Insert here your thesis' task.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

StudyPad - Android Client

Roman Levinzon

Department of Software Engeeniring
Supervisor: Ing. Miroslav Balík, Ph.D

March 10, 2019

Acknowledgements

THANKS (remove entirely in case you do not with to thank anyone)

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on March 10, 2019 . . . .

Czech Technical University in Prague
Faculty of Information Technology
© 2019 Roman Levinzon. All rights reserved.
This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis

Levinzon, Roman. *StudyPad - Android Client.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

Abstrakt

StudyPad je kombinace služby pro porižovaní poznámek a sociální sítě s cílem pomoci studentum zapamatovat si ruzné informace. Cílem práce je vyvinout aplikaci pro OS Android, která bude sloužit jako klient. Tento text uznává stávající řešení, obsahuje analýzu domén a požadavku, popis a výběr architektury aplikace a její implementace

Klíčová slova Android, Kotlin, MVVM

Abstract

StudyPad is a combination of note-taking service and a social network, aimed to help students to memorise different pieces of information. The goal of this thesis is to develop an application for Android OS which will serve as client. This text acknowledges existing solutions, contains domain and requirements analysis, description and choise of application's architecture and it's implementation

Keywords Android, Kotlin, MVVM

Contents

List of Figures

List of Tables

Introduction

Each year, our smartphones get smarter and mobile applications – more advanced. Arrival of smartphones and mobile applications completely changed our way of living, made it easier, and it is hard to come up with the single aspect of life, which has not been improved by one or the other application. They are everywhere: helping us to navigate in our neighbourhood, keeping us up to date with latest news, helping us to stay in touch with our loved ones, stay fit and healthy and more. And some of the most popular applications tend to teach us something new.

Educational applications are one of the most popular on both mobile platforms (iOS and Android). Many of them use flash card system in the study process – displaying small pieces of information one after another, so the user could memorise it more efficiently. However, most of these services are fairly limited in terms of what they are trying to teach and some of the greatest features are scattered across different applications and services. StudyPad wants to give its users freedom in what they could learn, and give them proper tools to share, exchange and collaborate on study materials to make the education process even more easier.

StudyPad service

StudyPad is a combination of a note taking service and a social network. It is intended for students and everyone who wishes to learn something new. StudyPad primary focus is on the content creation, its sharing and discovery.

StudyPad core concepts are *Notebooks* and *Notes*. Notebook is simply a collection of notes united by one theme. This may be a subject in school, a language that you would like to learn, or a set of questions that you could hear at a job interview. Note, in turn, is a part of the notebook and represents a single piece information that has a name and a content. It can also be interpreted as a question and answer, or a term and its definition.

Each user has his/her own space where he can create, store and edit notebooks and notes (hereafter *Library*). All notebooks stored in the library can be used in various tests and exercises to help users to memorise its content.

Sharing & Collaboration

StudyPad also allows users to easily share these notebooks with each other. Each notebook can be shared by creating a published version of it (hereafter Published Notebook),thereby making it available to everyone else for viewing and importing. The publication process involves providing additional information about the notebook, including its name, optional description, topic and optional tags, that serve to narrow the topic. Everything Author have provided along with his school and language will then be used in the process of searching and filtering – which will facilitate the search for the necessary materials. StudyPad also allows to quickly share a notebook by sending a link, which will act as a deep-link, navigating user directly to a Published Notebook.

The Author of the notebook reserves the right to make edits/corrections to the Published Notebook. Other users (hereinafter subscribers) in turn can save a Published Notebook to their library, or suggest some changes or corrections to improve the content inside. By saving Published notebook to his/her library, subscriber will be able to make any local changes as he/she sees fit and use it as normal. Subscriber will be notified about any updates to the Published Notebook so that they can apply changes to their local versions, however, any local changes made by him/her will be deleted. The author of the notebook, in turn, will be notified about latest suggestions and comments left by the subscribers.

Chapter 1

Analysis

This chapter contains StudyPad system description and analysis with the goal to identify requirements and how it is compared with its rivals

1.1 System description

StudyPad system follows classic client-server software architecture. Server part is represented by Spring Boot application, which is communicating with it clients using REST API. Client part consists of client applications for several platforms: Android, iOS and Web. Web client is being developed alongside the Android one to serve as an Admin Panel. Main task of this thesis is to deliver an Android client.

