúcich sa ako rozhrania, tak pozadia aplikácie.

Aplikácia bola postupne rozširovaná o rôzne funkcie, s čím súvisela potreba zoznámiť sa s viacerými, v dnešnej dobe používanými technológiami, ktoré tvoria niektoré podporné časti aplikácie a dopomáhajú vytvoriť celok.

Zaujímavou časťou implementácie bola určite aj fáza, keď bolo nutné riešiť problém so zabezpečením serveru notifikačných správ a bolo nutné preštudovať a vymyslieť inú, bezpečnejšiu možnosť, ako takéto správy posielat. Výsledkom je opäť zaujímavé riešenie v podobe *cloudovej* aplikácie na servery.

Aktuálna verzia aplikácie je dostupná z obchodu *Google Play* (dostupné naskenovaním *QR* kódu z obrázku 8.1), na ktorom sa nachádza v rámci verejného (nie *alfa*, ani *beta*) vydania s grafickým označením toho, že aplikácia je v rámci práce vo verzii *beta*.



Obr. 8.1: QR kód s odkazom na *Google Play* s logom aplikácie.

Za posledný mesiac, podľa štatistík *Google Firebase*, použilo aplikáciu viac ako 20 užívateľov. Niektorí, aj vďaka konferencii *Excel@FIT*¹, mali rôzne pripomienky, nápady a odporúčania pre túto konkrétnu aplikáciu, ale aj pre iné, budúce aplikácie.

Vo všeobecnosti by sa dalo tvrdiť, že z pohľadu zákazníkov sa jedná o prínosnú aplikáciu. Pri poskytovateľoch bolo viac skeptických názorov, čomu bol prispôsobený aj vývoj aplikácie pre poskytnutie pridanej hodnoty pre poskytovateľa.

¹<http://excel.fit.vutbr.cz/>

Literatúra

- [1] Mike Cantelon, Z. H. N. R., Marc Harter: *Node.js in Action*. Manning Publications Co., 2014, iISBN: 9781617290572.
- [2] Owens, M.: *The Definitive Guide to SQLite*. Apress, 2006, iISBN: 978-1-59059-673-9.
- [3] Singh, N.: *Study of Google Firebase API for Android*. IJIRCCE, 2016, iISSN: 2320-9801.
- [4] Ujbányai, I. M.: *Programujeme pro Android*. Grada Publishung, 2012, iISBN: 9788024739953.

Prílohy

Zoznam príloh

A	Analýza užívateľov - dotazník s odpoveďami	43
B	Pracovný postup	46
C	Návrhy obrazoviek pre verziu 1.2	50
D	Obsah CD	51

Príloha A

Analýza užívateľov - dotazník s odpoveďami

Otázka 1

Predstavte si, že ste si práve nainštalovali aplikáciu do Vášho telefónu. Akým spôsobom očakávate, že sa do nej prihlásite/identifikujete? (možnosť vybrať viac)

Odpovede:

- Pomocou Facebook účtu - 30%
- Pomocou Google účtu - 10%
- Pomocou klasickej registrácie a následného prihlásenia - 60%
- Je mi to jedno - 20%

Otázka 2

Ste prihlásení v aplikácií. Aké údaje chcete aby o Vás potenciálny klient videl? (možnosť vybrať viac)

Odpovede:

- Meno / Názov / prezývka - 90%
- Adresa - 50%
- Fotografia - 40%
- Vzdialenosť medzi Vami a klientom - 80%
- Telefónne číslo - 80%
- E-mail - 80%
- Odkaz na sociálne siete - 70%
- Recenzie Vašich iných zákazníkov - 70%
- Iné (možnosť doplniť) - 0%

Otázka 3

Využili by ste možnosť "súťažiť" s inými výrobcami? (predstavte si to ako súťaž, niečo ako verejné obstarávanie pre zákazníka)

Odpovede:

- Áno - 30%
- Momentálne neviem - 40%
- Nie - 30%

Otázka 4

Čo je pre Vás podstatné pri zákazníkovej objednávke? (viac možností)

Odpovede:

- Cena - 70%
- Druh výrobku - 60%
- Obrázok výrobku - 90%
- Termín objednávky (do kedy má byť tovar hotový) - 90%
- Kontakt na zákazníka - 60%
- Hmotnosť - 70%
- Počet kusov - 30%
- Spôsob dopravy tovaru - 20%
- Iné (možnosť doplniť) - 20% - Popis výrobku, príležitost na aku je, požadovaný vzhľad a chuť; príchúť torty, cenový limit

Otázka 5

Využili by ste chat so zákazníkom v rámci aplikácie?

