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

February 6, 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 February 6, 2019 .....

Czech Technical University in Prague
Faculty of Information Technology


### 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 socíální 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 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 apps are one of the most popular apps 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 easily. 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 collaborate to their study 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 library can be

used in various tests and exercises to help users to memorise its content.

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 other users for viewing and downloading . 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. All this data along with Author's school will then be used in the process of searching and filtering - which will facilitate the search for the necessary materials. There is also a functionality to quickly send a notebook using its link. In this case, if the notebook has not yet been published, its published version will be created with a minimum amount of details.Such notebook will be also excluded from the search results, making it accessible only by using its link, until all necessary details are provided. Link to the already created Published version will be sent otherwise

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 the author changes/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 sees fit and use it as normal. Subscribers will be notified about any updates to the Published Notebook so that they can apply changes to their local versions, though any local changes made by a subscriber will be deleted. The author of the notebook, in turn, will be notified about latest suggestions and comments left by subscribers

Chapter 1

# Analysis

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

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. Main task of this thesis is to deliver an Android application. iOS application is developed as a part subject called BI-IOS and hence, has some limitations in its implementation. Web client is being developed alongside the Android one is serves as Admin Panel for the backend which allows to view and modify and certain data without dealing with the server-side database directly.

The detailed structure of the Android client structure and its connection with other components is presented in 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 and Google Auth SDK to provide authentication options and Firebase SDK to enable analytics service, crash reporting and push notifications.