The detailed structure of the Android client structure and how it is connected with other components is presented on the component diagram on figure 1.1.

Android component will be divided into three layers: Data layer, Domain Layer and Presentation layer. Data layer is only responsible for downloading and storing data. Business layer responsibility is to handle user interactions and converting downloaded data to something that presentation layer can present. Presentation layer is only responsible for displaying data. Android component will be also connected to Facebook SDK, Google Auth SDK and Firebase SDK to provide authentication options. Firebase SDK will also enable analytics service, crash reporting and push notifications.

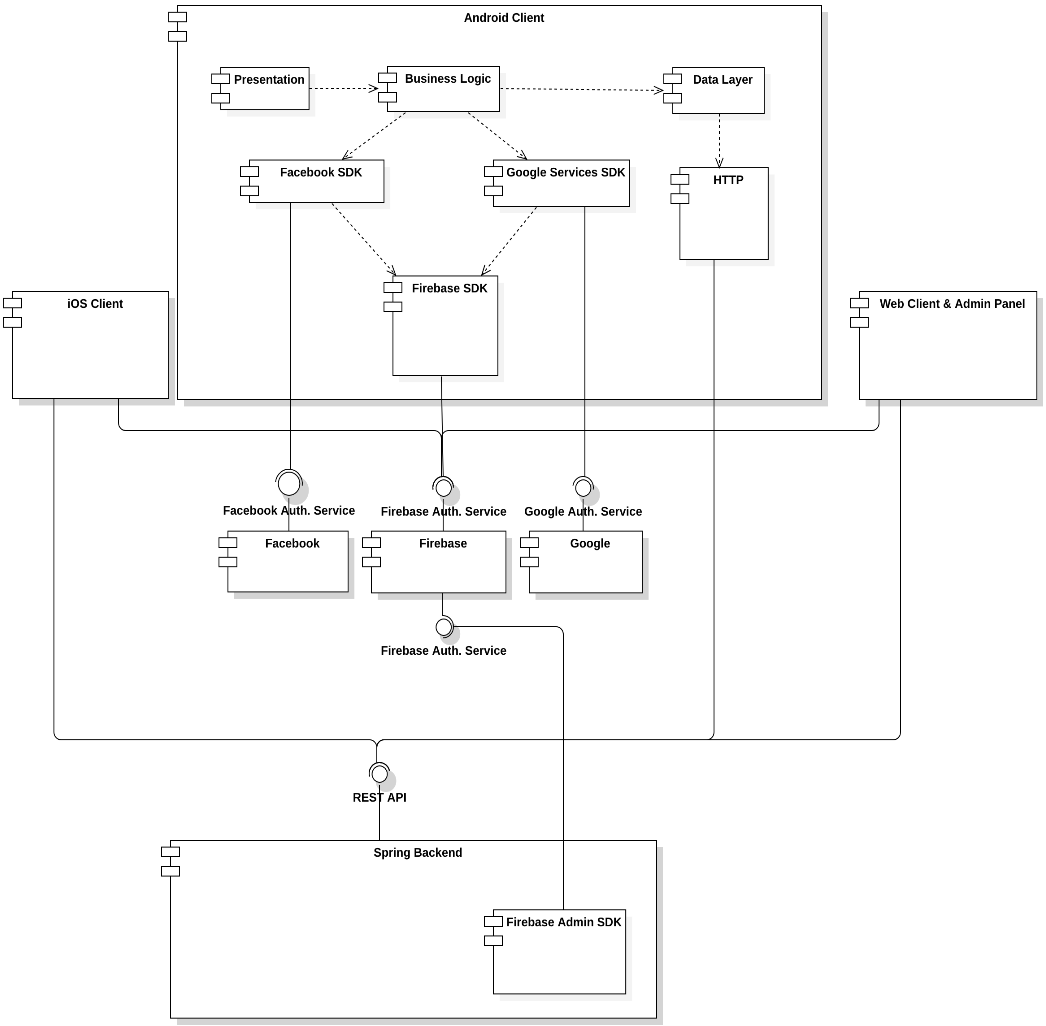

Figure 1.1: Component Diagram

1.2 Domain Description

Class Diagram on figure 1.2 represents Domain Model of the application, it provides visual representation of Entities and relations between them. Design is based on the entities used on server-side