Odpovede:

- Áno - 100%
- Momentálne neviem - 0%
- Nie - 0%

Otázka 6

Využili by ste chat so zákazníkom v rámci aplikácie?

Odpovede:

neviem; vkladanie terminov do kalendara; kalkulačka, adresár, kalendár; vkládání termínů do kalendáře, upozornění (např.nákup surovin); Vkladanie terminov, rozdelenie prijatych sprav - na oddelenie dohodnutych objednavok od dopytu, moznost oznacit si spravy podla terminov objednavky a moznost zoradit si spravy podla tychto oznaceni

Otázka 7

Vyskúšali by ste túto bezplatnú službu?

Odpovede:

- Áno - 100%
- Momentálne neviem - 0%
- Nie - 0%

Otázka 8

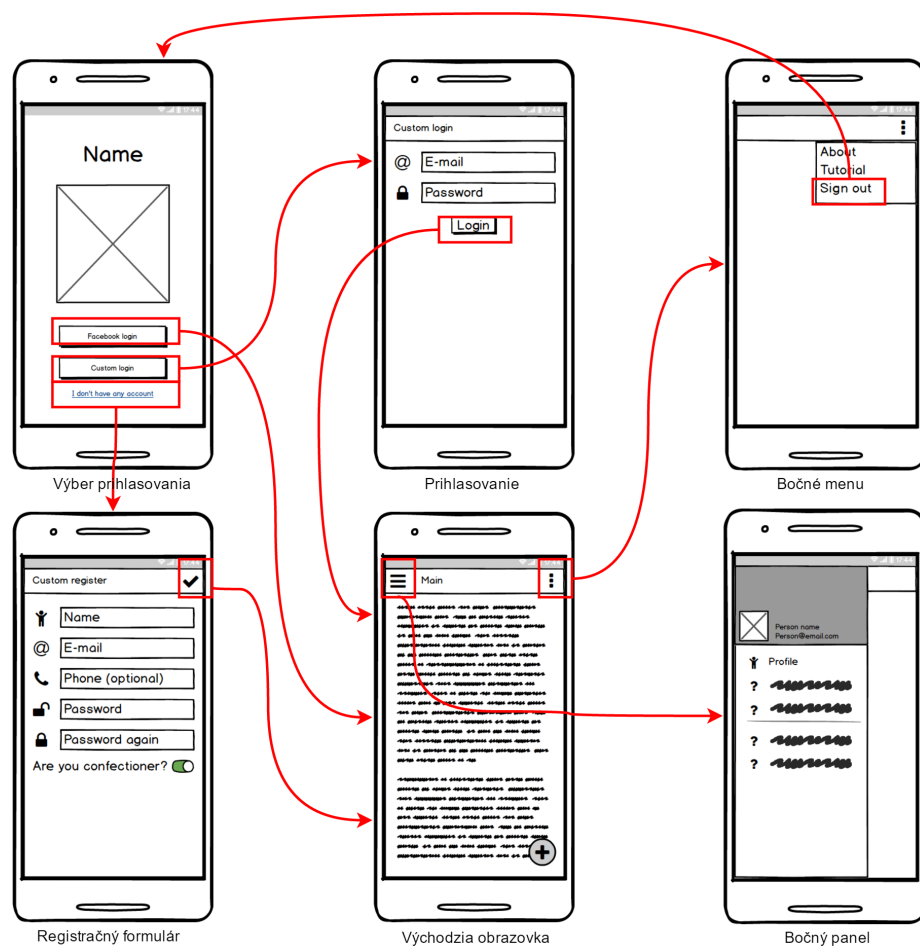
Napadá Vám podobné odvetvie (ako je cukrárstvo), ktoré by mohlo mať podobnú mobilnú aplikáciu?

