



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 SPRÁVU PRACÍ
VE SDÍLENÉM UBYTOVÁNÍ**

MOBILE APP FOR CHORE MANAGEMENT IN SHARED LIVING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETER MIKLÁNEK

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, 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: **Miklánek Peter**

Obor: Informační technologie

Téma: **Mobilní aplikace pro správu prací ve sdíleném ubytování**
Mobile App for Chore Management in Shared Living

Kategorie: Uživatelská rozhraní

Pokyny:

1. Prostudujte přístupy k tvorbě jednoduchých mobilních aplikací pro společnou správu informací více uživateli.
2. Navrhněte funkčnost aplikace pro správu prací ve sdíleném bydlení (v jiné malé komunitě).
3. Prototypujte dílčí prvky uživatelského rozhraní a funkčnosti navržené aplikace.
4. Implementujte řešenou aplikaci a testujte ji na uživateli.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 a 4.

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: **Herout Adam, prof. Ing., Ph.D.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 05 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Bakalárska práca sa zaoberá návrhom a implementáciou mobilnej aplikácie pre správu prác v zdieľanom ubytovaní. Aplikácia je určená pre mobilnú platformu Android. Zameriava sa na prívetivé užívateľské rozhranie a pre väčšinu funkcií využíva internetové pripojenie. Užívateľ môže efektívne pridávať nové práce a zdieľať ich so skupinou. Prvá časť práce sa zaoberá architektúrou a komponentami operačného systému Android. Nasledujúce časti sú venované analýze, návrhu a implementácii. Práca ďalej popisuje testovanie výsledného riešenia. Záver je venovaný zhodnoteniu výslednej aplikácie a možnostiam ďalšieho rozvoja.

Abstract

Bachelor's thesis describes the design and implementation of mobile application for core management in shared living. The application is targeted to the Android mobile platform. It focuses on the user interface and for the most functions uses internet access. The user may effectively add new works and share them with a certain group. The first part of this thesis focuses on the architecture and components of the operation system Android. The following parts are dedicated to the analysis, design and implementation. The thesis also describes the testing of the resulting solution. The final part is dedicated to evaluation of the application and possibilities for further development.

Klíčové slová

Android, mobilné zariadenie, aplikácia, Helpmate, užívateľské rozhranie, práca, sociálna skupina, aukcia

Keywords

Android, mobile device, application, Helpmate, user interface, work, shared group, auction

Citácia

MIKLÁNEK, Peter. *Mobilní aplikace pro správu prací ve sdíleném ubytování*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Mobilní aplikace pro správu prací ve sdíleném ubytování

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána prof. Ing. Adama Herouta, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Peter Miklánek
11. mája 2017

Podakovanie

Rád by som sa poďakoval vedúcemu bakalárskej práce, pánovi prof. Ing. Adamovi Heroutovi, Ph.D., za poskytovanie užitočných rád a nápadov, ktoré mi nesmierne pomohli.

Obsah

1 Úvod	3
2 Operačný systém Android	4
2.1 Podporované verzie operačného systému Android	4
2.2 Popis architektúry	5
2.3 Popis komponentov	6
2.4 Grafické rozhranie	8
3 Analýza problému a špecifikácia požiadaviek	11
3.1 Analýza problému	11
3.2 Existujúce obdobné aplikácie	12
3.3 Špecifikácia požiadaviek	13
4 Návrh mobilnej aplikácie	16
4.1 Uživatelské rozhranie	16
4.2 Triedny návrh aktivít	19
4.3 Databáza	20
5 Implementácia	22
5.1 Uživatelské rozhranie	22
5.2 Aktivity a ich implementácia	25
5.3 Synchronizácia a načítanie dát	29
5.4 Jazyková podpora	30
5.5 Alarm a notifikácie	31
6 Testovanie aplikácie	34
6.1 Emulátor v Android Studio	34
6.2 Reálne zariadenia	35
6.3 Zverejnenie aplikácie na Google Play	35
6.4 Stabilita a funkčnosť	36
6.5 Testovanie užívateľského rozhrania	37
6.6 Výsledok testovania	38
7 Záver	40
Literatúra	41
Prílohy	43

Kapitola 1

Úvod

Mobilné zariadenia sú dnes bežnou súčasťou každého dňa väčšiny ľudí. Za pomerne krátku dobu sa zo zložitých komunikačných nástrojov stali zariadenia, ktoré nemá problém ovládať ani malé dieťa. V minulosti sa mobilné zariadenia používali predovšetkým za účelom telefonovania, ale tie časy sú dávno preč. Mnohí z nás nosia vo svojich vreckách počítač malých rozmerov, ktorý nám umožňuje vytvárať profesionálne fotografie a následne ich zdieľať s našimi známymi na sociálnych sieťach. Jedným kliknutím bez problémov spustíme prepracované hry, ktorý ešte pred desiatimi rokmi nezvládol ani najlepší počítač, či bezkontaktné platiť nákupy v obchodoch jednoduchým priložením zariadenia k terminálu.

Operačný systém Android už nie je žiadnym nováčikom a momentálne patrí medzi najrozšírenejšie platformy medzi mobilnými zariadeniami. V čase, kedy som sa rozhodoval vo výbere témy záverečnej práce bakalárskeho štúdia, existoval už obrovský počet aplikácií pre Android, ktoré sa venujú rozsiahlym problematikám od nástrojov zvyšujúcich efektivitu, cez zábavu, až po aplikácie zobrazujúce virtuálnu realitu. Dôvodom, prečo som sa rozhodol vytvoriť aplikáciu pre správu prác v zdieľanom ubytovaní bolo, že podobná aplikácia neexistovala a aj osobne som sa stretol so situáciami, kedy by mne alebo iným osobám aplikácia zjednodušila život.

Cieľom aplikácie je zefektívniť organizáciu a rozhodnutie o pridelení prác v sociálnych skupinách. Aplikácia umožňuje pridávať nové práce a členovia príslušnej skupiny môžu vyjadriť svoj záujem prostredníctvom aukcie. Problémom v podobných situáciách je predovšetkým rozdelenie prác. Aplikácia zobrazuje prehľad aktivity a splnených úloh, na základe ktorého je rozdelenie prác jednoduchšie.

Práca je rozčlenená do siedmych kapitol. Snažil som sa rozpracovať túto problematiku od základu. V prvej kapitole sú popísané základy operačného systému Android, jeho architektúra, základné komponenty, či grafické rozhranie. V ďalšej kapitole sa nachádza analýza problému spojeného s organizáciou prác v sociálnych skupinách a o celkovej špecifikácii požiadaviek. Kapitola číslo štyri približuje návrh mobilnej aplikácie začínajúci pri návrhu užívateľského rozhrania a končiaci pri objektovom návrhu aplikácie. Nasledujúca kapitola sa zaoberá samotnou implementáciou aplikácie pre Android. Okrem iného sa tu nachádza aj synchronizácia a načítanie dát spojené s online databázou. V predposlednej kapitole som sám zvolil a popísal metódy testovania a zverejnil som aplikáciu na Google Play. V závere sa nachádza zhrnutie dosiahnutých výsledkov a priblíženie možného vývoja do budúcnosti.

Kapitola 2

Operačný systém Android

Android¹ je rozsiahla open-source platforma na báze Linuxového jadra, ktorá vznikla hlavne pre mobilné zariadenia (múdre telefóny, tablety, múdre hodinky, ...). Systém Android vyvíja organizácia Open Handset Alliance, ktorej súčasťou sú firmy Google, Intel, NVIDIA, HTC, Samsung a podobne. Informácie uvedené nižšie sú čerpané z knihy [14].

2.1 Podporované verzie operačného systému Android

Číslo, ktoré nám jednoznačne identifikuje verziu systému Android sa nazýva API² level. Na začiatku vývoja každej aplikácie pre Android je potrebné určiť minimálny API level, teda minimálnu verziu systému Android, na ktorej by mala aplikácia plynule bežať v plnej dostupnosti všetkých funkcionalít.

Pokiaľ by bola zvolená menšia verzia API, pokryje sa tým oveľa väčší počet zariadení využívajúcich staršie verzie operačného systému Android, ale obmedzí sa tým počet dostupných funkcií a možností riešenia návrhu aplikácie. Pri rozhodovaní o minimálnej podporovanej verzii Androidu je potrebný kompromis, aby bol API level čo najvyšší, ale zároveň aby pokryl dostatočný počet zariadení s týmto operačným systémom.

Android 4.4 KitKat

Prvou podporovanou verziou operačného systému Android v tejto bakalárskej práci je Android 4.4. API level verzie KitKat je 19, ktorý prináša veľké zmeny hlavne vo vylepšení plynulosti a rýchlosti prostredia oproti predchádzajúcim verziám. Android KitKat umožňuje bezkontaktné platby, detektor pohybu, fullscreen režim pre niekoľko aplikácií a tiež inteligentné vypínanie nepotrebných procesov na pozadí.

Android 5.0 Lollipop

I keď bol Android Lollipop predstavený už pred tromi rokmi, jeho popularita stále neklesá. Jednou z veľkých zmien je nahradenie používaného virtuálneho stroja Dalvik za nový ART³. Hlavným prínosom ART je možnosť kompilovania aplikácií už pri inštalácii. Pri ver-

¹Android – <https://www.android.com>

²API – Application Programming Interface

³ART – Android Runtime

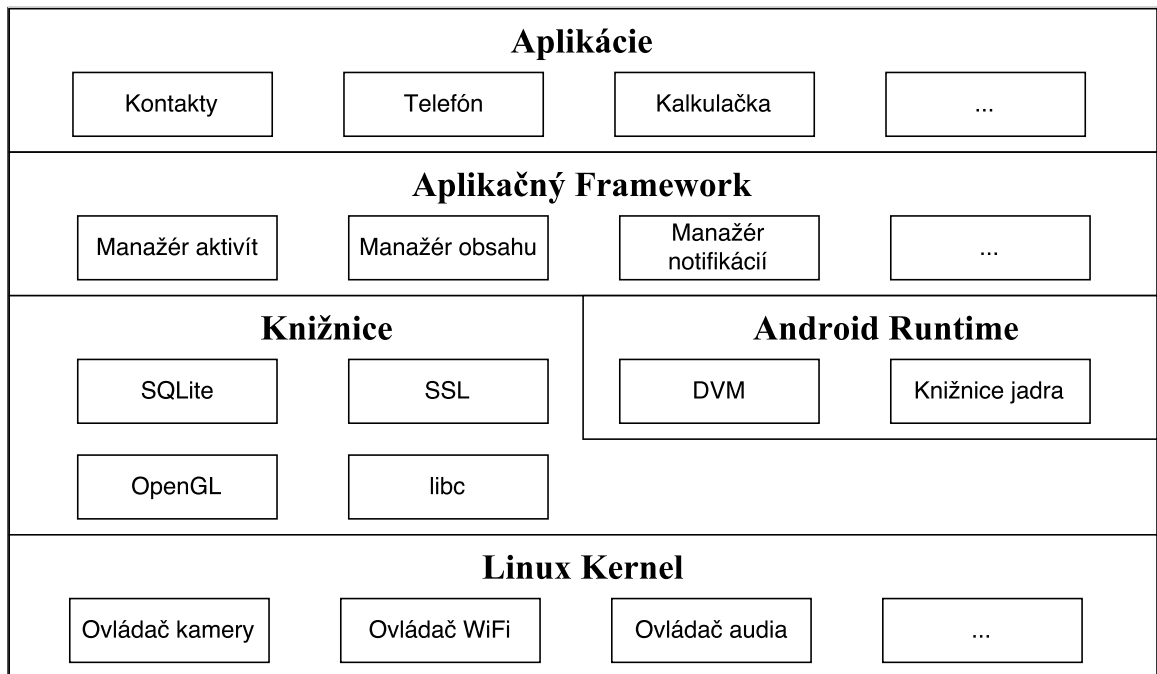
zii Lollipop prešiel aj dizajn veľkou zmenou a to konkrétne prepracovaným užívateľským rozhraním nazývaným Material Design.

Android 7.0 Nougat

Android Nougat je doposiaľ najnovšia verzia s kódovým označením Android N. Nougat prináša radu nových API pre virtuálnu realitu. Za zmienku stojí aj možnosť práce s viacerými oknami, v ktorej dve aplikácie môžu byť každá spustená na jednej polovici obrazovky.

2.2 Popis architektúry

Pre vývoj mobilných aplikácií pre platformu Android je dôležitou časťou oboznámenie sa s jeho architektúrou [10]. Architektúra je rozdelená do štyroch hlavných vrstiev a piatich sekcií znázornených na obrázku 2.1.



Obr. 2.1: Schéma architektúry operačného systému Android

- **Linux Kernel** – Najnižšou vrstvou architektúry Android je upravené jadro populárneho operačného systému Linux. Zabezpečuje správu pamäti a procesov, ovládačov, zabezpečenie systému, ...
- **Knižnice** – Na vrcholu linuxového jadra je sada knižníc, ktoré sú napísané v programovacom jazyku C/C++ a využívajú rôzne funkcie systému, ktoré sú vývojárom dostupné prostredníctvom Android Application Framework. Medzi základné knižnice patria SSL, SQLite, OpenGL, libc, ...
- **Android Runtime** – Každá aplikácia pre Android je samostatný proces využívajúci vlastnú inštanciu virtuálneho stroja DVM⁴. ART používa rovnaký bytekód ako

⁴DVM – Dalvik Virtual Machine

Dalvik, poskytovaný štandardnými .dex súbormi ako súčasť APK⁵ súborov. ART je dostupný od verzie Android 4.4.

- **Aplikačný Framework** – Aplikačný framework je najdôležitejšia vrstva pre developéra, ktorá poskytuje prístup k najrôznejším službám. Tieto služby môžu poskytovať prístup k prvkom užívateľského rozhrania, prístup k aplikáciám bežiacim na pozadí či prístup k dátam z iných aplikácii.
- **Aplikácie** - Vrstva Aplikácie predstavuje aplikácie, ktoré sú implementované za pomoci vrstvy Aplikačný framework. Typickým príkladom sú aplikácie ako Telefón, Email, Správy a podobne.

2.3 Popis komponentov

Pri vytváraní vlastnej aplikácie pre platformu Android je veľkou výhodou možnosť využitia už existujúcich častí iných aplikácii. Aplikácie boli pre tento účel navrhnuté zámerné tak, aby reprezentovali balíky komponentov, ktoré je možné samostatne inštanciovat'. K dispozícii máme štyri základné komponenty.

Aktivity a ich životný cyklus

Pri spustení aplikácie sa užívateľovi zobrazí hlavná trieda, ktorá sa nazýva Aktivita [4]. Aplikácie pozostávajú z ľubovoľného počtu aktivít, ktoré si predávajú potrebné údaje medzi sebou. Primárnym účelom triedy `Activity` je zobrazenie užívateľského rozhrania, prostredníctvom ktorého zachytáva interakciu užívateľa. Pri spustení aplikácie sa ako prvá spustí hlavná aktivita (main), z ktorej je možné spustiť ďalšie. Z toho vyplýva, že usporiadanie aktivít je hierarchické.