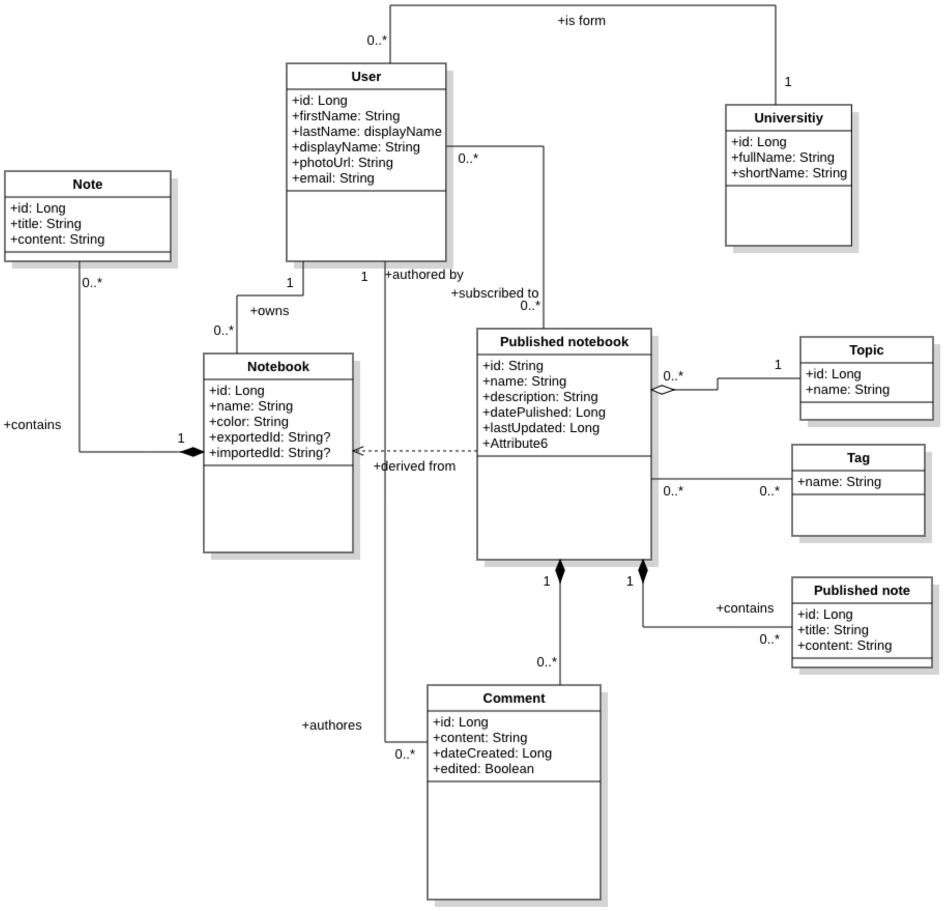

Figure 1.2: Domain Diagram

1.2.1 User

User entity represents someone who have completed registration flow using one of the client applications. This entity contains such properties as: firstName, lastName, email, password, university. Due to the fact, that StudyPad provides several ways for user to authorise, some of the properties will either come from the user's input or from the 3d party API (such as Firebase)

1.2.2 Note

Note represents a single piece of information. It consists of two properties: title and content. These can be described as term and definition or question and answer. Every note must be assigned to one of the notebooks, hence theres a 1:N relation

1.2.3 Notebook

Notebook is one of the main entities used in the application flow, and can be created by an authenticated user. Soul purpose of the Notebook is to store Notes and serve as a source for Published Notebook. Properties name and colour are used to help users distinct between different Notebooks

1.2.4 University

University represents school, where User can assign himself as a student during registration flow. It is used to unite users from the same school to make content searching easier.

1.2.5 Published Notebook

Published Notebook represents a shareable content. It can be created by user, based on one of his/her notebooks by providing some additional details: name, optional description, topic and optional list of tags. All these details are later used in a searching flow to optimise results.

1.2.6 Published Note

Published Note represent the note inside of the Published Notebook and contains the exact same properties as usual note

1.2.7 Topic

Topic represents main topic or subject of the Published Notebook. Topic consist of only one property – its name

1.2.8 Tag

Tag is a short label thats attached to the Published Notebook. It is mainly used to narrow the topic or school. Tag has only one property – its actual value stored as name

1.2.9 Comment

Users can comment on published notebooks. Most of the properties are assigned automatically, the only exception is content which is property that represents the body of the comment.