Odpovede:

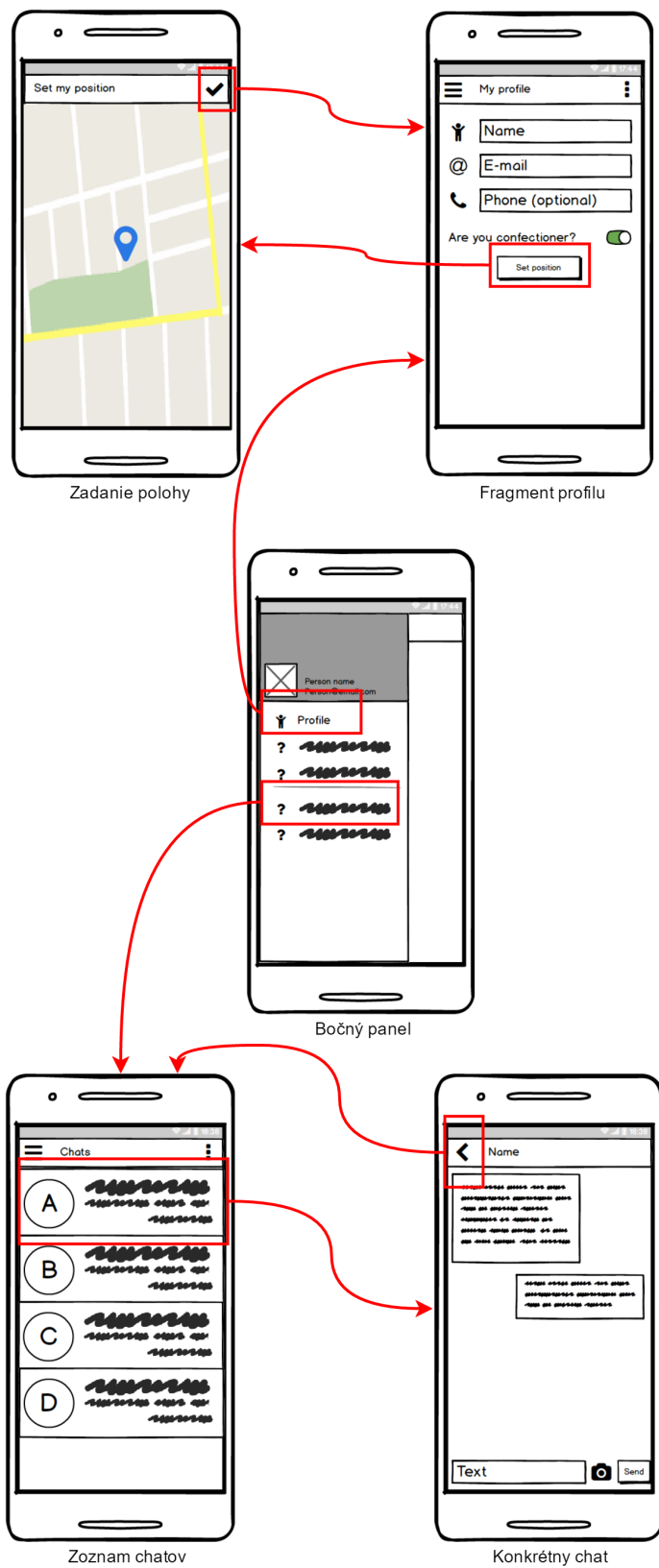
určitě by využili jiné obory; Kvety; Vyroba cohokolvek na objednavku....napr. apka pre stolara, krajcirku...; všetky možné živnosti- kominár, stolár, murár, krajčírka; ano, vela ludi si privyraba syrovymi tortami alebo inymi vyrobkami :)