Aktivity majú v systéme určitý životný cyklus, pričom nie je vždy jasné, ako dlho daná aktivita pobeží. Životný cyklus aktivít má tri hlavné fázy:

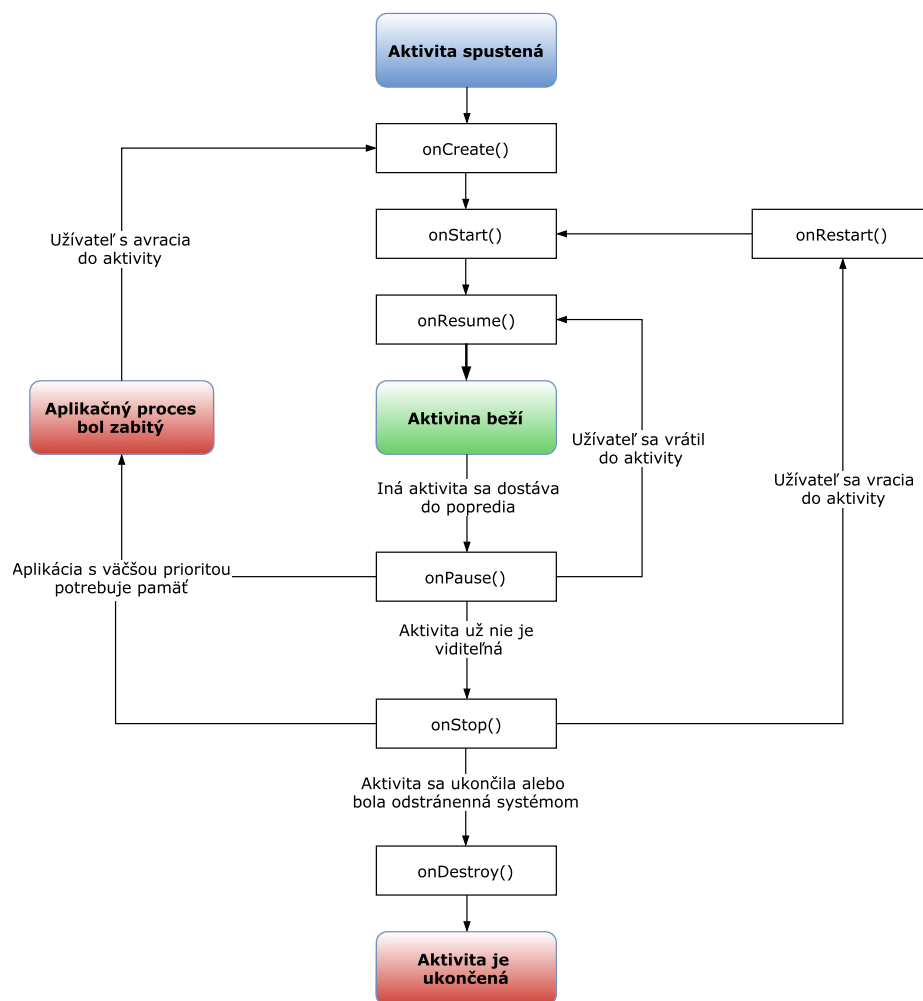
- **Aktivita na popredí** – Aktivita na popredí má fokus, čo znamená, že je aktuálne zobrazená na displeji.
- **Pozastavená aktivita** – Pozastavená aktivita stráca fokus, ale je stále v pamäti.
- **Zastavená aktivita** – Zastavená aktivita je úplne prekrytá inou aktivitou a je sústavne uložená v pamäti až do chvíle, kedy nastáva nedostatok pamäti. Vtedy dochádza k jej vymazaniu a uvoľneniu miesta v pamäti.

Životný cyklus aktivít je znázornený na obrázku 2.2.

Services

Services [11] (v preklade Služby) sú komponenty aplikácie, ktoré neposkytujú užívateľovi užívateľské rozhranie ale realizujú dlho trvajúce operácie a operácie na pozadí. Operáciám realizovaných dlhšiu dobu umožňujú prevádzať služby asynchrónne. Príkladom je komponenta slúžiaca k sieťovej prevádzke, ktorá prijíma a odosiela dáta na pozadí.

⁵APK – Android Package Kit



Obr. 2.2: Schéma životného cyklu aktivity. Pri vytváraní mobilnej aplikácie je potrebné túto schému poznať, aby výsledná aplikácia správne pozastavovala, či ukončovala aktivity. Pri jej nesprávnom použití dochádza k nedostatku pamäti.

Broadcast Receivers

Broadcast Receiver [6] je komponenta slúžiaca k počúvaniu broadcastových oznámení na pozadí a reagovaniu na udalosti odohrávajúce sa na zariadení. Rovnako ako služby, ani Broadcast Receiver nie sú užívateľsky viditeľné. Bežným príkladom na predstavenie môže byť prijatie emailu alebo SMS správy. Pri doručení takéhoto typu správy informuje užívateľa operačný systém pomocou notifikácie.

Content providers

V preklade poskytovateľa obsahu [7] riešia problémy v správe dát, problémy s databázou a predovšetkým umožňujú ukladať a zdieľať dáta medzi aplikáciami. Aplikácie, ktoré nesú označenie poskytovateľa obsahu prístupujú dáta iným aplikáciám.

Intent

Zdieľanie (v originále nazývané Intent [8]) sa používa pre komunikáciu medzi jednotlivými komponentami. Intent oznamuje systému zámer aktivity pomocou inštancie triedy `Bundle`, ktorá uchováva potrebné údaje. Intent Filter definuje typy zámerov danej aplikácie a nachádza sa v súbore Android Manifest, ktorého popis sa nachádza nižšie.

Android Manifest

Pre správne fungovanie aplikácie je potrebné mať prostriedok, ktorý operačnému systému oznámi, aké komponenty sú k dispozícii. Práve pre tento účel slúži XML súbor pomenovaný Android Manifest⁶. V súbore Android Manifest sa uvádza aj zoznam povolení k prístupu, ktoré sú potrebné pre plnohodnotný beh určitej aplikácie. Jednoduchým príkladom je povolenie prístupu ku fotoaparátu zariadenia.

2.4 Grafické rozhranie

Grafické rozhranie (GUI⁷) v aplikácii reprezentuje prezentačnú vrstvu vizuálnej komponenty, pričom umožňuje ovládať zariadenie pomocou interaktívnych obrazových prvkov. Nasledujúca sekcia popisuje základné grafické prvky, ktoré sú potrebné pri návrhu GUI operačného systému Android. Informácie sú čerpané z knihy [16]

Views

Views [12] (v preklade pohľady), taktiež známe ako widgety, sú základnou triedou pre tvorbu GUI. Všetky ovládacie prvky užívateľského rozhrania sú od nich odvodené. Môžu byť členené do View Groups (skupiny pohľadov), ktoré obsahujú viacero pohľadov ako je vidieť na obrázku 2.3.

Najčastejšie využívané prvky pri tvorbe GUI sú napríklad:

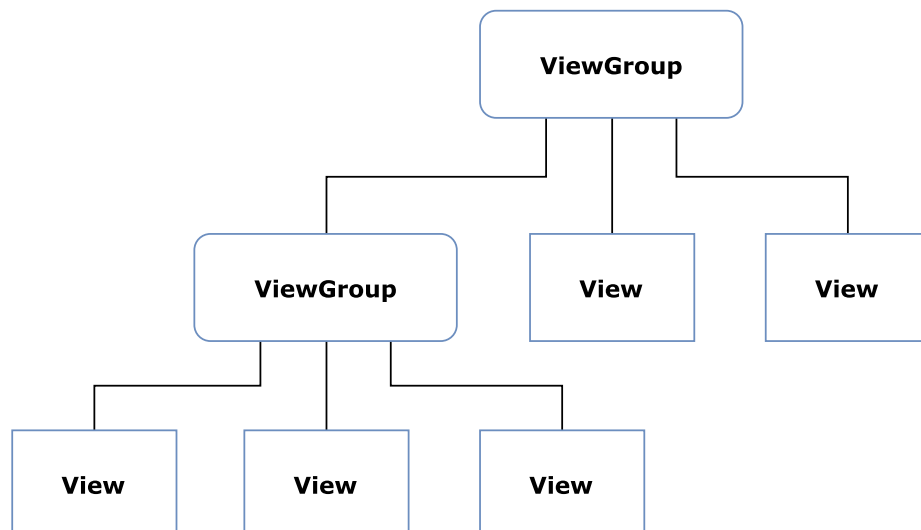
- **TextView** – zobrazuje textové výpisy
- **EditText** – na rozdiel od TextView umožňuje aj úpravu zobrazeného textu
- **Button** – štandardné tlačítko reagujúce pri dotyku
- **CheckBox** – prvok v tvare štvorca, ktorý slúži ako zaškrtačacie políčko
- **ProgressBar** – prvok, ktorý sa typicky zobrazuje počas načítavania

Kontajnery

Kontajnery sú odvodené od triedy View Groups a reprezentujú rozmiestnenie prvkov užívateľského rozhrania, nazývané tiež layouts. Poznáme päť základných layoutov:

⁶Android Manifest – <https://developer.android.com/guide/topics/manifest/manifest-intro.html>

⁷GUI – graphic user interface



Obr. 2.3: Hierarchia objektov View Groups. View Group môže obsahovať ďalšie View Groups, alebo Views-

- **LinearLayout** – umožňuje rozmiestniť vnorené prvky, pričom ich orientácia môže byť horizontálna alebo vertikálna pomocou parametra `orientation`. Ďalšie základné parametre definujú prioritu vnorenia alebo centrálny bod zarovnania v rámci layoutu.
- **RelativeLayout** – narozdiel od LinearLayoutu umožňuje relatívne rozmiestnenie vnorených prvok vďaka unikátne zvoleným identifikátorom daných prvok.
- **FrameLayout** – najčastejšie obsahuje iba jeden vnorený prvok, pretože pri väčšom počte sa prekrývajú.
- **TableLayout** – ako už názov predpovedá, využíva sa na organizovanie prvkov užívateľského rozhrania do formy tabuľky.
- **GridLayout** – posledným základným layoutom je GridLayout, ktorý sa využíva pre zobrazenie prvkov vo forme matice.

Material Design

Material Design [9] je komplexná príručka pre vizuálny interaktívny dizajn naprieč platformami a zariadeniami. Material Design bol predstavený s verziou Androidu 5.0 (API level 21) v roku 2014 na Google konferencii. Google uvádza, že Material Design je založený na princípe papiera a atramentu, čo znamená, že rovnako ako klasický papier, má plochy a hrany, ktoré odrážajú svetlo, vrhajú tieň a už aj samotný názov *Material* má reprezentovať návrh, ktorý je založený na myšlienke reálneho sveta. Výhodou Material Designu oproti reálnemu papieru je možnosť pohybovať sa, meniť tvar a veľkosť, a podobne.

Material Design⁸ priniesol viaceré rozsiahle novinky, ktoré si stručne popíšeme v nasledujúcej časti.

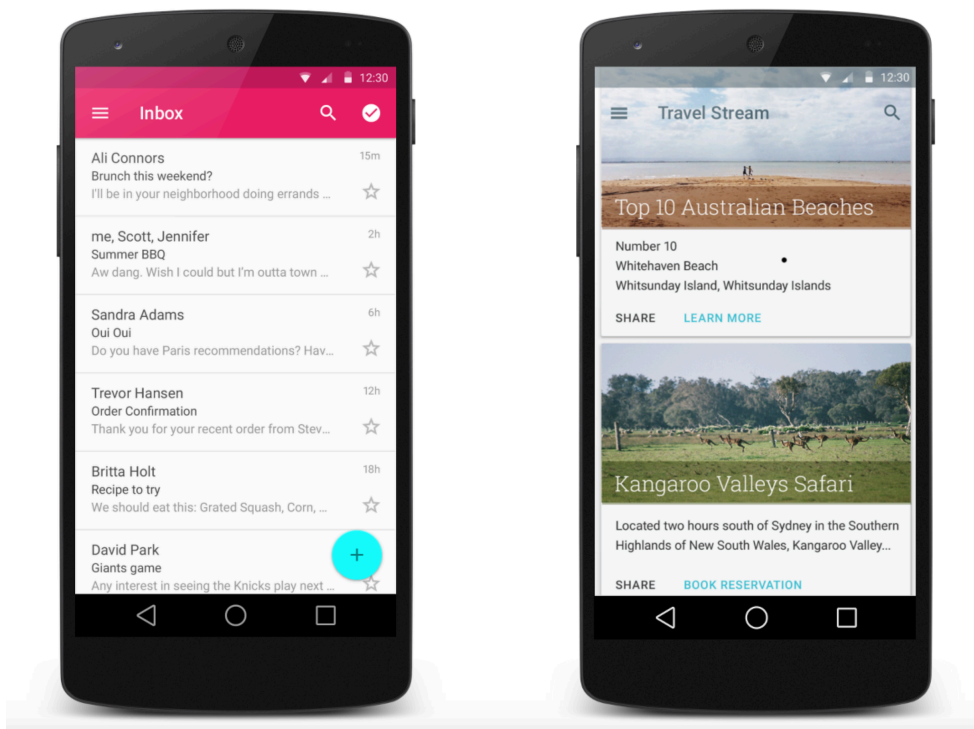
⁸Material Design – <https://material.io>

Material Theme

Material Theme poskytuje nový štýl pre aplikácie a systémové widgety, kde umožňuje nastaviť farebnú paletu a východiskové animácie pre spätnú väzbu dotyku a prechody medzi aktivitami.

Lists and Cards

V preklade Zoznamy a Karty prinášajú dva nové widgety pre zobrazenie zoznamov a kariet v štýle Material Dizajnu. Sú to RecyclerView a CardView. Znázornenie tejto novinky je na obrázku 2.4.



Obr. 2.4: Na obrázku vľavo znázornený RecyclerView a vpravo CardView. Tieto novinky priniesli rozšírenú funkcionality oproti predchodcom a ich popis sa nachádza nižšie.

- RecyclerView je widget, ktorý je rozšírená verzia základného ListView a prináša rôzne typy rozvrhnutia a vyšší výkon.
- CardView zobrazuje dôležité informácie vnútri karty, pričom majú vzbudzovať jednotný dojem a vzhľad.

Tabs

Tabs (záložky) sa typicky nazývajú *Tab Layout* a prinášajú nový štýl prechodu medzi sebou s novou dizajnovou podobou. Prepínanie medzi jednotlivými záložkami už nieje len pomocou kliknutia, ale prechod je umožnený aj jednoduchým potiahnutím obrazovky zo strany na stranu.

Kapitola 3

Analýza problému a špecifikácia požiadaviek

Aké problémy by mala aplikácia riešiť a aké vlastnosti by mala spĺňať? Odpovede na tieto ale aj iné otázky sú rozobraté v nasledujúcich sekciách.

3.1 Analýza problému

Väčšina z nás býva alebo niekedy bývala v zdieľanom ubytovaní, či už to boli internáty alebo byty so spolubývajúcimi. Pokiaľ v takýchto priestoroch býva viac ako jedna osoba, nastávajú rozpory týkajúce sa rozdelenia spoločných prác. Je potrebné pravidelne umývať riad, podlahu, kúpeľňu, raz za čas umyť okná a veľa ďalších povinností. Častým problémom býva, že spolubývajúcim nezáleží na poriadku tak, ako iným. Je potrebné ich motivovať, aby si svoje povinnosti splnili vtedy, kedy je to potrebné. Lenže nájsť vhodný a účinný spôsob motivácie nie je jednoduché. Bežným riešením býva tabuľka rozpisu služieb, ktorá nanešťastie býva časom neprehľadná a taktiež pri nesplnení úlohy dotyčnému nehrozia žiadne následky.

Pri analýze problémov spojených so zdieľaným ubytovaním som si uvedomil dôležitý fakt. Prehľadne organizovať práce a motivovať ľudí k splneniu istých povinností môže byť starosť, s ktorou sa stretávame v oveľa viac prípadoch. Rôzne spory nastávajú v ľubovoľných sociálnych skupinách, ako napríklad skupiny zaoberajúce sa školským projektom, kolektív kolegov v práci so zdieľanými priestormi alebo garážová hudobná kapela. Taktiež sa tieto práce nemusia týkať len upratovania, ale často krát je potrebné niečo dokúpiť, priniesť, odnieť, zariadiť, a podobne.

Zaujímavým zistením bola navyše nepotrebná zmena pôvodnej myšlienky a špecifikácie aplikácie, ktoré sú spomenuté nižšie.