1.3 Requirements

It is important to establish all functional and non-functional requirements for StudyPad. Section bellow contains all requirements designed before the start of the development

1.3.1 Functional requirements

User Authentication

- **F1: Registration/Login using email** Access to StudyPad is possible by creating an account using email address/password combination.
- **F2: Registration/Login using Facebook** User will be able to use his/her Facebook account to access StudyPad.
- **F3: Registration/Login using Google** User will be able to use his/her Google account to access StudyPad.
- **F4: Store OAuth token** API Authentication Token will be stored in device memory.
- **F5: Token refreshment** API Token will be refreshed when needed, so user won't have to login again.
- **F6: University selection** As a part of user registration flow, user will be able to select his/her university.
- **F7: University Addition** User will be able to add his/her school, in case of not finding it in StudyPad database.

Library Management (Notes & Notebooks)

- **F7: Notebook creation** User will be able to create new notebooks with the name he/she choose.
- **F8: Notebook deletion** User will be able to delete existing notebooks.
- **F9: Notebook name edition** User will be able to edit notebooks names.
- **F10: Note creation** User will be able to create a note with specific title and content.
- **F11: Note edition** User will be able to edit existing note, or completely delete it.

- **F12: Show Notebooks** : User will be able to view all the notebooks he/she created.
- **F13: Show Notes**: By clicking on notebook item, user will be able to view the list of notes that are assigned to this notebook.
- **F14: MathJax/ASCII Math support**: User will be able to use complex mathematical expressions as notes content

Sharing Hub

- **F15: View published notebooks** User will be able to view notebooks published by other users.
- **F16: Search through published books** User will be able to search through the published notebooks by applying different filters (such as author, university and subject/topic).
- **F17: Browse through published notebook** User will be able to see notes inside the notebook that's been published.
- **F18: View comments** User will be able to view others users comments discussing a notebook that's been published.
- **F19: Leave a comment** User can comment on other user published notebook.
- **F20: Delete a comment** Application will let user to delete his/her comment.
- **F21: Edit a Comment** Application will let user to edit his/her comment
- **F21: Save published notebook** User will be able to save published notebook to his/her library.
- **F22: Publish notebook** User will be able to publish his/her notebook.
- **F23: Update published notebook** Author of the published notebook will be able to update its information.
- **F24: Delete published notebook** Author of the published notebook will be able to delete the his/her notebook from shared space.
- **F25: Share notebook** User will be able to share his/her notebook by generating a deep-link.
- **F26: Notification on update** User will be notified on updates to the Published notebook hes subscribed to

- **F27: Suggestions** Subscribers will be able to suggest a change or correction to the Published Notebook. Author will be able to approve it, which will result in the update of the Published Notebook, or decline it.
- **F28: Copy Published Notebook** User will be able to copy published notebook. This will create a standalone copy of this notebook in the users library.

Study Hub

- **F29: Start a basic self-check** User will be able to use an interactive way to look through his/her notes
- **F30: Start a written test** User will be able to participate in a written test based on one of notebooks to test his/her knowledge
- **F31: Start a quiz** User will be able to participate in quiz challenge that will be based on one of his/her notebooks

Settings

- **F32: View Profile Information** User will be able to view his/her profile information such as first name, last name and university.
- **F33: Edit Profile Information** User will be able to edit his/her profile information.
- **F34: Logout** User will be able to logout from the application.

1.3.2 Non-functional requirements

- **N1: Native Android application** Application will be written using native Android SDK.
- **N2: Android Version** Application minimal SDK version must be low enough to support as many devices as possible and high enough to use most applicable Android APIs considering other functional and non-functional requirements.
- **N3: Material Design** Application user interface will follow latest Material design guidelines and best practises.
- **N4: Scalable app architecture** Application's architecture must be scalable and easy testable.

- **N5: Tablet & Phone support** Application GUI must be well suited for multiple screen sizes.
- **N6: App Localisation** Application will be able to adapt to different languages based on user locale

1.4 Existing solutions

There are several services out there, whose goal is similar to StudyPad. However, most of the solutions are specialised for learning languages and have limited sharing and/or searching options. Each application/service will be reviewed separately and will include requirements and main processes comparison.