Príloha B

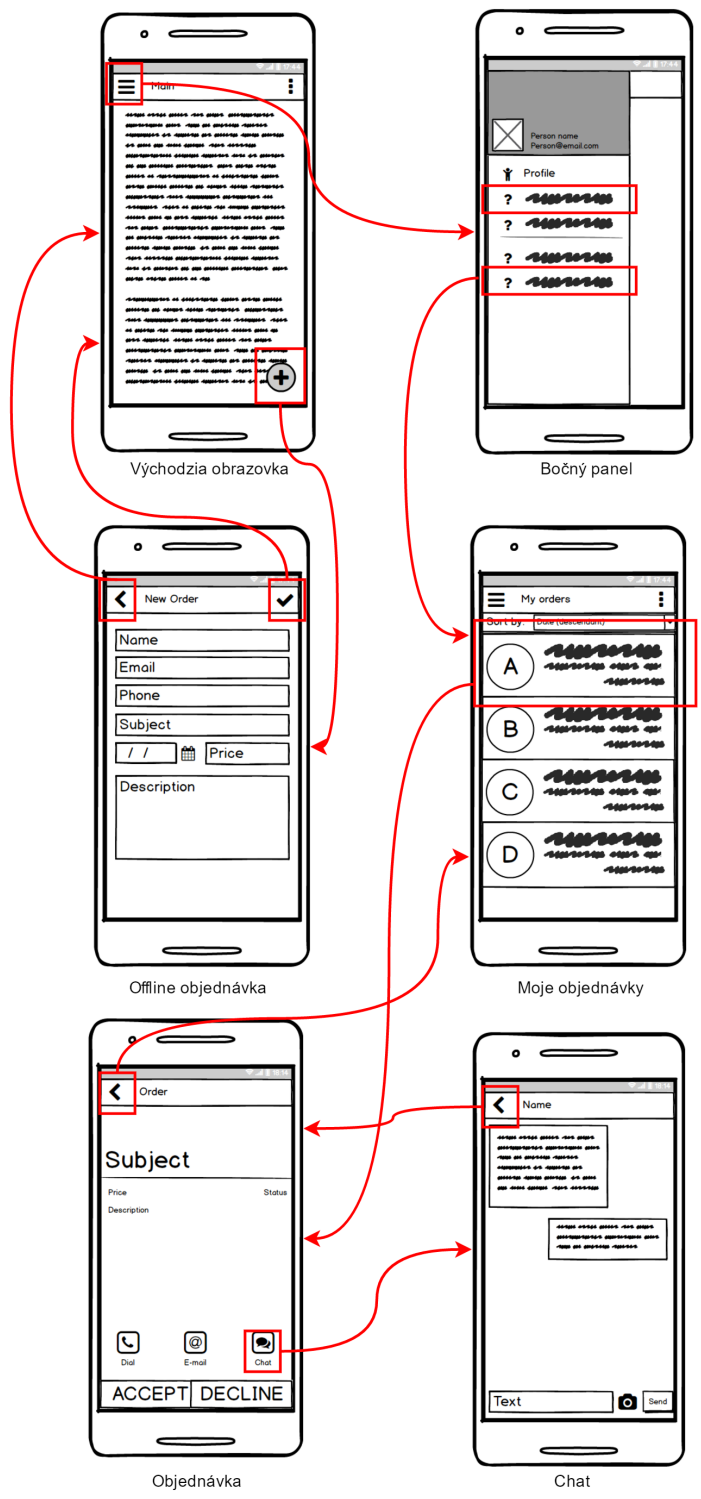
Pracovný postup



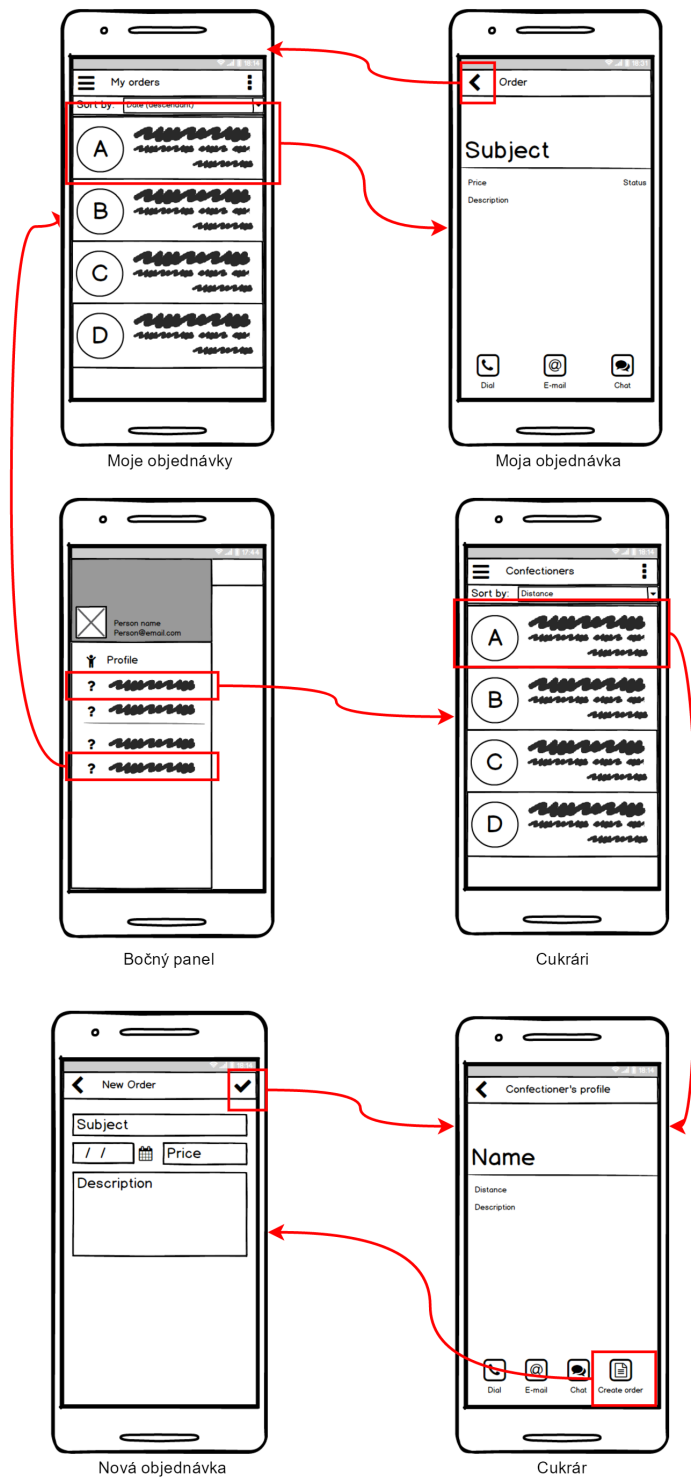
Obr. B.1: Všeobecný pracovný postup



Obr. B.2: Všeobecný pracovný postup pokračovanie



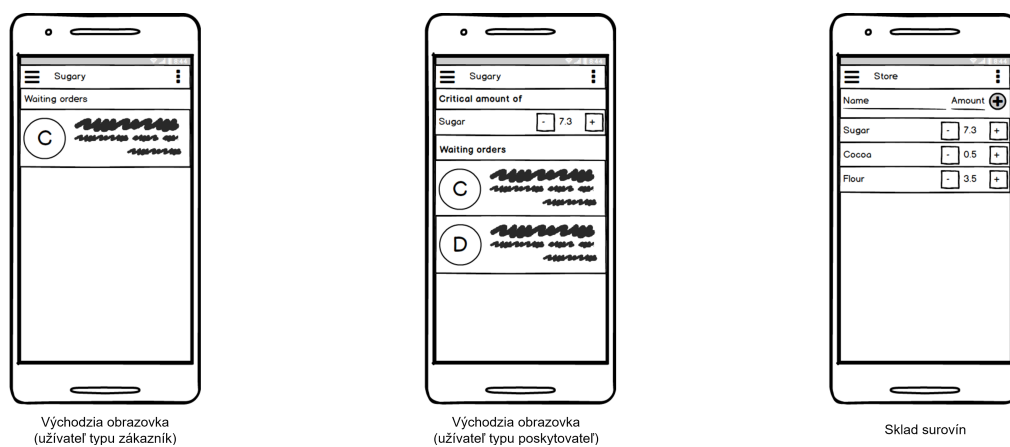
Obr. B.3: Pracovný postup cukrára



Obr. B.4: Pracovní postup zákazníka

Príloha C

Nárhly obrazoviek pre verziu 1.2



Obr. C.1: Doplnené úvodné obrazovky jednotlivých užívateľov s novým fragmentom pre užívateľa typu poskytovateľ, do ktorého sa dostane kliknutím na položku v bočnom paneli.

Príloha D

Obsah CD

Priložené CD obsahuje:

- `bp.pdf` - písomná správa vo formáte PDF
- `tex` - zložka so zdrojovým tvarom písomnej práce \LaTeX
- `src` - zložka so zdrojovými kódami aplikácie a serverov
- `manual.txt` - návod pre spozajzdnenie aplikácie a serverovej časti
- `sugary.apk` - inštalačný súbor aplikácie
- `poster.pdf` - propagačný plagát
- `2017-xblizn01-mobilni-aplikace-pro-poptavani-sluzeb.mp4` - propagačné video
- `popis-videa.xml` - súbor s popisom videa