Cielová skupina

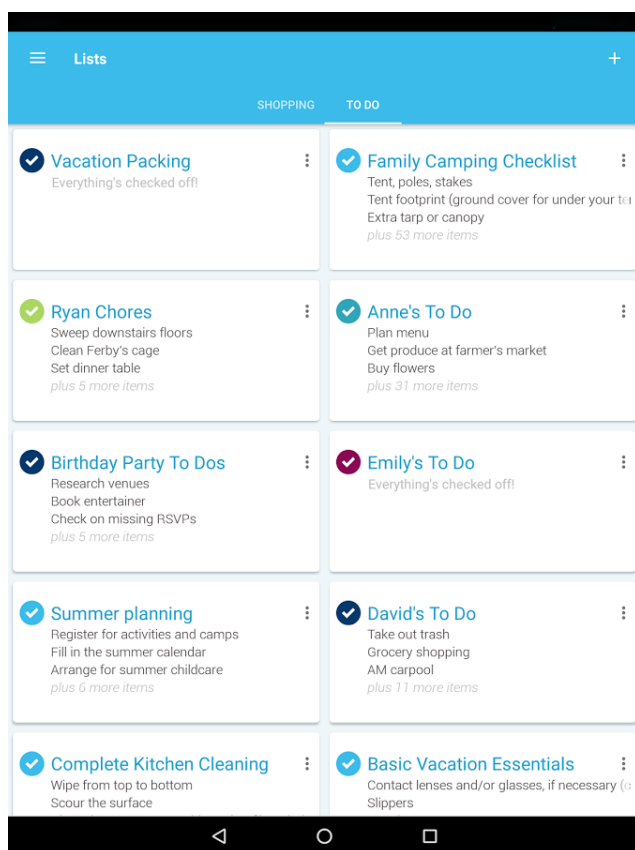
Za pomoci analýzy problému nie je zložité určiť cieľovú skupinu užívateľov. Pôvodne ňou mali byť predovšetkým osoby žijúce v zdieľanom ubytovaní. Avšak po prehodnotení skutočnosti, že základná myšlienka sa dá jednoduchým spôsobom rozšíriť, sa cieľovou skupinou stáva každý, kto potrebuje vhodným spôsobom organizovať rôzne práce v ktorejkoľvek sociálnej skupine a inšpirovať jej členov k splneniu povinností.

3.2 Existujúce obdobné aplikácie

V nasledujúcich riadkoch sa nachádzajú tri existujúce aplikácie venujúce sa obdobnej problematike. Pri každej z nich sa nachádzajú mnou zhodnotené pozitíva a negatíva. Záver odvodený zo sekcii 3.1 a 3.2 bude použitý na vytvorenie špecifikácie požiadaviek na aplikáciu navrhnutou v tejto práci.

Cozi Family Organizer

Aplikácia Cozi Family Organizer¹ je šikovný nástroj pre organizovanie rôznych povinností v rodine. Užívateľské rozhranie je jednoduché, prehľadné a ponúka viacero praktických funkcií, ktoré sú v rodinnom kruhu užitočné. Užívateľom poskytuje zdieľaný kalendár, nákupný zoznam, zoznam prác, či úložisko receptov. Aplikácia však neponúka prehľad spojený so splnenými povinnosťami v minulosti a taktiež nenapomáha k motivovaniu splnenia daných úloh. Ukážku Cozi Family Organizer nájdete na obrázku 3.1.



Obr. 3.1: Screenshot z aplikácie Cozi Family Organizer. Na ukážke je vidieť jednotný dizajn aj pri väčšej uhlopriečke. Popis pri jednotlivých položkách obsahuje zoznam nesplnených úloh v danej kategórii, čím rozhodne sprehladní zoznam. Ak by úlohy v tomto prípade neboli kategorizované, zoznam by bol príliš dlhý.

¹Cozi Family Organizer – <http://www.cozi.com>

Band

Band² sa zaoberá organizovaním rozličných skupín, ako napríklad tímy/ kluby, práce/ projekty, herné klany/ organizácie a podobne. Z týchto príkladov je vidieť, že Band je pomerne všestranná aplikácia, ktorá sa na rozdiel od Cozi Family Group nezaobera len oblasťou ohľadom rodinných starostí. Aplikácia však pripomína sociálnu sieť, v ktorej je možné pridávať príspevky a dokonca si medzi sebou písať správy. Náhľad na aplikáciu Band sa nachádza na obrázku 3.2.

Flow Tasks

Posledným obdobným riešením je aplikácia Flow Tasks³. Tá, rovnako ako Band a Cozi Family Organizer, sa predovšetkým zaoberá plánovaním projektov a ich organizáciou. Pozitívnou vlastnosťou je možnosť pripojenia rôznych formátov súborov k danej úlohe. Avšak negatívom, podľa môjho úsudku, je striktné pridelovanie práce členovi tímu, ktoré môže mať za následok jej nesplnenie v stanovenom termíne. Pokiaľ by užívatelia mali príležitosť rozdelenia si prác aj na základe predošlých úspechov či neúspechov, mohlo by to mať výrazný vplyv na splnenie dokončenia. Screenshot je k dispozícii na obrázku 3.2.

3.3 Špecifikácia požiadaviek

Sekcia špecifikácia požiadaviek zhrňuje poznatky nadobudnuté z predchádzajúcich sekcií analýzy problému a analýzy existujúcich obdobných riešení.

Motivácia užívateľov

Táto časť sa zaoberá jednou z najdôležitejších častí požiadaviek na aplikáciu. Dlhú som premýšľal nad spôsobom inšpirácie budúcich užívateľov pre splnenie náležitých povinností. Osvedčeným spôsobom, ktorý je zavedený už stovky rokov, sa ukázala byť technika odmena a trest. Odmena za aktivnosť, trest za pasívnosť. Táto praktika je užitočná a prináša výborné výsledky. V aplikácii nie je žiadna konkrétna odmena, ani trest. Údaje sú zbierané v intervaloch dlhých jeden mesiac, pričom má každý prehľad o tom, kto a koľko toho spravil. Na základe týchto informácií je možné rozdeliť spoločné náklady v skupinách, poskytnúť odmenu za snahu atď. Toto rozhodnutie je prenechané samotným užívateľom a je len na nich, ako s nazhromaždenými dátami naložia.

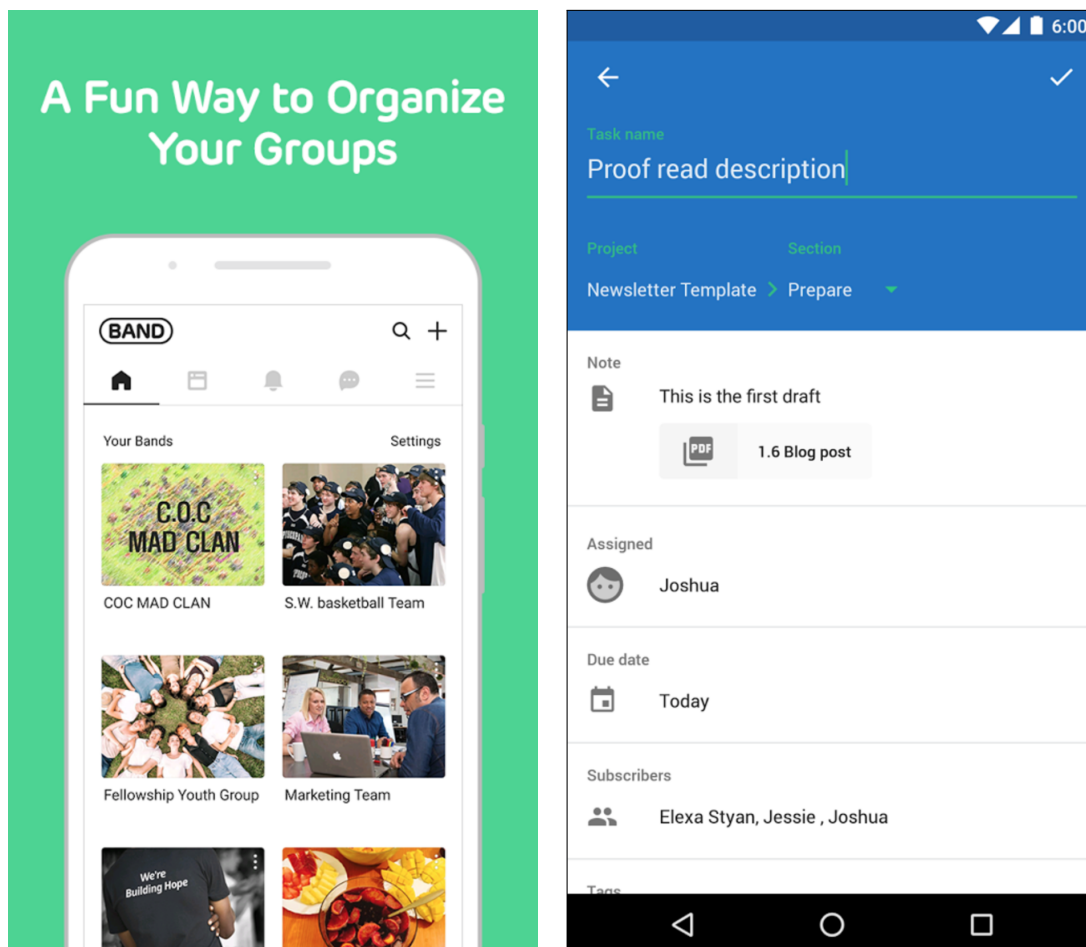
Obrátená aukcia

Aukcia alebo dražba je predaj, pri ktorom sa akákoľvek vec (živá alebo neživá) predáva väčšiemu počtu záujemcov. Dražený objekt získa ten, kto zaň ponúkne najviac. Ako už z nadpisu vyplýva, pri obrátenej aukcii platí pravidlo vydraženia za najnižšiu ponuku.

Pri pridaní novej činnosti sa na základe času dĺžky práce vypočíta doba, po ktorú bude daná činnosť v stave draženia. Tento stav znamená časový interval, kedy môže každý člen v aktuálnej skupine vyjadriť svoj záujem a to v troch prípadoch. Pri prvom prípade pridá

²Band – <http://band.us/home>

³Flow Tasks – <https://www.getflow.com>



Obr. 3.2: Screenshoty z aplikácie z ľavej strany Band, Flow Tasks. Band zobrazuje zoznam skupín reprezentovaných aj obrázkom, pričom je vidieť na ukážke väčší počet záložiek, ktoré poskytujú širokú funkčnosť. Náhľad na aplikáciu Flow Tasks zobrazuje pridanie novej úlohy, kde je možné nahráť aj súbor vo formáte PDF.

svoju ponuku k vybratej úlohe, ktorá je reprezentovaná bodovým ohodnotením nazývaným kredit. Pri druhom smie člen skupiny označiť momentálnu povinnosť stavom „nemám záujem“. Posledným prípadom je jednoducho situácia nevyjadrenia záujmu a rozhodnutie neinformovať tak ostatných členov skupiny o záujme.

Akým spôsobom prebieha vyhodnotenie aukcie vytvorené v tejto aplikácii? Aukcia sa k danej činnosti vyhodnocuje na základe vyjadrenia všetkých príslušníkov skupiny. Ak sa do konca časového intervalu aukcie nevyjadria všetci príslušníci, práca pripadne tomu, kto za ňu ponúkol najmenej. Žiaľ, niekedy o danú prácu nemusí mať nikto záujem. Vtedy dochádza k informovaniu užívateľov o nezáujme k danej práci a je možné prácu vymazať, alebo opätovne pridať.

To Do listy

Po otvorení a úspešnom prihlásení/registrovaní do aplikácie sa užívateľovi zobrazí hlavné okno obsahujúce 3 zoznamy. Primárnym je zoznam obsahujúci všetky práce. Jedna položka

v zoznamu musí obsahovať potrebné údaje pre informovanie. Druhým listom, nie menej dôležitým, je výpis úloh, ktoré má daná prihlásená osoba na starosti. Tu sa úlohy zobrazujú po pridelení práce na základe aukcie. V poslednom súpise sa nachádza história už dokončených prác.

Dodatočné požiadavky

V záverečnej časti tretej kapitoly sa nachádzajú dodatočné požiadavky. V nasledujúcich riadkoch je stručne popísaná pri každom bode jedna vlastnosť dôležitá pre správny a efektívny beh aplikácie.

- dostupnosť dát online (databáza dostupná prostredníctvom internetového pripojenia)
- graf zobrazujúci štatistické informácie o prograse v aktuálnej skupine
- jeden užívateľ členom viac ako jednej skupiny
- jednoduché vytváranie nových skupín
- rýchle pridávanie nových úloh

V podkapitole špecifikácia požiadaviek boli popísané teoretické princípy fungovania aplikácie pre rozličné skupiny ľudí. Taktiež sa tu nachádza úvodná špecifikácia funkcionality, ktorá bude upresnená a rozšírená v nasledujúcich kapitolách.

Kapitola 4

Návrh mobilnej aplikácie

Vďaka analýze podobných riešení som získal vlastnosti, ktoré by aplikácia mala obsahovať a naopak tie, na ktoré by som si mal dávať pozor. Aplikácia by nemala ponúkať len bezchybnú funkčnosť, ale predovšetkým intuitívne užívateľské rozhranie jednoduché pre pochopenie na prvý pohľad.

Kapitola Návrh mobilnej aplikácie popisuje návrh GUI, objektový návrh a taktiež výber databázy so zdôvodnením.

4.1 Užívateľské rozhranie

Návrh užívateľského rozhrania [1] je dôležitou súčasťou väčšiny systémov, či už sa jedná o mobilné, webová aplikácie alebo operačné systémy. Prvým krokom pri návrhu by mal byť nákres našich predstáv na papier. Takýto náčrt sa nazýva **Wireframe**. Wireframe sa dá jednoducho prekresliť alebo zahodiť do odpadkového koša a začať znovu. Až keď je wireframe dostatočne premyslený, prichádza na rad jeho prenesenie do virtuálnej podoby. Za týmto účelom som si zvolil internetový program drawio¹.

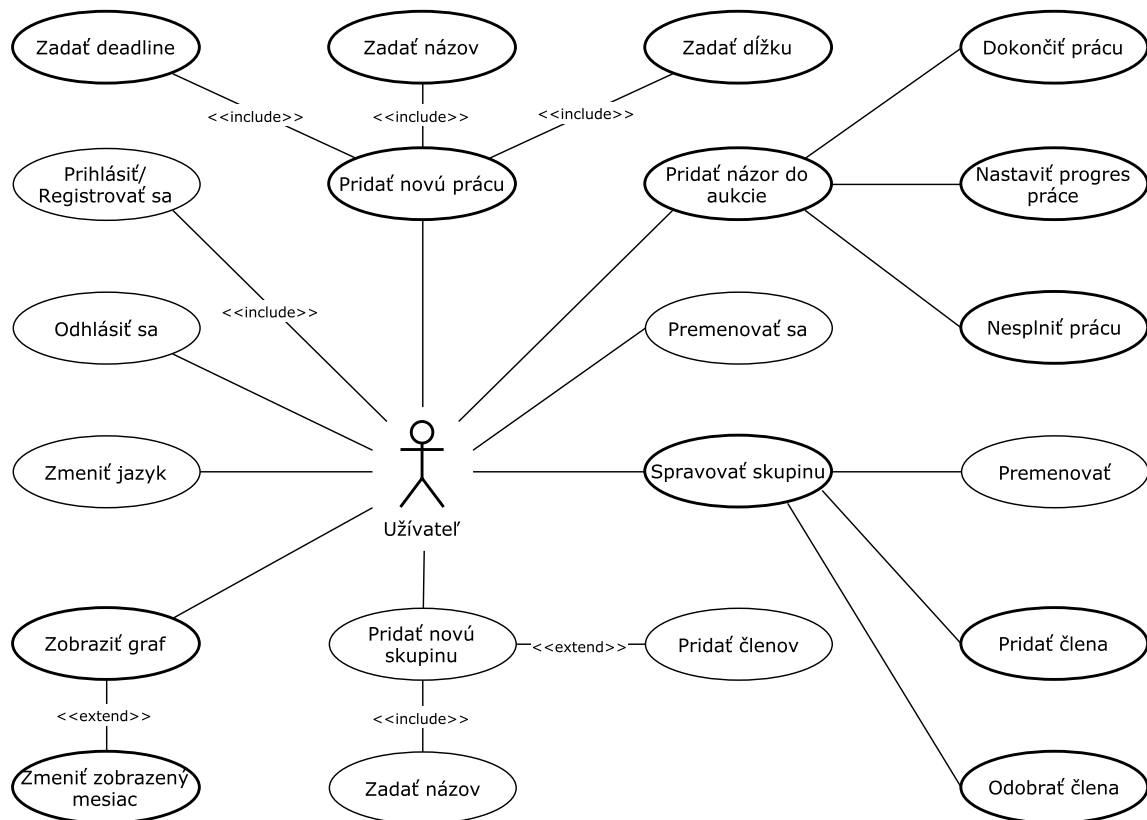
Drawio ponúka užitočné vlastnosti, pomocou ktorých je jednoduché navrhnuť ľubovoľný graf a aj mobilnú aplikáciu. Jediné, čo je pri tom potrebné, je vyhľadať v ich zozname grafické komponenty a umiestniť ich kurzorom myši na editačnú plochu programu.