1.4.1 StudyBlue

StudyBlue not only partially shares a name with StudyPad but also a goal and a wide set of features. It is clearly on of the biggest rivals on the market and it is very important to determine what StudyBlue does right and what not. How most important StudyPad requirements are compared to the StudyBlue functionality can be seen on table 1.1

Notebooks here are called *decks* and notes are represented by *cards*. Moreover, all decks in StudyBlue are stored in specialised folders named Classes and Interests. Interest, as an entity, not only stores decks, but also unites them by a subject or a topic (eg Spanish, Literature). Being a student of the university, user can also create an interest, that will be connected to his/her school – Class. Both Interests and Classes are split into 2 parts, shared space, where user can found all the decks created by other users, and personal space, where user can keep decks he/she created, saved or copied.

Table 1.1: StudyPad & StudyBlue requirements comparison

Requirements / Application name	StudyPad	StudyBlue
F6: University Selection	Present	Present
F13: Show Notes	Present	Limited
F14: MathJax /ASCII Math support	Present	Limited
F16: Browse Through Published Notebooks	Present	Present
F17: Search Through The Published Notebooks	Present	Present
F18-F21: User Commentaries	Present	Absent
F27: User Suggestions	Present	Absent

- **Library management:** Main difference in library management comes with the fact, that each deck must be associated with an Interest or Class, moreover, it's not possible just to create deck, user have to create at least 2 cards first. Because of that, library management not only includes managing your decks and card, but also interests and classes. Everything user have saved or created, whether class, interest or a deck can be accessed via *Backpack*.

- **Publishing:** Publishing process is quite different comparing to what StudyPad is trying to achieve. When creating a deck, user can choose whether he wants to make his/her deck visible for other users in this Class/Interest or to make it private. It is not possible to collaborate on a deck or suggest a change

- **Importing**: As mentioned before, user can save a deck from an existing Interest/Class to his personal space, but he would not be able to make any editions without making a copy.

- **Discovering:** The fact that all decks are either assigned to an Interest or a Class makes searching quite simple. User can search for deck, classes and users by its name.All these searching options are separated in the UI, so there is no way of combining them. Searching based on the Interest is also available, but user have to add this Interest to his/her "Backpack"

- **UI & UX**: StudyBlue has been in development roughly since 2009, and is for now, UI looks outdated and UX can be greatly improved. The most common UI issue here – Inconsistent usage of UI elements styles.

1.4.2 Quizlet

Quizlet is primarily used for learning languages, from where most of the limitations are coming from. Closest analogy to a Notebook here is a Study set with Terms inside. Being an application for language learners makes the process of generating different tests and quizzes quite easy, and this is where this application shines the most. Table 1.2 shows requirements comparison between StudyPad and Quizlet

Table 1.2: StudyPad & Quizlet requirements comparison

Requirements / Application name	**StudyPad**	**Quizlet**
F6: University Selection	Present	Absent
F13: Show Notes	Present	Absent
F14: MathJax /ASCII Math support	Present	Absent
F16: Browse Through Published Notebooks	Present	Present
F17: Search Through The Published Notebooks	Present	Limited
F18-F21: User Commentaries	Present	Absent
F27: User Suggestions	Present	Limited

- **Publishing**: Similar to StudyBlue, all Study Sets are visible to everyone else by default. However, Quizlet provides a wider variety of privacy options. Study Set can either be completely public, private or semi-private using a password protection. Password protection can also be used to allow certain users modify a study set

- **Importing**: Importing flow allows user to either copy or save study set to a specific folder. This flow may confuse some of the users, because only copy allows actually add a study set to user library and modify it. Saving study set to the specific folder only saves the link to it and splits library management in 2 parts.

- **Discovering:** This limitation comes from the fact, that Quizlet is an app for studying foreign languages. As a consequence, the only distinctions between Study Sets are its name and its language. These are the only 2 options available when searching through study sets.

1.4.3 Cram

Cram is very similar to Quizlet but seems highly outdated in terms of UX/UI and brings some sharing limitations to the table. Table 1.3 shows requirements comparison between StudyPad and Cram

Table 1.3: StudyPad & Cram requirements comparison

Requirements / Application name	**StudyPad**	**Cram**
F6: University Selection	Present	Absent
F13: Show Notes	Present	Present
F14: MathJax /ASCII Math support	Present	Limited
F16: Browse Through Published Notebooks	Present	Present
F17: Search Through The Published Notebooks	Present	Limited
F18-F21: User Commentaries	Present	Absent
F27: User Suggestions	Present	Absent

- **Publishing:** Content publishing is similar to Quizlet – All sets are either visible by other users or not. Sharing a deep-link to a single study set was not functional at the time of writing this section.