Prípady užitia

Pre vyjadrenie prípadov užitia bol využitý diagram prípadov užitia [15], tiež známy ako **use case diagram**, ktorý je navrhnutý jazykom UML. Pred rozhodnutím rozloženia grafických komponentov je dôležité si položiť otázku, čo má náš systém vedieť. Tento diagram nám ukazuje práve odpoveď na túto otázku, a nie to, čo bude robiť. Obrázok 4.1 zobrazuje diagram prípadu užitia aplikácie pre správu zdieľaných prác, kde aktérom je bežný užívateľ.

Úkon, ktorý bude najfrekvencovanejšie používaný, je na obrázku 4.1 zobrazený priamo nad aktérom. Činnosť *Pridať novú prácu* môže byť využitá v bežnej prevádzke niekoľko krát za deň. Vyplnenie troch parametrov v nej je povinné a ich zadanie musí byť z toho dôvodu rýchle a jednoduché. Naopak pri *Pridať novú skupinu* sa nachádza aj nepovinný údaj, pretože minimálny počet členov skupiny je jeden.

¹ Drawio – <https://www.draw.io>



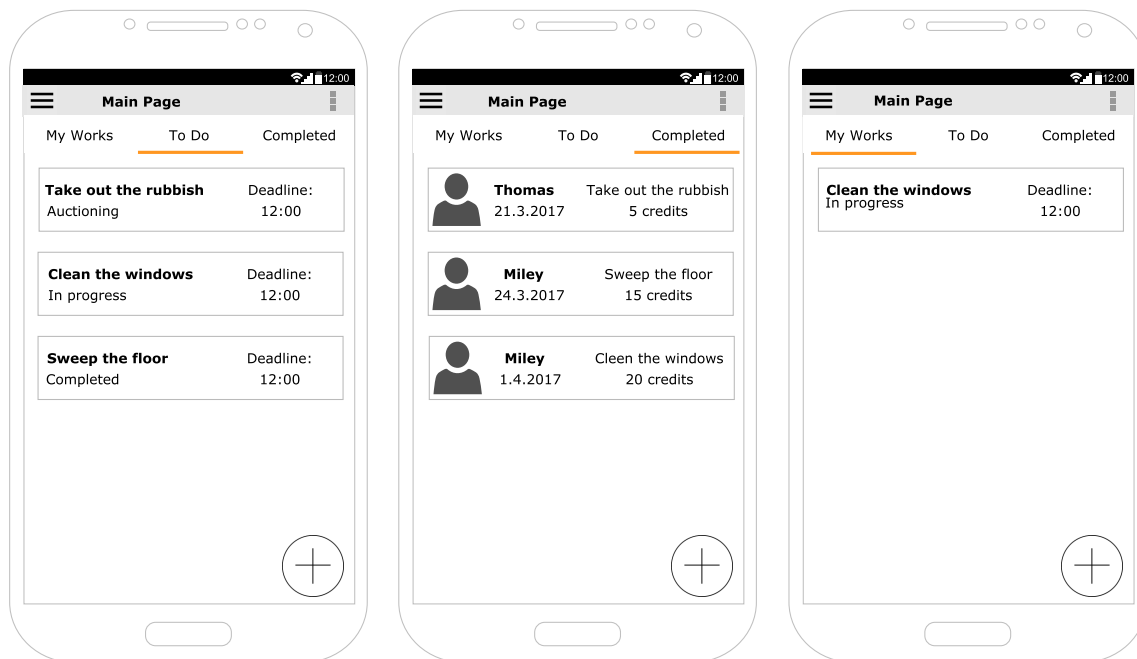
Obr. 4.1: Diagram prípadov užívania. Položky znázornené na obrázku s hrubším okrajom vyjadrujú úkony, ktoré budú užívatelia pravdepodobne využívať aj niekoľko krát za deň. Naopak, s tenším okrajom sú znázornené úkony menej často využívané.

Hlavná obrazovka aplikácie

Obrazovka sa vždy zobrazí každému ako prvá pri prihlásení alebo zaregistrovaní sa do aplikácie. Rozhodol som sa ju rozdeliť na tri záložky **Tab**. Primárnou je záložka *To Do* a zobrazuje zoznam aktuálnych prác v skupine, či už v stave draženia, vykonávania alebo krátky čas aj po dokončení. Sekundárnymi záložkami sú *My Works* a *Completed*. Na prvej je zobrazený list povinností, ktoré boli prihlásenému užívateľovi pridelené a pri druhej sú už dokončené práce v skupine. Úkon pridať novú prácu, spomínaný vyššie, je potrebné ukotviť na trojicu týchto záložiek na známu pozíciu, aby mal k nej každý okamžitý prístup.

V aplikáciach známych ako kontakty, email či správy sa používa tlačítka umiestnené v pravom dolnom rohu a ľudia si už na toto umiestnenie zvykli [17]. Tlačítka pre pridanie novej práce by preto nemalo byť nikde inde, ako na tom mieste.

Jeden riadok zoznamu obsahuje vždy len tie najnutnejšie informácie. V záložke, reprezentujúcej zoznam všetkých prác. Je to názov, status a koniec aukcie. Vrchná lišta obsahuje 2 klasické Android widgety. Menu je v ľavom hornom rohu a dodatočné nastavenia v tom pravom. Nápis *Main Page* bol neskôr pozmenený za iným zámerom, ktorý bude spomenutý v kapitole 5.2.



Obr. 4.2: Návrh hlavnej obrazovky aplikácie. Na všetkých záložkách sa nachádza zoznam s obsahom príslušným k danej záložke a tlačítko so znakom plus reprezentuje pridanie novej práce.

Pridanie novej práce a Detaily práce

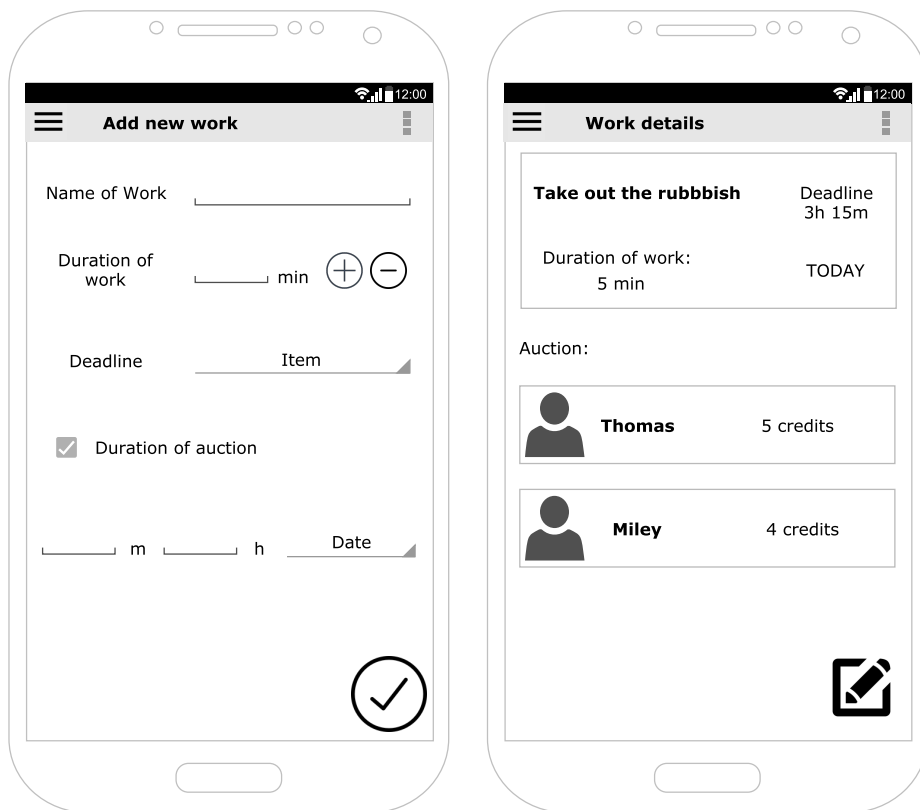
Z hlavnej stránky sa jedným kliknutím dá dostať na tieto dve aktivity. Obrazovka znázornená na obrázku 4.3 v ľavo sa spustí pri stlačení tlačítka plus a obrazovka v pravo po rozkliknutí konkrétnej úlohy. Ako možno vidieť, každá z nich je zložitejšia a skladá sa z viacerých častí. Návrh preto nebol jednoduchý a pre maximalizovanie dodržania prehľadnosti musí byť zachovaný Material Design. Opätovne symboly v pravom dolnom rohu evokujú funkčnosť skrývajúcu sa za daným tlačítkom. Rozhodnutie o ich použití nebolo náhodné, ale vybrané na základe krátkeho dotazníku. Pre pridanie názoru alebo ponuky do aukcie k práci som uvažoval nad použitím ikony kladiva známeho zo súdnej siene. Avšak osobám vyplňujúcim dotazník by to pripomínalo skôr dokončenie aukcie a výsledný verdikt.

Farba a názov

Dodatočný nadpis návrhu užívateľského rozhrania sa zaoberá vhodným výberom farby a názvu pre aplikáciu a jej názvom.

Zahraničné štúdia [3] zistili zaujímavý fakt. Jednotlivé farby u ľudí budia rôzne emócie, pocity a dojmy. Napríklad modrá je farbou vyváženosti a pohody. Pri kapitolách a ich podsekciami vyššie som analyzoval problém spojený s motivovaním ľudí dokončiť si svoje povinnosti. Rozhodol som sa v aplikácii použiť oranžovú farbu so zámerom prínosu myšlienky vykonania činnosti. Táto farba symbolizuje energiu a prináša dojem, pri ktorom človek získava väčší podnet pre aktivnosť a pohyb. Nejedna aplikácia túto vlastnosť využila tiež a spravidla to bývajú aplikácie zamerané na cvičenie, beh, cyklistiku, plávanie, ...

Poslednou časťou je názov. Ani ten som nenechal na náhodu. Uvažoval som nad názvom Flatmate, Share Deals, Help Out, AddDone, Make It Happen, ale žiaden z nich mi neprišiel vhodný. Nakoniec som sa rozhodol použiť **Helpmate**² pre pomenovanie aplikácie. V preklade toto označenie vyjadruje pomocníka alebo priateľa, ktorý Vám pomôže. Na šťastie bolo meno Helmate v obchode Google Play voľné a definitívne som ho tak pomenoval.



Obr. 4.3: Návrh UI z ľava pridanie novej práce, detaily práce. Na ľavej obrazovke v pravom dolnom rohu vyjadruje tlačítko dokončenie pridávania novej práce a na pravej obrazovke pridanie novej ponuky k aukcii.

4.2 Triedny návrh aktivít

Počas návrhu užívateľského rozhrania som získal predstavu, aké aktivity je potrebné implementovať. Každá aktivita reprezentuje práve jednu triedu. Z návrhu užívateľského rozhranie (UI³) vychádzajú tri aktivity, avšak tento počet ani z ďaleka nepostačí pre implementáciu aplikácie Helpmate. Je potrebné myslieť na nastavenia aplikácie, vykresľovanie grafu a podobne. Nasledujúce riadky popisujú návrh tried, ktoré reprezentujú všetky potrebné aktivity.

²Helpmate – <https://play.google.com/store/apps/details?id=com.flatmate.flatmate>

³UI – User Interface

MainActivity, AuctionActivity a AddNewWorkActivity

Hlavnú obrazovku aplikácie reprezentuje trieda `MainActivity`, pridanie novej práce reprezentuje trieda `AddNewWorkActivity` a aukciu konkrétnej práce reprezentuje `ActionActivity`. Návrh a zmysel týchto aktivít bol popísaný v sekcii 4.1.

GraphActivity

Aby mohol mať užívateľ dostatočný prehľad nad výsledkami vo svojich skupinách, je potrebné vykresliť graf v samostatnej aktivite. Uvažoval som nad stĺpcovým grafom, ale atraktívnejšou a prehľadnejšou voľbou je koláčový graf. V grafe bude v každej časti napísané meno a taktiež percentuálny podiel vykonanej práce v danom mesiaci.

CreateNewGroupActivity a GroupInfoActivity

Pre vytvorenie novej skupiny a zobrazenie informácií k danej skupine som sa rozhodol vytvoriť dve samostatné aktivity. K prvej z nich je možné pristúpiť cez postranné menu. Pre vytvorenie novej skupiny je potrebné zadať meno skupiny a v ďalšom kroku pridať ľubovoľný počet jej členov. Informácie o danej skupine slúžia predovšetkým na zobrazenie jej členov a premenovanie danej skupiny. Tvorca skupiny navyše môže rozhodovať o pridaní nových členov a taktiež o vymazaní jednotlivých členov, pričom má každý možnosť vymazať samého seba z danej skupiny. Hoci bol samotný návrh aktivít jednoduchý, implementácia bola už o niečo komplikovanejšia. Viac sa o tom nachádza v sekcii 5.2.

SignInActivity a LogInActivity

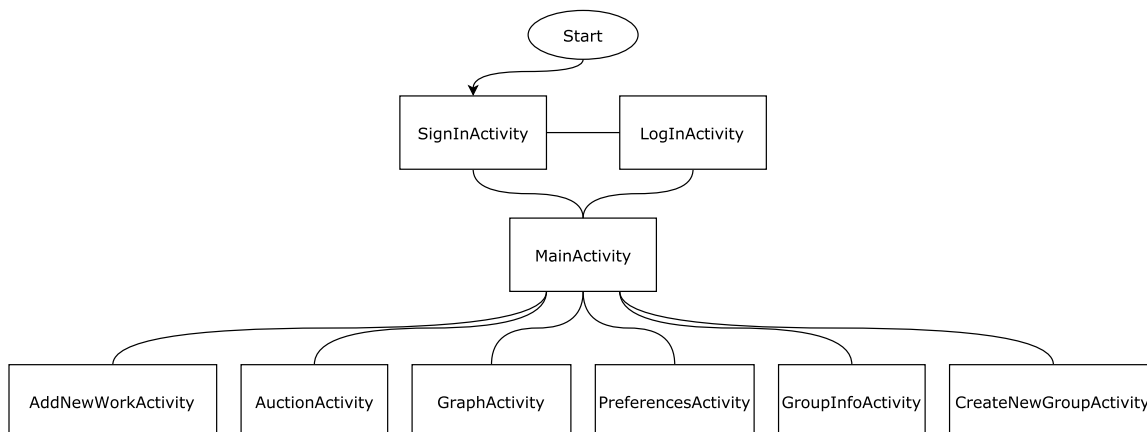
Prihlásenie a registrácia nových užívateľov je samozrejmosťou pri podobných aplikáciach, ako je Helpmate. Pre každý z týchto úkonov existuje samostatná aktivita, aby nevznikli problémy, pri ktorých by sa užívateľ zaregistroval namiesto prihlásenia. Pri prihlásení je bežné zadanie emailu a hesla. Registrácia je doplnená len o vyplnenie mena užívateľa, čím je veľmi jednoduchá a nezaberá zbytočne veľa času. V dnešnej dobe je pri týchto úkonoch už potrebou spolupracovať so spoločnosťami Facebook či Google, a preto je možné sa v aplikácii prihlásiť aj pomocou nich. Implementáciu opisujem v sekcii 5.2.

PreferencesActivity

Poslednou zmienanou aktivitou je `PreferencesActivity`, ktorá slúži pre nastavenia aplikácie. Nastavenia ponúkajú predovšetkým premenovanie daného užívateľa, či zmenu jazyka.

4.3 Databáza

Aby Helpmate spĺňal všetky predtým stanovené vlastnosti, je potrebné zdieľanie dát aplikácie cez internetové pripojenie. Existuje veľa druhov databáz, ktoré náležitú špecifikáciu spĺňajú. Ja som sa rozhodol využiť službu Firebase [5].



Obr. 4.4: Diagram aktivít aplikácie

Firestore

Firestore⁴ je webová a mobilná platforma s nástrojmi a infraštruktúrou. Jej cieľom je pomôcť vývojárom vytvárať vysoko kvalitné aplikácie a nesie označenie Baas (Backend-as-a-Service). Je vyvinutý spoločnosťou Google a začínal ako startup, ktorý vyrástol do vývojovej platformy pre budúcu generáciu Google Cloud Platform.

Firestore využívam hlavne ako realtime databázu, ale ponúka oveľa viac. Autentizácia, online úložisko, crash reporting. Všetky spomínané funkcie a oveľa viac ponúka Firestore, čím vývojárom webových a mobilných aplikácií uľahčuje prácu so serverom. Viac informácií o použití služby Firestore je popísaných v nasledujúcej kapitole.

⁴Firestore – <https://firebase.google.com>

Kapitola 5

Implementácia

Nasledujúca kapitola popisuje implementáciu mobilnej aplikácie Helpmate. Aplikácia prešla niekoľkými návrhmi užívateľského rozhrania. Výsledný návrh bol implementovaný a zhrnutie implementácie sa nachádza v nasledujúcich riadkoch.

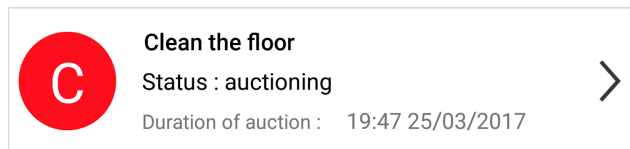
5.1 Užívateľské rozhranie

Rozloženie jednotlivých obrazoviek je uložené v samostatných súboroch deklarovaných v jazyku XML. Ako už bolo spomenuté v kapitole 2, Android týmto spôsobom oddeľuje funkčnú logiku od vzhľadu. V tejto bakalárskej práci je celkovo 81 súborov v jazyku XML, ktoré popisujú rozloženie komponentov na obrazovke. Na ich implementáciu sa využíva predovšetkým *LinearLayout*, ktorý umožňuje rozloženie prvkov lineárne a je potomkom triedy *ViewGroup*. Okrem *LinearLayout* som často využíval aj *RelativeLayout*. Definícia oboch layoutov sa nachádza na 2.4.

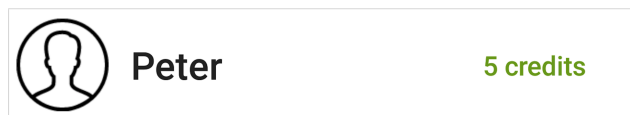
Vlastný ListView

ListView predstavuje zoznam položiek pod sebou v predom definovanom tvare. V predchádzajúcich kapitolách som viac-krát spomenul list a zoznam prác, skupín či členov. Pre všetky tieto zoznamy som implementoval vlastný vzor položky zoznamu, ktorý som následne zobrazoval prostredníctvom triedy *CustomAdapter*.

Jedna položka bola definovaná pomocou horizontálneho lineárneho layoutu a skladá sa celkom z troch častí. Prvou je priestor určený pre obrázok avatara užívateľa obr. 5.2 alebo pre farebný kruh obsahujúci vo svojom strede jedno písmeno obr. 5.1. Pre vykreslenie farebného kruhu som použil *TextView*. Jeho pozadie som upravil a vložil doň XML súbor definujúci tvar kruhu. Druhou časťou je vertikálny lineárny layout obsahujúci jeden alebo tri riadky s komponentou *TextView*. Počet sa mení v závislosti na situácii. Poslednou časťou vlastnej položky listu je priestor určený pre 2 vzory. Prvým je zobrazenie ponuky v aukcii. V tomto prípade sa zobrazí počet ponúknutých kreditov. Druhým vzorom je obrázok šípky smerujúcej doprava, ktorá indikuje možnosť rozkliknutia položky.



Obr. 5.1: Položka listu pre zobrazenie práce. Písmeno vo farebnom kruhu reprezentuje prvé písmeno názvu práce. Status sa mení v závislosti na situácii. Šípka smerujúca do pravej strany indikuje možnosť stlačenia a zobrazenia podrobnejšieho obsahu.



Obr. 5.2: Položka listu pre zobrazenie ponuky v aukcii. Obsahuje zľava obrázok avatara, meno užívateľa a jeho pridanú kreditovú ponuku.

Implementácia triedy `CustomAdapter` sa nachádza o pár riadkov nižšie v sekcii 5.3.

Menu aplikácie

`NavigationView` je typická komponenta, ktorá sa využíva pre implementáciu menu. Pôvodné menu obsahovalo šesť položiek, pričom každá z nich spúšťala inú aktivitu. Výsledná implementácia však zobrazuje položiek päť. Šiestou položkou sú takzvané *Moje Skupiny*, ktoré sa rozdeľujú na ľubovoľný počet častí, podľa počtu skupín. Android má vlastnú reprezentáciu rozhrania menu, ktorá umožňuje jednoduchým spôsobom zvoliť prvky menu aj s ich ikonami. Keďže sa však šiesta položka rozdeľuje na viaceré časti, bolo potrebné implementovať menu iným spôsobom bez využitia spomínanej reprezentácie rozhrania menu. Vytvoril som preto súbor `custom_menu` v jazyku XML, ktorý sa zaraďuje do zložky `layouts`. Vlastná implementácia menu mi umožnila využitie `ListView`. Toto rozšírenie som pridal ku komponente `NavigationView` ako môžete vidieť v ukážke zdrojového kódu:

```
<android.support.design.widget.NavigationView
    android:id="@+id/navigation_view"
    app:theme="@style/ToolBarTheme"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:background="#363636">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <include layout="@layout/nav_header_to_do"/>
        <include layout="@layout/custom_menu" />
    </LinearLayout>

</android.support.design.widget.NavigationView>
```

Ikona aplikácie

Prvá interakcia užívateľa aplikácie je práve prostredníctvom ikony. Ak by už tá pôsobila neprofesionálne, užívateľa by pravdepodobne aplikácia nezaujala. Z tohto dôvodu bolo mojim cieľom navrhnuť na pohľad atraktívnu, a pritom jednoduchú ikonu. Realizácia návrhu je sprostredkovaná vo frameworku Adobe Photoshop.

Prvým návrhom bol obrázok zobrazujúci podanie dvoch rúk. Myšlienka nebola zlá, avšak veľkosť ikony nepostačovala na to, aby užívateľ pochopil jej význam. Mnoho slávnych aplikácií má v ikone iniciálky názvu alebo jeho skratku. Povedal som si, že tento spôsob bude tou správnou cestou. V strede ikony som umiestnil 3 znaky, ktoré môžete vidieť na obrázku 5.3. Písmená **h** a **m** reprezentujú názov Helpmate, a bodka za týmito znakmi vyjadruje skrátený tvar tohto názvu.

Samozrejme, ikona je vo farbách aplikácie a to konkrétne oranžová s bielou na čiernom pozadí. Ikona aplikácie je vektorová, aby bola zabezpečená neklesajúca kvalita aj pri väčšom rozlíšení. Tvar ikony je typu **square**, ktorý je vlastne štvorec so zahnutými rohmi.



Obr. 5.3: Ikona aplikácie Helpmate

Animácie

Pod pojmom animácie sú v tomto zmysle myslené prechody medzi dvoma aktivitami. Bežným prechodom je jednoducho stmaviť jednu aktivitu a zobrazit druhú na tom istom mieste. Operačný systém Android od verzie Nougat využíva na prechod medzi aktivitami animáciu, pri ktorej sa akoby obrazovka posunie do ľavej strany a zmizne, a druhá prichádza z pravej strany a zostane zobrazená. Keďže aplikácia Helpmate je dostupná od verzie Android KitKat, rozhodol som sa tieto animácie implementovať sám, aby boli dostupné vo všetkých podporovaných verziách. Ukážka zdrojového kódu animácie *right_slide_in* sa nachádza tu:

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_decelerate_interpolator">
    <translate
        android:fromXDelta="-100%p"
        android:toXDelta="0"
        android:duration="200"
    />
</set>
```

V aplikácii sú celkovo implementované animácie pohybu:

- zhora nadol
- zdola nahor
- zľava doprava
- zprava doľava

5.2 Aktivity a ich implementácia

V kapitole 4 je spomenutý návrh najdôležitejších aktivít. V tejto sekcii sa nachádza popis problematiky spojenej s implementáciou zaujímavých aktivít v jazyku Java [13] a taktiež ich ukážky.

MainActivity

Hlavná obrazovka aplikácie sa skladá z troch hlavných widgetov: `Toolbar`¹, `TabLayout`² a `ViewPager`³.

V toolbare sa nachádza text s názvom aktuálnej skupiny, ikona reprezentujúca tlačítko menu a ikona reprezentujúca tlačítko nastavenia. Pomocou metódy `onOptionsItemSelected()` získavam informácie o stlačení tlačítka. Následne podľa unikátneho ID sa vykonáva stanovená akcia.

`TabLayout` poskytuje prácu so záložkami. Na hlavnej aktivite sa nachádzajú tri záložky. Každá z nich zobrazuje iný obsah a to pomocou `ViewPager`. Po kliknutí na niektorú záložku listener `TabLayoutu` informuje o tom, ktorá bola stlačená. Dochádza k volaniu metódy v triede `Pager`, ktorá rozhoduje o zobrazení určitého fragmentu. Existujú tri fragmenty: `Tab_ToDo`, `Tab_MyWorks` a `Tab_Completed`. Jednotlivé fragmenty reprezentujú triedy, ktoré zobrazujú predom definovaný obsah. Napríklad trieda `Tab_Completed` zobrazuje list dokončených prác, ktoré získava zo serveru. Ukážka aktivity `MainActivity` sa nachádza na obrázku 5.4.

Trieda `MainActivity` však neslúži len pre zobrazenie užívateľského rozhrania. Taktiež kontroluje a zobrazuje notifikácie, nastavuje dĺžku nešpecifikovanej aukcie, jej priebežne vyhodnocovanie a riadi ukladanie dát novej práce na server. Bližší popis týchto vlastností sa nachádza nižšie..

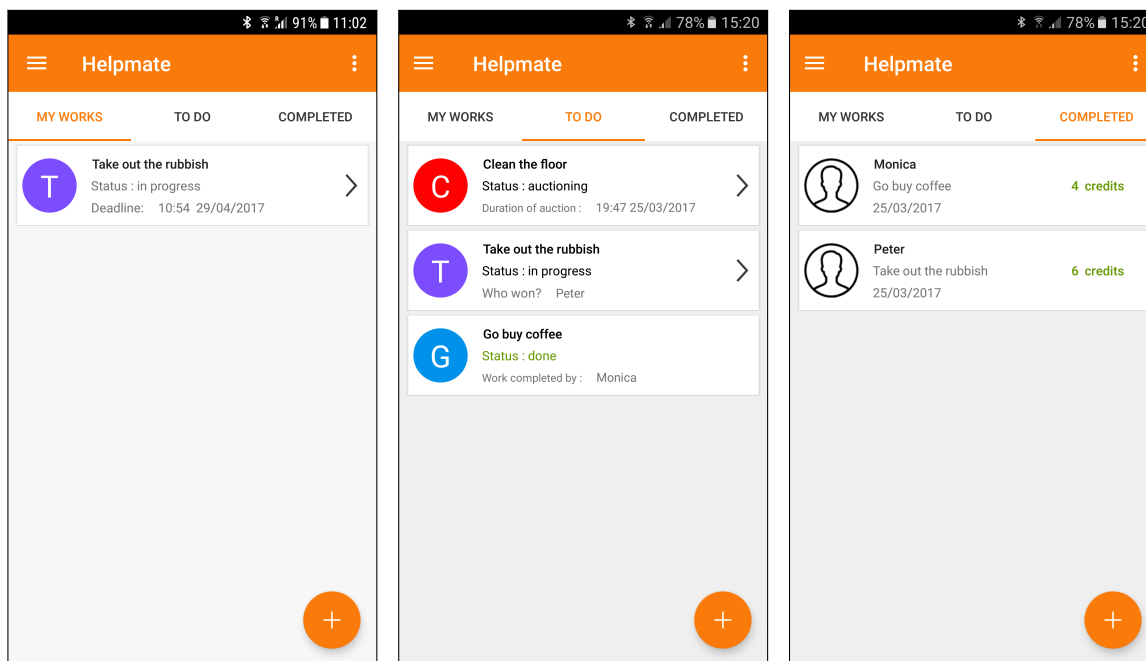
AuctionActivity

Pri otvorení jednej práce z listu na hlavnej obrazovke sa zobrazí aktivita `AuctionActivity`. Funkcia `putExtra()` zabezpečuje prenos dát objektu z jednej aktivity do druhej. Prenášajú sa tak z `MainActivity` všetky potrebné dáta na zobrazenie detailov práce v `AuctionActivity`. Pod detailami je už spomínaný list s mnou implementovanými položkami. Pri vytvorení novej práce je tento list prázdny a neskôr každá položka reprezentuje jednu ponuku v akcii. Po pridaní novej ponuky prostredníctvom floating action buttonu sa na server uloží nová

¹`Toolbar` – <https://developer.android.com/reference/android/support/v7/widget/Toolbar.html>

²`TabLayout` – <https://developer.android.com/reference/android/support/design/widget/TabLayout.html>

³`ViewPager` – <https://developer.android.com/reference/android/support/v4/view/ViewPager.html>



Obr. 5.4: Ukážka hlavnej obrazovky aplikácie Helpmate. Vľavo zobrazená záložka so zoznamom prác pridelených konkrétnemu užívateľovi, v strede záložka so zoznamom prác pre celú skupinu a vpravo záložka so zoznamom dokončených prác.

ponuka a následne sa stiahne pre zobrazenie v liste. Trieda `AuctionActivity` taktiež pri pridaní každej ponuky kontroluje, či nie je daná ponuka rozhodujúcou. V takom prípade sa použije metóda `finish()`, ktorá ukončí aktivitu `AuctionActivity`. Oznámi hlavnej aktivite aktuálne zmeny a tá sa podľa toho aktualizuje. Screenshot tejto aktivity sa nachádza na obrázku 5.7b.

Táto aktivita však nezobrazuje len aukciu. Po pridelení práce členovi skupiny sa danému užívateľovi zobrazí táto práca v záložke Moje Práce. Pri otvorení položky práce sa zobrazí mierne pozmenené užívateľské rozhranie. Detailné informácie k úlohe sa nezmenia, avšak namiesto aukcie sa tu nachádza `ProgressBar`. `ProgressBar` slúži predovšetkým ako nápoveda pre ostatných členov skupiny, aby boli informovaný o aktuálnom vykonávaní určenej činnosti. Dokončenie práce je potvrdené kliknutím na príslušné tlačítko. Dochádza k volaniu metódy `addToGraph()`, ktorá ukladá do databázy informácie o dokončenej práci pre zobrazenie v grafe. Popis implementácie vykresľovania grafu sa nachádza o pár riadkov nižšie.

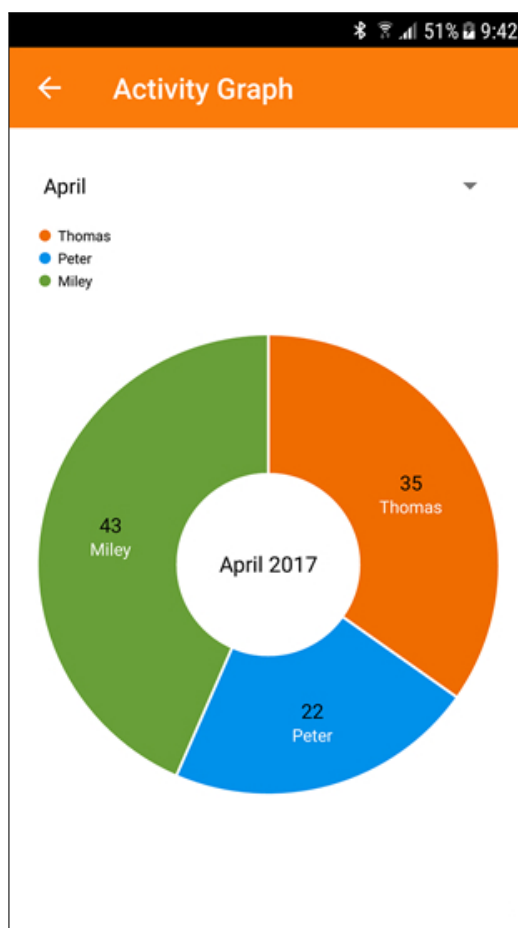
AddNewWorkActivity

Aby bolo pridávanie novej práce čo najrýchlejšie a najjednoduchšie, bolo potrebné jej UI patrične premyslieť. Pre pridanie novej práce je potrebné špecifikovať 4 údaje: meno práce, dĺžku práce a čas s dátumom, dokedy je potrebné prácu dokončiť. Pre nastavenie času a dátumu existujú dve pomocné triedy. Jedná sa o takzvané `PopUp` aktivity, keďže sa jej okná nezobrazujú cez celú šírku a výšku obrazovky. Tieto aktivity sú podobné klasickým dialógovým oknám operačného systému Android. `PopUp` aktivity využívajú grafické widgety `TimePicker` a `DatePicker`. Po uložení dátumu a času sú z nich získavané údaje, ktoré sú

následne parsované do potrebného tvaru. Nešpecifikovaná dĺžka aukcie sa vypočítava metódou `setUnspecifiedDeadline()`, ktorá prideli primeranú dĺžku aukcie na základe času a dátumu dĺžky práce. Ukážka tejto aktivity sa nachádza na obrázku 5.7a.

GraphActivity

`GraphActivity` využíva pre zobrazenie grafu knižnicu `PieChart`⁴ od Mitch Tabian. Táto knižnica vykresľuje v aplikácii `Helpmate` koláčový graf obr. 5.5 podľa dát získaných z databáze `Firestore`. Podiel vykonanej práce jednotlivých členov skupiny sa zobrazuje v percentách pre jednoduchšiu orientáciu v grafe. Na tejto aktivite sa nachádza aj `Spinner` zobrazujúci zoznam všetkých mesiacov roku. Po kliknutí na daný mesiac sa vyfiltrujú dáta zo serveru a aktualizuje sa graf vďaka metóde `changeMonthToView()`. Dáta sú v databáze uchovávané jeden rok a ich vymazanie prebieha prostredníctvom aktualizovania novými dátami. Napríklad dáta z Apríla 2017 budú nahradené novými dátami až v Apríli 2018. Týmto spôsobom je umožnený dostatočne dlhý čas na prehľad pre užívateľov.



Obr. 5.5: Screenshot aktivity `GraphActivity`. V hornej časti sa nachádza `Spinner` pre zmenu aktuálne zobrazeného mesiaca. Pod ním sa nachádza legenda ku grafu. Samotný graf zobrazuje v percentách aktívnosť jednotlivých členov.

⁴`PieChart` – <https://github.com/mitchtabian/Pie-Chart-Tutorial/tree/master/PieChartTutorial>

Pri odobraní užívateľa zo skupiny sa spolu s ním nevymažú jeho kredity. Tie sú ponechané v databáze. S novým mesiacom sa v databáze vytvoria položky členov skupiny, v ktorých už odstránený užívateľ nie je, ale jeho aktivita v predchádzajúcich mesiacoch je stále dostupná.

Počas načítania grafu je pre každú skupinu určená konkrétna vetva, z ktorej je načítanie potrebných informácií jednoduché. Ukážka zdrojového kódu použitia knižnice PieChart sa nachádza o pár riadkov nižšie.