- **Discovering** Searching for content in Cram is even more limited comparing to Quizlet, only name of the study set is used

- **Importing** Library management here is split in 3 parts: User personal sets, Favourite sets, and Recently studied. When searching, there is no way to save published study set to personal library, however, it will be automatically saved to Recent section, or user can add it manually to Favourites. It is not possible to make any local editions.

1.4.4 TinyCards

TinyCards is an application made by Duolingo, one of the biggest app for learning languages. TinyCard is meant to be more generic as it offers users to create custom study sets, often not limited to languages. Table 1.4 shows how StudyPad requirements are compared to TinyCards functionality

Table 1.4: StudyPad & TinyPads requirements comparison

Requirements / Application name	**StudyPad**	**TinyCards**
F6: University Selection	Present	Absent
F13: Show Notes	Present	Present
F14: MathJax /ASCII Math support	Present	Absent
F16: Browse Through Published Notebooks	Present	Present
F17: Search Through The Published Notebooks	Present	Limited
F18-F21: User Commentaries	Present	Absent
F27: User Suggestions	Present	Absent

- **Importing:** Similar to Cram, it is not possible to edit the study set user have downloaded and saved to his library

- **Challenges:** Tests are generated automatically and there is no way to choose test type

1.5 Android Platform

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialised user interface.

Since its inception several versions have been released, with the latest one being 9.0, codenamed “Pie”. Every new version provides new and improved APIs to work with and backward compatibility for applications running on older versions. Originally, the Support Libraries were used to provide backward compatibility while targeting newer versions. This was a case until AndroidX was introduced. AndroidX is a major improvement to the original Android Support Library. Like the Support Library, AndroidX ships separately from the Android OS and provides backwards-compatibility across Android releases.

When developing applications for Android, developer has to consider which OS versions application will support. Specifically, the minSdkVersion and targetSdkVersion attributes identify the lowest API level with which your application is compatible and the highest API level against which you’ve designed and tested your application. StudyPad will target the latest Android version available, with the minimal version set to 5.0, codenamed Android Lollipop. This will ensure high compatibility rate of 85% and Material Design Support, which is one of non-functional requirements.

1.6 Firebase

Firebase is Google's mobile platform that helps to quickly develop high-quality applications and grow businesses. Firebase provides high number of services, some of which are a necessity in today's world of software development. StudyPad will be using following services:

- **Firebase Analytics** is a cost-free app measurement solution that provides insight into app usage and user engagement.
- **Firebase Cloud Messaging** Formerly known as Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) is a cross-platform solution for messages and notifications for Android, iOS, and web applications, which as of 2016 can be used at no cost.
- **Firebase Auth** is a service that can authenticate users using only client-side code. It supports social login providers Facebook, GitHub, Twitter and Google (and Google Play Games). Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.
- **Crashlytics** Crash Reporting creates detailed reports of the errors in the app. Errors are grouped into clusters of similar stack traces and triaged by the severity of impact on app users.

While Firebase Analytics and Crash Reporting services are a great help while developing a mobile applications, Firebase Auth and FCM are essential for some of the functional requirements.

Storing user credentials, and most importantly, storing it safely is not an easy task. With a little effort, Firebase Auth will handle everything related to user management, including access token generation and refreshment, and credentials storing and encryption.

Firebase Cloud Messages will enable push-notifications, which is a great way to communicate with users and introduce some real-time functionality to the application.

Chapter 2

Design

2.1 Wireframes

2.2 Application architecture

2.3 Platform-specific model

2.4 Main sequence diagrams

Chapter 3

Implementation

3.1 Choice of technologies

3.2 Component diagram

3.3 Installation

Chapter 4

Testing

Conclusion

Appendix A

Acronyms

GUI Graphical user interface

XML Extensible markup language

Appendix B

Contents of enclosed CD

- `readme.txt` ... the file with CD contents description
- `exe` ... the directory with executables
- `src` ... the directory of source codes
  - `wbdcm` ... implementation sources
  - `thesis` ... the directory of LaTeX source codes of the thesis
- `text` ... the thesis text directory
  - `thesis.pdf` ... the thesis text in PDF format
  - `thesis.ps` ... the thesis text in PS format