```
pieChart = (PieChart) findViewById(R.id.pieChart);
pieChart.setRotationEnabled(true);
pieChart.setUsePercentValues(true);
pieChart.setNoDataTextColor(Color.parseColor("#EF6C00"));
pieChart.setCenterTextColor(Color.BLACK);
pieChart.setHoleRadius(40f);
pieChart.setTransparentCircleAlpha(0);
pieChart.setCenterText(monthToShow(monthString) + " " + monthYear);
pieChart.setCenterTextSize(15);
pieChart.setEntryLabelTextSize(12);
```

Ukážka zdrojového kódu sa tu nachádza z dôvodu nedostatočnej dokumentácie k tejto knižnici. Zdrojový kód presne odpovedá zobrazeniu grafu v aplikácii Helpmate, kde je predom definovaný rádius, text v strede koláčového grafu, či povolenie zobrazenia hodnôt v percentách.

IntroActivity

Trieda pre zobrazenie intra aplikácie neobsahuje viac ako 50 riadkov zdrojového kódu. Využitá je knižnica AppIntro⁵, ktorá uľahčuje prácu a navyše je výsledný vzhľad malebný.

Intro sa delí na štyri takzvané *slidy*, kde každý jeden zobrazuje iný súbor XML. Intro aplikácie by malo poskytovať jednoduchý a rýchly prehľad toho, čo aplikácia vykonáva. V strede každého slidu sa nachádza obrázok vyjadrujúci danú akciu aplikácie a pod ním je stručný popis.

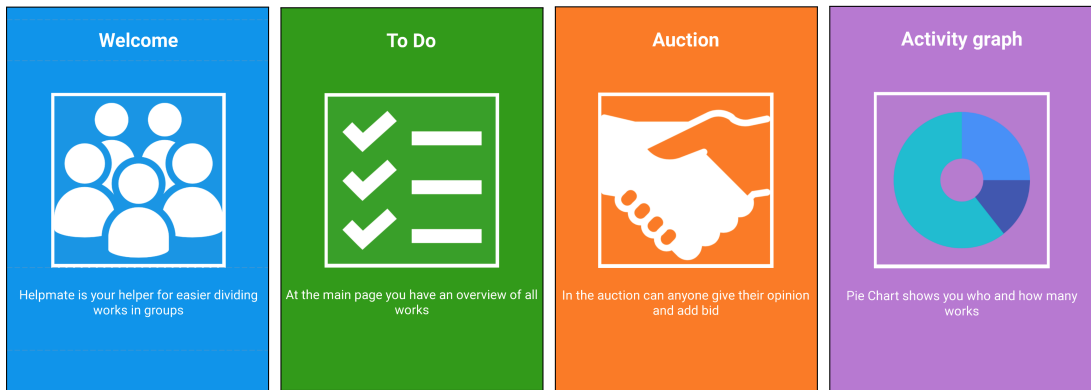
Intro aplikácie sa spustí len jedenkrát a to pri prvom spustení aplikácie. Spodná lišta zobrazuje animačným spôsobom počet slidov a tiež aktuálne zobrazený slide. V pravom dolnom rohu je zobrazené tlačítko s ikonou šípky indikujúce dokončenie tutoriálu. Screenshot sa nachádza na obrázku 5.6.

SignInActivity a LoginActivity

Poslednými aktivitami sú aktivity pre registráciu a prihlásenie užívateľa. Nový užívateľ má na výber z troch metód registrácie:

- Registrácia na základe emailu, mena a hesla
- Registrácia prostredníctvom účtu Facebook
- Registrácia prostredníctvom účtu Google

⁵AppIntro – <https://github.com/apl-devs/AppIntro>



Obr. 5.6: Screenshoty intra aplikácie Helpmate

Prvú možnosť zabezpečuje služba Firebase. Ďalšie dve možnosti boli na implementáciu o niečo náročnejšie. Facebook aj Google poskytujú však vo svojich dokumentáciách dostatočné informácie. Po kliknutí na ikonu Google, ktorá je vidieť na obrázku 5.7c, sa zobrazí dialógové okno účtami. Vďaka registrácii na stránke Google (určenej pre developerov) som získal prístupový token (kód). Ukážka zdrojového kódu zobrazuje postup umožňujúci toto prihlásenie. Následne, po vybratí účtu získavam vyžiadané informácie potrebné pre registráciu nového užívateľa.

```

GoogleSignInOptions gso = new GoogleSignInOptions.Builder(
    GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(
        R.string.google_access_token))
    .requestEmail()
    .build();

mGoogleApiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this, this)
    .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
    .build();

```

5.3 Synchronizácia a načítanie dát

Kľúčovou vlastnosťou aplikácie Helpmate je možnosť ukladania a získavania dát online. Táto sekcia popisuje akým spôsobom pristupujem k dátam, ako ich ukladám, spracujem či zobrazujem.

FirestoreHelper

Aby bolo možné dáta získavať, je potrebné ich v prvom rade uložiť na server. Za týmto účelom je implementovaná metóda `save()` v triede `FirestoreHelper`. Táto trieda však dáta len neukladá, ale pomocou metódy `retrieve()` sú implementované listenery, ktoré detekujú zmeny na serveri. Výsledná databáza poskytuje veľký počet informácií potrebných pre správny beh aplikácie, a z toho dôvodu jedna trieda `FirestoreHelper` nevystačovala. Konečný počet

takýchto tried je deväť. Každá jedna trieda spracováva dáta mierne iným spôsobom, aby bola výsledná aplikácia čo najrýchlejšia.

V základnom tvare je metóda `save` jednoduchá. Prijíma objekt triedy (napríklad `NewWork`) reprezentujúci novú prácu a následne ho posiela na sever. Nasledujúci riadok zobrazuje ukážku zo zdrojového kódu:

```
db.child("groups").child(groupID).child("works").child("todo").push()  
.setValue(newWork);
```

V ukážke zdrojového kódu metóda `child()` reprezentuje názov vetvy v databáze, `push()` reprezentuje odoslanie novej zmeny do databázy, pričom sa využíva `setValue()` pre nastavenie hodnôt k odoslaniu.

Metóda `retrieve()` je už o niečo komplikovanejšia. `AddChildEventListener()` zabezpečuje detekovanie zmien v databáze na serveri. Pomocou `override` metód sleduje v určenej kategórii pridanie do databázy, zmenu, odstránenie, premiestnenie a prípadnú chybu spojenia. Detekovaná zmena sa predáva metóde `fetchData()`, ktorá zapíše dáta do objektu triedy a uloží do poľa `ArrayList`.

CustomAdapter

Trieda `CustomAdapter` umožňuje zobrazenie vlastných položiek zoznamu `ListView`. Je to takzvaný `BaseAdapter`⁶, ktorý na základe objektu triedy reprezentujúcej dané dáta zobrazuje jednotlivé položky zoznamu. Okrem iného, táto trieda umožňuje po kliknutí na danú položku získať z nej potrebné informácie, ktoré sa cez `putExtra()` predávajú ďalej.

Pri spustení aplikácie sa pomocou triedy `FirebaseHelper` získajú potrebné dáta zo servera a výsledok získania dát sa predáva do metódy `CustomAdapter`. Ten zobrazí prijaté informácie spôsobom popísaným vyššie.

5.4 Jazyková podpora

Implementácia podpory iných jazykov je nutnosťou pri zverejnení aplikácie na Google Play. Aplikácia `Helpmate` poskytuje slovenský, český a anglický jazyk. Dobudúcna by som však chcel jazykové balíčky rozšíriť aj o ďalšie.

Pri prvom spustení aplikácie mi knižnica `Locale` umožňuje získať predvolený jazyk z mobilného zariadenia. Pokiaľ to nie je ani slovenský ani český jazyk, implicitne je nastavený jazyk anglický. Táto voľba je uložená prostredníctvom `SharedPreferences`⁷, ktoré ukladajú vybrané údaje lokálne v danom zariadení. Pri otvorení nastavení aplikácie dochádza k načítaniu tohto údaje zo `SharedPreferences` pre informatívny prehľad určený užívateľovi. Samozrejme, každý má možnosť si jazykový balíček zmeniť v nastaveniach.

Aby bolo možné jazykové balíčky jednoducho meniť, bolo potrebné všetky textové reťazce v aplikácii ukladať do súborov `strings` vo formáte XML. Prístup k reťazcom zabezpečuje funkcia `getString()` prostredníctvom identifikačného čísla.

⁶`BaseAdapter` – <https://developer.android.com/reference/android/widget/BaseAdapter.html>

⁷`SharedPreferences` – <https://developer.android.com/reference/android/content/SharedPreferences.html>

5.5 Alarm a notifikácie

V predchádzajúcich riadkoch je spomenutá veľká funkcionálnosť triedy `MainActivity`. Táto sekcia podrobnejšie popisuje funkcionálnosť alarmu a notifikácií.

Alarm

Pod pojmom alarm nie je samozrejme myslený klasický budík. Využívam však rozhranie `WakefulBroadcastReceiver`⁸, ktoré umožňuje prebudenie mobilného zariadenia v stanovenom čase. Prvý alarm sa vždy nastaví podľa dĺžky aukcie, aby bolo možné aukciu v čas vyhodnotiť. Pri reštartovaní mobilného zariadenia by však bol alarm ukončený. Z toho dôvodu je v triede `MainActivity` implementovaný listener, ktorý kontroluje nastavený alarm.

K ďalšiemu alarmu dochádza po vyhodnotení aukcie. Ten slúži pre stanovenie konca práce. Ak užívateľ neukončí prácu do stanoveného termínu, práca mu je odobraná a nezíska za ňu žiadne ohodnotenie. Za týmto rozhodnutím je trieda `AlarmProgressReceiver`.

Posledným alarmom implementovaným v aplikácii je `WorkDoneReceiver`. Dĺžka tohto alarmu je implicitne 6 hodín a slúži na informovanie ostatných členov skupiny. Tí vidia ešte 6 hodín na hlavnej obrazovke aplikácie prácu so statusom dokončenia a po uplynutí tejto doby sa práca sama odstráni. Pre odstránenie je potrebné poznať konkrétne umiestnenie v databáze, konkrétnu vetvu nastaviť na `null` a následne, prostredníctvom Firebase funkcie `removeValue()`, sa vetva odstráni.

K spusteniu alarmu však nemôže prísť bez internetového pripojenia. Najnovšia aktualizácia by nebola poslaná na server a neinformovalo by sa tak ostatných o zmene stavu. Metóda `isOnline()` prostredníctvom `ConnectivityManager`⁹ zisťuje internetové pripojenie. Iba v prípade, že má mobilné zariadenie prístup na internet, dochádza k vyhodnoteniu. V opačnom prípade sa stanovená akcia nevykoná, až pokiaľ toto pripojenie nie je obnovené.

Notifikácie

Samotné vytvorenie notifikácii nie je komplikovaný proces. Je potrebné vytvoriť objekt triedy `PendingIntent`¹⁰, `Notification`¹¹, `NotificationManager`¹² a priradiť im správne hodnoty s notifikujúcim textom. Takáto implementácia nezaberie viac ako 10 riadkov. Lenže, z dôvodu intuitívnejšej implementácie notifikácii bola moja skúsenosť s touto problematikou relatívne zaujímavá.

Prvý je vytváraný objekt triedy `SetNotification`. Trieda `SetNotification` pripravuje osem typov správ pre notifikáciu a taktiež vytvára identifikačné číslo notifikácii prostredníctvom metódy `generateRandomNumber()`. Trieda `SetNotification` ale samotné notifikácie nevytvára. Pripravenú správu s vyžadovanou špecifikáciou ukladá do databázy. Každý užívateľ má vlastnú vetvu v databáze s názvom `notification`. Trieda `MainActivity` obsahuje listener priamo na tieto vetvy v databáze. Pokiaľ dôjde k zisteniu pridania novej notifikácie, tento listener to okamžite zachytí.

⁸`WakefulBroadcastReceiver` – <https://developer.android.com/reference/android/support/v4/content/WakefulBroadcastReceiver.html>

⁹`ConnectivityManager` – <https://developer.android.com/reference/android/net/ConnectivityManager.html>

¹⁰`PendingIntent` – <https://developer.android.com/reference/android/app/PendingIntent.html>

¹¹`Notification` – <https://developer.android.com/reference/android/app/Notification.html>

¹²`NotificationManager` – <https://developer.android.com/reference/android/app/NotificationManager.html>

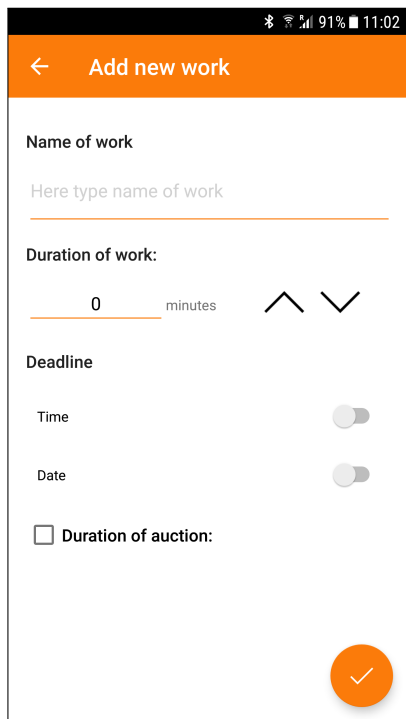
Identifikačné číslo notifikácie má dôležitú úlohu. Bolo by neefektívne, aby sa napríklad k dokončenej práci zobrazili po neskoršom prihlásení aj notifikácie z pridania danej práce. Za týmto účelom slúži práve vyššie spomenutý generátor náhodných čísiel, ktorý vytvára identifikačné číslo notifikácie. Pri ukladaní novej predpripravenej notifikácie na server sa v prvom rade zisťuje, či už notifikácia s rovnakým ID v databáze nie je. Ak už existuje, správa sa aktualizuje a užívateľovi sa zobrazí len najaktuálnejšia notifikácia.

Poslednou dôležitou časťou pri vytváraní notifikácii je trieda `StringForNotifications`. Predpripravená správa notifikácie v databáze neobsahuje konkrétny reťazec, ale iba číslo reprezentujúce jej správu. V momente, keď metóda `control_notification()` v triede `MainActivity` detekuje novú notifikáciu, vytvára si inštanciu na túto triedu. Ďalej sa posielajú získané dáta z `dataSnapshot` do metódy `setNotificationString()`. Krátka ukážka metódy zobrazuje jej funkčnosť:

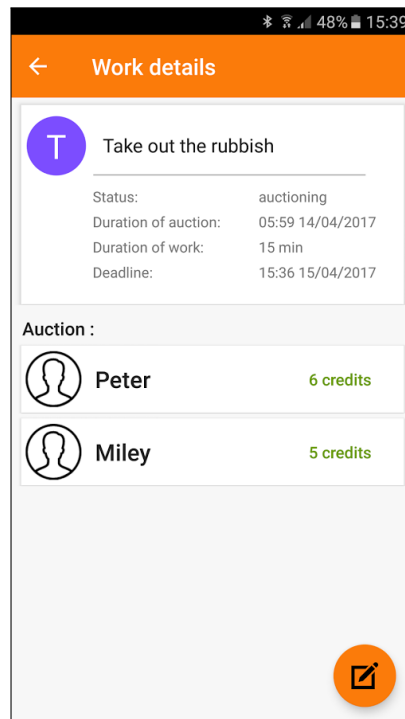
```
public String setNotificationString(String notifNumber, String addToNotif,
Context a)
{
    String returnNotif;

    if(notifNumber.equals("1"))
    {
        returnNotif = a.getString(R.string.new_work_added) + addToNotif;
    }
    else if(notifNumber.equals("2"))
    {
        returnNotif = a.getString(R.string.new_user_added) + addToNotif;
    }
    else if(notifNumber.equals("3"))
    {
        returnNotif = a.getString(R.string.user) + addToNotif +
a.getString(R.string.removed_from_group);
    }
    ...
}
```

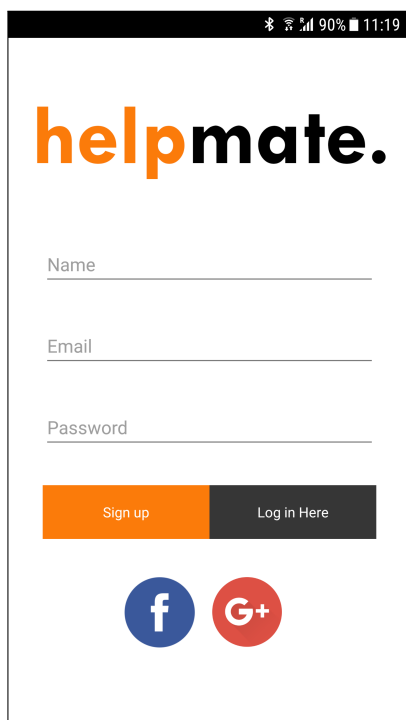
Na ukážke je vidieť dôležitosť jazykovej podpory. V prípade, že jeden člen skupiny by bol Angličan a druhý Slovak, notifikácia by sa jednému z nich zobrazila v nesprávnom jazyku.



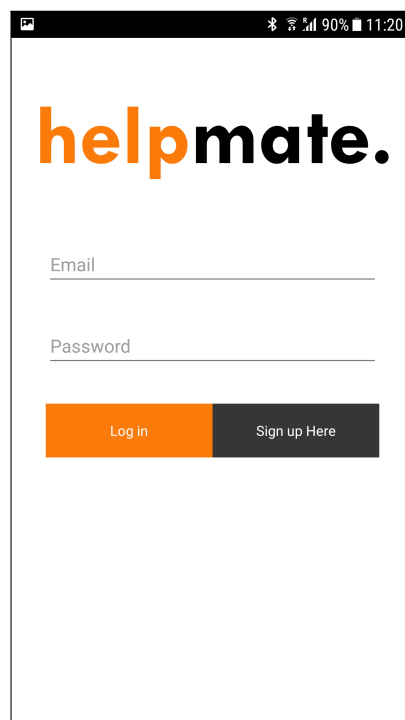
(a) Pridanie novej práce



(b) Aukcia danej práce



(c) Registrácia užívateľa



(d) Prihlásenie užívateľa

Obr. 5.7: Ukážky aktivít aplikácie s popisom ich reprezentácie

Kapitola 6

Testovanie aplikácie

Testovanie [2] je neoddeliteľnou súčasťou vývoja softwaru. Pri vývoji aplikácie Helpmate boli za účelom testovania použité reálne mobilné zariadenia a emulátory mobilných zariadení. Testovanie slúži pre overenie kvality softwaru. Zisťuje sa, či software spĺňa potrebné parametre funkčnosti, spoľahlivosti, či použiteľnosti.

Predposledná kapitola s názvom testovanie aplikácie popisuje priebeh testovania na reálnych zariadeniach, aj emulátoroch zariadení. Zverejnením aplikácie na Google Play sa zaoberám na záver tejto kapitoly.

6.1 Emulátor v Android Studio

Pri stiahnutí frameworku Android Studio¹ sa nachádza v balíčku SDK² okrem iného aj emulátor operačného systému Android. Predchádzajúce verzie spomínaného frameworku neposkytovali dostatočnú rýchlosť pre testovanie aplikácie prostredníctvom emulátoru. Všetko sa však zmenilo s príchodom verzie 2.0, kde sa spoločnosť Google pýši až 50-krát vyššou rýchlosťou emulátoru.

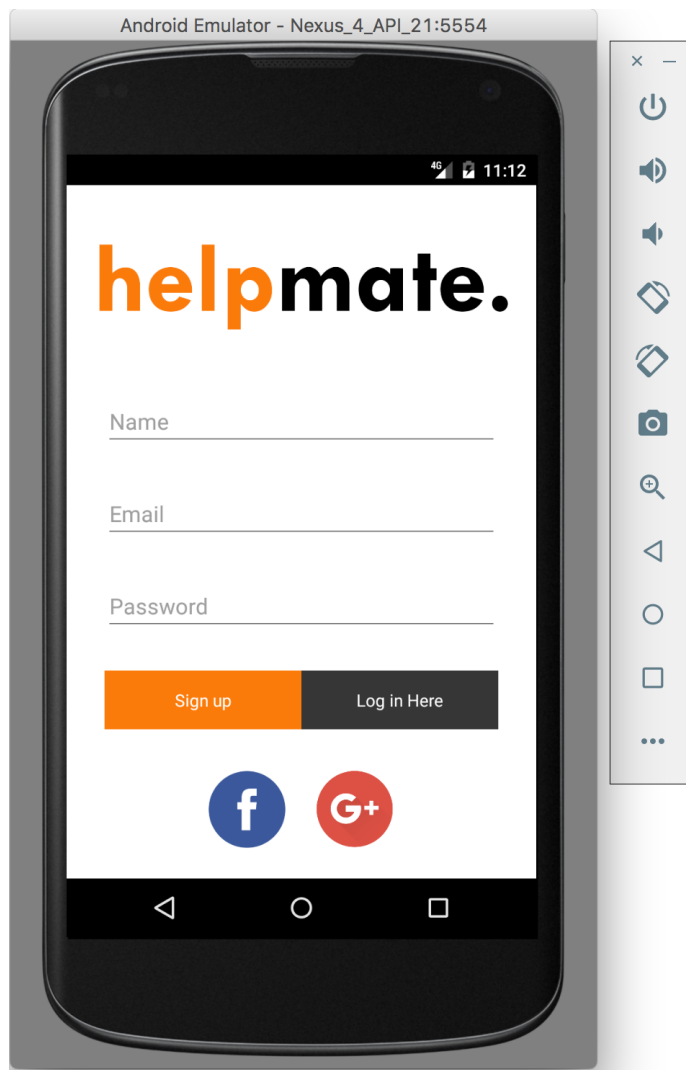
V priebehu vývoja aplikácie som pre testovanie aplikácie Helpmate využíval na začiatku emulátor mobilného telefónu Nexus 4. Jeho ukážka sa nachádza na obrázku 6.1. Podstatnou výhodou emulátorov je možnosť prispôbiť si ich podľa seba. V ponuke je pripravených viacero zariadení, ktoré môžeme jednoducho využiť. V nastaveniach však nachádzame aj možnosť vytvorenia si vlastného emulátoru podľa nami zvolených parametrov. Nastavenie veľkosti displeja, určenie verzie Androidu, použitá časť procesoru a ďalšie.

Keďže je aplikácia určená pre skupiny užívateľov, mohol som ľubovoľne veľké skupiny simulovať pomocou vytvorenia viac ako jedného emulátoru. Emulátor má prístup k internetovému pripojeniu, takže aj počas testovania aplikácie na reálnom zariadení som ho mohol využívať pre rozšírenie.

Stretol som sa aj s niekoľkými nevýhodami. Za spomenutie by stála práve rýchlosť. Pokiaľ počítač nie je nadpriemer výkonný, tak aj napriek tvrdeniu Googlu je stále použitie reálneho zariadenia výrazne lepšou voľbou.

¹Android Studio – <https://developer.android.com/studio/index.html>

²SDK – Software Development Kit



Obr. 6.1: Emulátor mobilného zariadenia Nexus 4

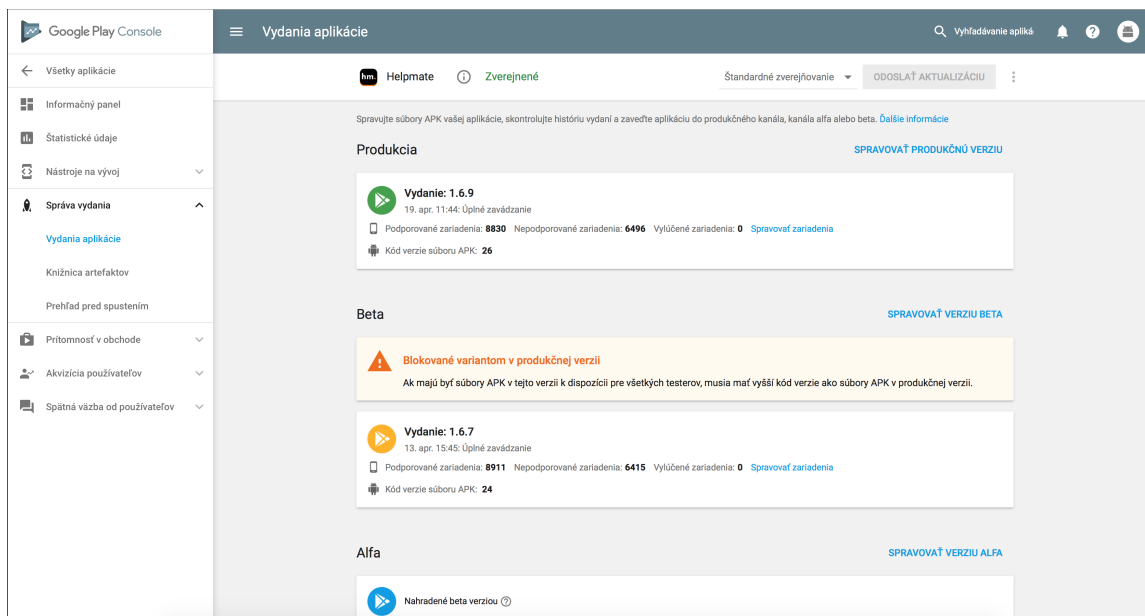
6.2 Reálne zariadenia

Ako som už spomenul v sekcii 6.1, v priebehu testovania som preferoval reálne mobilné zariadenie. Využíval som mobilné telefóny Samsung Galaxy S6, Samsung Galaxy S5 mini a tablet Samsung Galaxy Tab 2. Tablet poslúžil na testovanie užívateľského rozhrania na väčších zariadeniach. Vďaka nemu som mohol upraviť UI do finálnej podoby, aby aplikácia vyzerala rovnako a nie menej atraktívne aj na zariadeniach s nemalou uhlopriečkou.

6.3 Zverejnenie aplikácie na Google Play

V dnešnej dobe zaberie založenie vývojárskeho účtu na Google Play³ necelú hodinu. Za menší poplatok takýto účet poskytuje funkcie, ktoré určite ocení každý Android Developer. Google Play umožňuje sledovať štatistické údaje k aplikáciám, hodnotenie a recenzie, zlyhanie a chyby, typy na optimalizáciu, či propagáciu. S hodnotením obsahu sa stretne pri

³Google Play – <https://play.google.com/>



Obr. 6.2: Screenshot obrazovky z informačného systému Google Play Developer Console. Na obrázku je vidieť, že sa aplikácia nachádza v produkcii. Pod ňou sa nachádzajú kategórie Alfa a Beta, ktoré boli využité počas testovania aplikácie. Nahranie aktuálnej verzie aplikácie je možné prostredníctvom tlačítka reprezentovaného textom „Spravovať produkčnú verziu“

pridávaní novej aplikácie. Google sa týmto snaží filtrovať aplikácie, ktoré by mohli zahŕňať nepriaznivý obsah.

Časť *App releases* spravuje súbory formátu APK, zobrazuje históriu vydaní a umožňuje zavádzať aplikáciu do produkčného, alfa alebo beta kanálu. Ukážka sa nachádza na obrázku 6.2.

- Alfa a Beta kanál sa používajú pri testovaní aplikácie. Alfa testovanie je zavedené v prvej fázy, a kanál Beta sa aplikuje až po úspešnom dokončení predchádzajúcej fázy. Umožňujú otvorené a uzavreté testovanie. Pod pojmom uzavreté je myslený kolektív testerov, ktorý si môžu aplikáciu stiahnuť len prostredníctvom hypertextového odkazu. Developer týmto spôsobom zamedzuje prístup k jeho aplikácii osobám, ktoré nie sú súčasťou testovacieho tímu.
- K produkčnému kanálu sa dostávame po úspešnom dokončení testovania, kedy je už aplikácia pripravená na zavedenie do reálnej prevádzky.

6.4 Stabilita a funkčnosť

V počiatočných fázach som aplikáciu testoval sám. Samotný vývojár je zlou voľbou pre testovanie vlastnej aplikácie, pretože pozná celú implementáciu a vie, v akých situáciách sa ako aplikácia chová.

Alfa testovanie prostredníctvom Google Play je ideálnym riešením. Vytvoril som si testovaciu skupinu, ktorá sa skladala z väčšej časti z mojich spolužiakov, ale tiež z priateľov

a rodinných príslušníkov. Google Play v spojení s Firebase mi poskytli dostatočný prehľad nad týmto testovacím tímom a prostredníctvom internetu som mohol sledovať prípadné chyby a zlyhania systému.

Chyby v aplikácii

Zásluhou testovacej skupiny som narazil na niekoľko problémov. Prvou bola nekompatibilitnosť so staršími verziami Androidu. V kapitole 2 je spomenutá prvá podporovaná verzia tohto operačného systému, a to Android 4.4 KitKat. Malou nepozornosťou sa mi podarilo použiť novšiu knižnicu, ktorá nie je dostupná v spomínanej verzii. Testerom sa z toho dôvodu PopUp okná neroztiahli na stanovenú všeobecnú šírku obrazovky a nebolo preto možné v aukcii vyjadriť ich záujem k práci. Dodatočným vyhľadáním kompatibilnej verzie knižnice aj so staršou verzou Androidu bol problém vyriešený. Cez Google Play Developer konzolu trvalo updatovanie na opravenú verziu len niekoľko minút.

Na druhý a zložitejší nedostatok som narazil osobne. Pri dlhšom a nepretržitom testovaní bez ukončenia aplikácia padala. Pod pádom je myslené zlyhanie systému s vynúteným ukončením. Najšť chybu bolo zložitejšie ako jej opravenie. Ukázalo sa, že pri prechodoch z aktívnej aktivity do neaktívnej bola tá aktívna nesprávne ukončená. Nastala teda situácia, pri ktorej sa systém snažil zobrazit načítané zmeny, avšak daná aktivita už bola nahradená novou. V tom momente nebolo možné vykonať isté časti zdrojového kódu a aplikácia sa ukončila. Túto problematiku som si následne dôkladnejšie naštudoval, aby sa už podobné situácie neopakovali. Po aplikovaní nových poznatkov v aplikácii bol systém dôkladne ošetrený a testovaný práve na spomínanú chybu.

Ako hovorí jedno staré príslovie, učíme sa na vlastných chybách, a ja som si z tohto do budúcnosti odniesol užitočné vedomosti.

6.5 Testovanie užívateľského rozhrania

V priebehu testovania prešla aplikácia viacerými zmenami v UI. Dôvodom boli dotazy testerov. Aby ľudia využívali aplikáciu Helpmate radi, musí im okrem funkcionality poskytovať aj premyslené prostredie. Také však nebolo od začiatku. Aplikácia prešla v priebehu fázy Alfa niekoľkými premenami. Prvou zásadnou zmenou bol dizajn úvodnej a zároveň hlavnej aktivity. Okrem využitia inej farby bolo odstránené aj tlačítko, ktoré je zobrazené na obrázku 6.3a. Tlačítko reprezentovalo zmenu nastavenia notifikácii. Nastavenie sa ukázalo byť zbytočným, pretože editácia zvuku je vstavanou funkciou operačného systému Android a užívatelia toto tlačítko absolútne nevyužívali.

Vzhľad jednej položky zoznamu bol upravený s cieľom sprehľadniť list úloh. Každá položka má v ľavej časti okrúhlu ikonu, ktorá je vyplnená farbou a v jej strede sa nachádza písmeno. Písmená majú pridelenú farbu na pevno, pričom niektoré písmená zdieľajú rovnakú farbu. Písmeno v strede kruhu reprezentuje prvé písmeno názvu práce. Pri dlhšom používaní aplikácie Helpmate si užívatelia zvyknú na pridelené farby k určitým písmenám. To pomôže pri orientácii na hlavnej stránke a užívatelia budú ihneď vedieť, o akú činnosť sa jedná.

Vrchná lišta, tiež známa ako *Toolbar*, reprezentovala názov aplikácie. Ak chceli členovia daných skupín zistiť, v akej skupine sa práve nachádzajú, bolo potrebné vysunúť navigačné menu, alebo otvoriť aktivitu so zoznamom skupín. V tejto aktivite sa nachádzal farebná ikona pri každej položke zoznamu skupín, ktorá zelenou farbou označovala aktivovanú skupinu. Vo finálnej verzii Helpmatu je názov aplikácie v toolbare pozmenený na názov aktuálne

aktivovanej skupiny. Aktivita so zoznamom skupín bola odstránená a nahradil ju prehľadnejší výpis skupín, ktorý sa nachádza v menu. Tieto spomínané zmeny sú zobrazené na obrázku 6.3.

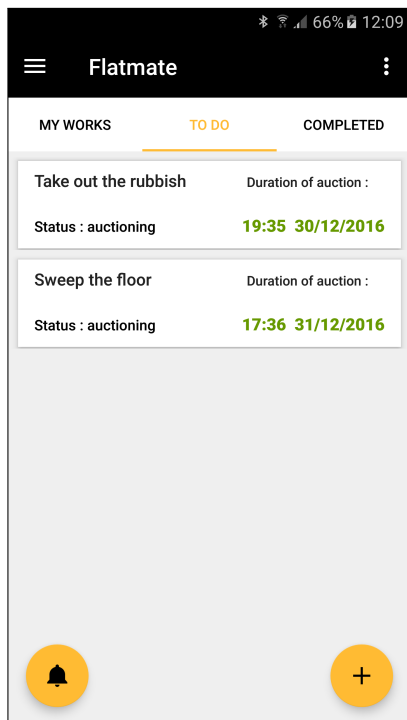
6.6 Výsledok testovania

Na konci testovania som získal reálne výsledky vďaka dvom testovacím skupinám. Obe skupiny boli prevažne študenti bývajúci v bytoch ako spolubývatelia. V jednej z týchto skupín som bol aj ja sám členom. Úloha bola jednoznačná. Je potrebné testovať aplikáciu v reálnej prevádzke. To neznamenal len prechádzať možnosti, ktoré aplikácia ponúkala a overovať funkcionality. Zámerom bolo zistiť nedostatky aplikácie, či užívateľom nejaká funkcia nechýba a či využívanie aplikácie prináša predom stanovené výsledky.

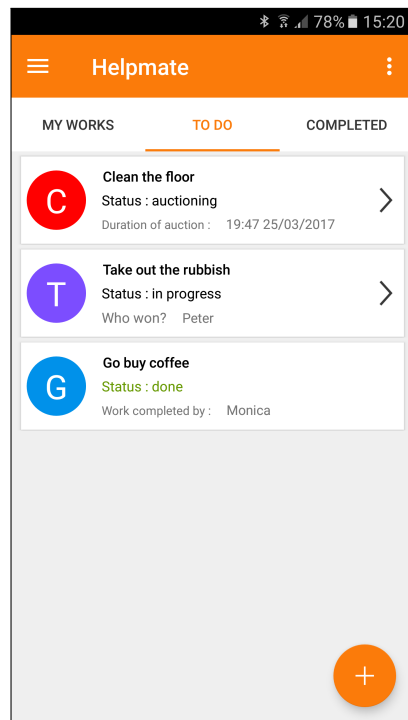
Prvá fáza trvala 7 dní. Po jej uplynutí som nadviazal kontakt s testovacími skupinami a spísal ich dotazy. Niektoré z chýb, na ktoré sa narazilo, boli popísané v predošlej sekcii. Aplikáciu Helpmate skupiny využívali každý deň a už po uplynutí vyššie stanovenej doby sa preukázali prvé pozitívne výsledky. Práca na byte boli prehľadne organizované, jednoduchým spôsobom si spolubývatelia rozdeľovali a prideliť úlohy, a už po týždni graf zobrazoval, kto zo spolubývajúcich vyčnieva. Pod vyčnievaním sú myslené dva prípady. Buď člen skupiny robil priveľa rozdiel od ostatných, alebo naopak primálo. Aplikácia teda v tomto smere splnila svoju úlohu a spolubývatelia sa v ďalšej fáze testovania snažili o rovnomernejšie a spravodlivejšie rozdeľovanie prác.

Pre druhú fázu testovania bol stanovený dvojnásobný časový úsek, čiže 14 dní. Priebežný prehľad o aktivite testerov mi poskytoval Google Play, a rovnako tak aj databáza Firebase, v ktorej som mohol sledovať celkový priebeh. Na konci druhého týždňa som sa opäť spojil s testovacími skupinami, či už osobne, alebo prostredníctvom internetového hovoru. Výsledky druhej fázy mi priniesli zaujímavý poznatok. V pôvodnej myšlienke aplikácie mala jedna minúta dĺžky práce odpovedať jednému kreditu ako ohodnotenie za vykonanú prácu. Často vyskytujúcou sa úlohou bolo umývanie riadu. Dĺžka tejto práce je v priemere 5 minút, ale samozrejme závisí na množstve riadu. Pri tejto činnosti sa skutočne využívala odmena za splnenú činnosť vo výške 5 kreditov. Avšak pri časovo náročnejších úlohách, ako je napríklad umývanie podlahy či okien, bolo toto bodové ohodnotenie podstatne vyššie. Pri dĺžke práce 30 minút bola počiatočná ponuka v akcii 100 kreditov. Z toho vyplýva, že za prácu s väčšou časovou náročnosťou si užívatelia zvykli dávať väčší počet kreditov, čo mohlo mať pozitívny vplyv na motiváciu pre jej splnenie.

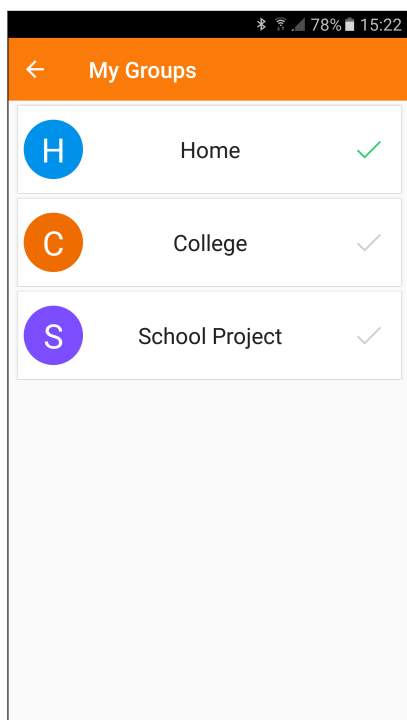
A aký je teda výsledok testovania? Dozvedel som sa pri ňom o rôznych chybách v aplikácii, o absencii funkcionalít, ktoré si jej užívatelia vyžiadali, ale predovšetkým som získal odpoveď na otázku, či aplikácia skutočne pomôže jej užívateľom. Všetky chyby a nedostatky boli opravené a výsledná aplikácia je schopná reálneho používania. Myslím si, že aplikácia Helpmate môže skutočne pomôcť rôznym sociálnym skupinám, priniesť im organizáciu a prehľad nad ich povinnosťami a taktiež motivovať členov s menším zanietením.



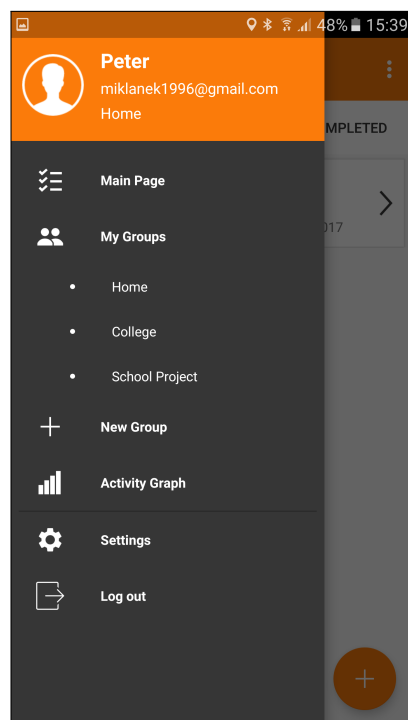
(a) Hlavná aktivita pre úpravou



(b) Hlavná aktivita po úprave



(c) Zoznam skupín pred úpravou



(d) Zoznam skupín po úprave

Obr. 6.3: Ukážky zmien v aplikácii, ktoré boli prevedené v priebehu testovania. Vrchné dva obrázky znázorňujú užívateľské rozhranie pred a po úprave, pričom hlavnými zmenami boli zmena farby a rozloženie v položke zoznamu. Spodný obrázok v ľavo zobrazuje pôvodný zoznam skupín reprezentovaný vlastnou aktivitou, pričom po úprave sa tento zoznam nachádza v menu aplikácii.

Kapitola 7

Záver

Zhodnotenie výsledku

Cieľom bakalárskej práce bolo navrhnúť a implementovať mobilnú aplikáciu pre správu práce v zdieľanom ubytovaní. Druhá kapitola obsahuje teóriu architektúry a komponentov operačného systému Android, ktoré je potrebné poznať pre správne pochopenie aplikácie Helpmate. V kapitole popisujúcej analýzu problému boli rozvedené existujúce obdobné riešenia. Tie mi spolu s analýzou poskytlí nutné informácie, z ktorých boli odvodené špecifikácie požiadaviek. Prvý návrh mobilnej aplikácie Helpmate bol uskutočnený na bloku papiera, z ktorého sa neskôr stal vzor pre implementáciu. Diagram prípadu užívania som vytvoril podľa prvého návrhu a špecifikovaných požiadaviek na funkčnosť. Diagram bol užitočný pri vývoji aplikácie a poskytol užitočné informácie využité v priebehu implementácie a testovania. V piatej kapitole som stručne rozviedol zaujímavú problematiku spojenú s vytváraním užívateľského rozhrania a implementáciou back-endu. Aplikácia bola patrične otestovaná a popisom zvolených techník som rozviedol v predposlednej časti tejto bakalárskej práce.

Počas vývoja mobilnej aplikácie som neraz narazil na rozličné zložité problémy. Niektoré z nich mi zabrali niekoľko hodín, či dní, avšak každý problém sa mi podarilo úspešne vyriešiť. Poznatky získané pri tejto bakalárskej práci sú pre mňa veľmi cenné a odnášam si tak do budúcnosti veľa skúseností. Zvažujem, že sa vývoju mobilných aplikácií budem venovať aj v budúcnosti, pretože ma platforma Android nesmierne zaujala.

Aplikácia bola spočiatku uverejnená na Google Play prostredníctvom verzie alpha. Neskôr bola posunutá do stavu beta testovania a momentálne sa aplikácia nachádza v produkcii uverejnená pre kohokoľvek.

Budúcnosť aplikácie

Ako už bolo popísané v predchádzajúcich kapitolách, záverečné riešenie bolo rozšírené pre využívanie aplikácie v ľubovoľnej sociálnej skupine. Dúfam len, že aplikáciu si oblúbia ľudia po celom svete a rád by som s rozširovaním funkcionality pokračoval. Možnými rozšíreniami by mohlo byť použitie vlastnej fotky namiesto obrázku avatara, či umožnenie rýchleho prechodu z aplikácie Helpmate do Messengeru so zámerom rýchlejšej komunikácie v danej skupine.

Literatúra

- [1] Eberts, R. E.: *User interface design*. Prentice Hall, 1994, ISBN 0-13-107053-3.
- [2] Ekambaram, V.; Sharma, V.; Rajput, N.: *Mobile Application Development, Usability, and Security*. IGI Global, 2016, ISBN 978-15-225-0946-2.
- [3] Gong, R.; Wang, Q.; Hai, Y.; aj.: Investigation on factors to influence color emotion and color preference responses. *Optik - International Journal for Light and Electron Optics*, ročník 136, č. 1.
URL <http://www.sciencedirect.com.ezproxy.lib.vutbr.cz/science/article/pii/S0030402617301699>
- [4] Google: *Activity*. [online]. [cit. 2017-05-01].
URL <https://developer.android.com/reference/android/app/Activity.html>
- [5] Google: *Add Firebase to Your Android Project*. Firebase Documentation, [online]. [cit. 2017-03-25].
URL <https://firebase.google.com/docs/android/setup>
- [6] Google: *Broadcast Receivers*. [online]. [cit. 2017-04-25].
URL <https://developer.android.com/reference/android/content/BroadcastReceiver.html>
- [7] Google: *Content Providers*. [online]. [cit. 2017-04-25].
URL <https://developer.android.com/guide/topics/providers/content-providers.html>
- [8] Google: *Intent*. [online]. [cit. 2017-04-25].
URL <https://developer.android.com/reference/android/content/Intent.html>
- [9] Google: *Material Design*. [online]. [cit. 2017-05-01].
URL <https://material.io>
- [10] Google: *Platform Architecture*. Android Developers, [online]. [cit. 2017-02-04].
URL <https://developer.android.com/guide/platform/index.html>
- [11] Google: *Services*. [online]. [cit. 2017-04-25].
URL <https://developer.android.com/guide/components/services.html>
- [12] Google: *View*. [online]. [cit. 2017-04-25].
URL <https://developer.android.com/reference/android/view/View.html>
- [13] Herout, P.: *Učebnice jazyka Java*. KOPP vydavatelstvo, 2. vydání, 2006, ISBN 978-80-723-2318-0.

- [14] Luboslav Lacko: *Vývoj aplikací pro Adroid*. Computer Press, 1. vydání, 2015, ISBN 978-80-251-4347-6.
- [15] Silberschatz, A.; Korth, H.: *Database system concepts*. McGraw Hill, 2007, ISBN 0-07-295886-3.
- [16] Ujbányai, M.: *Programujeme pro Android*. Grada Publishing, 1. vydání, 2012, ISBN 978-80-247-3995-3.
- [17] Unger, R.: *V A project guide to UX design : for user experience designers in the field or in the making*. New Riders, 2012, ISBN 978-03-218-1538-5.

Prílohy

Príloha A

Obsah DVD

Priložené DVD obsahuje nasledujúcu štruktúru súborov:

- application – zdrojové kódy aplikácie
- poster – prezentačný plagát
- video – prezentačné video
- apk – inštalačný súbor aplikácie
- latex – zdrojové kódy textovej časti práce
- pdf – elektronická verzia bakalárskej práce
- manual.pdf – návod k inštalácii aplikácie