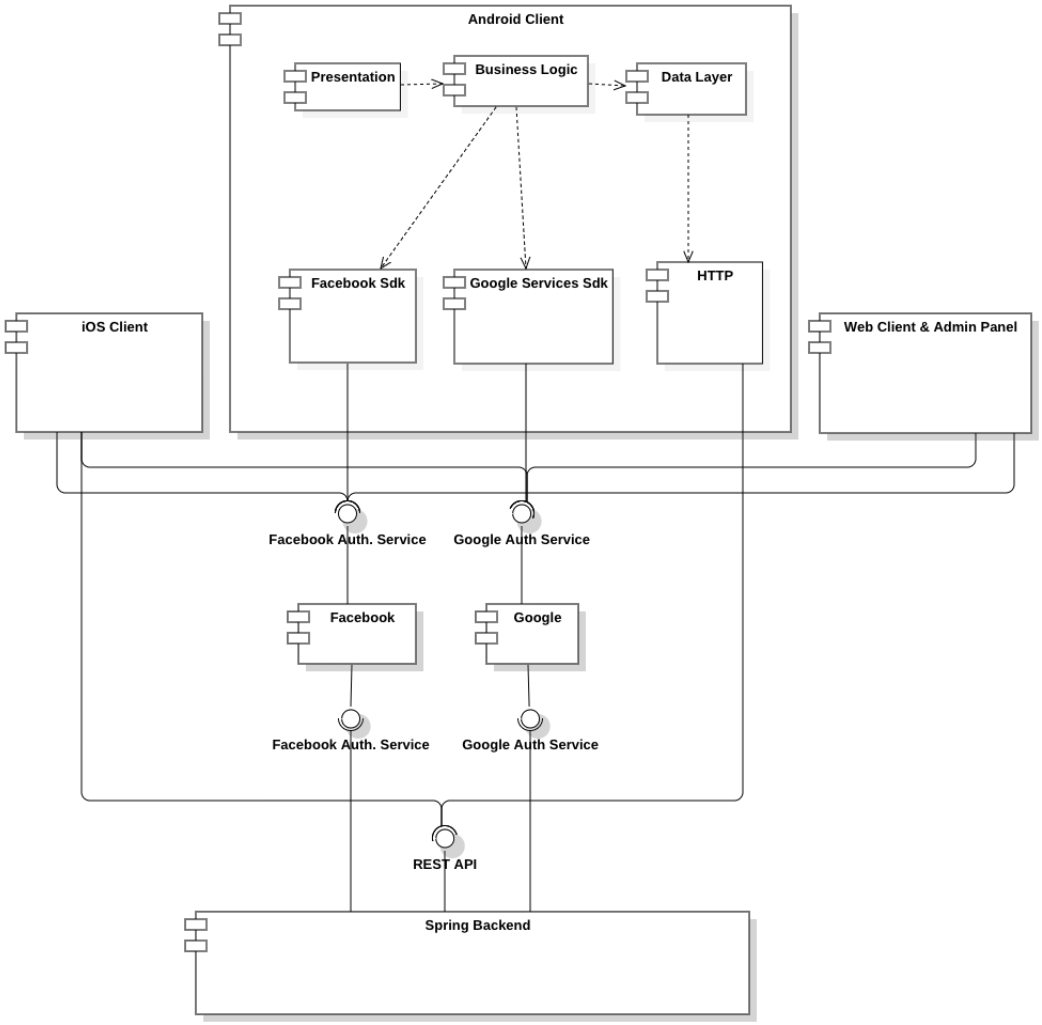


Figure 1.1: Component Diagram

## 1.1 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

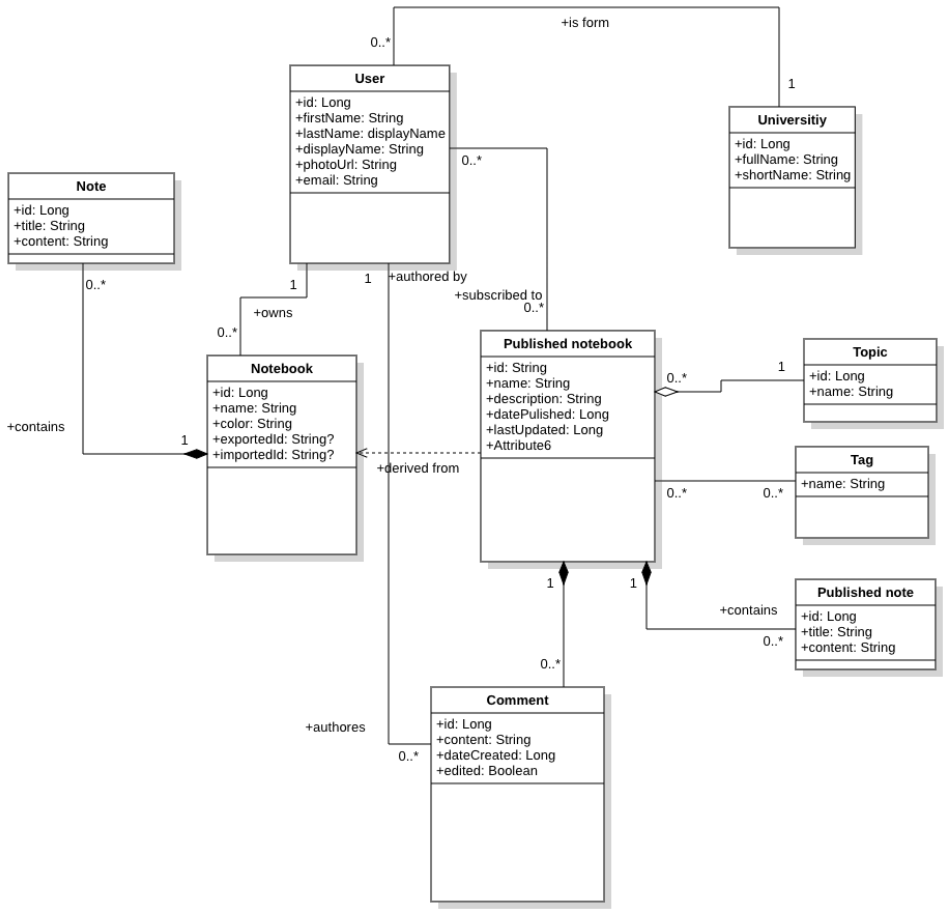


Figure 1.2: Domain Diagram

### 1.1.1 User

User entity represents someone who have completed registration flow using one of the client app. 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 3rd party API ( Google or Facebook ).

### 1.1.2 Note

Note represent 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

### 1.1.3 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 color are used to help users distinct between different Notebooks

### 1.1.4 University

University represents school, where User can assign himself as a student during registration flow. It is used to unite users from the same schools, so they could faster find content they are looking for.

### 1.1.5 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 set of Tags. All these details are later used for Search flow to optimise searching results.

### 1.1.6 Published Note

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

### 1.1.7 Topic

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

### 1.1.8 Tag

Tag is us short label thats attached to the Published Notebook. It is mainly used to narrow the topic or school. Tag has only one property - it's actual value stored as name

### 1.1.9 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. All other properties are assigned automatically and can not be changed

## 1.2 Android Platform

## 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 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: 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:q** User will be notified on updates to the Published notebook he subscribed to
- **F27: Suggestions** Subscribers will be able to suggest a change or correction to the Published Notebook

- **F28: Copy Published Notebook** User will be able to copy published notebook. This will

**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 limited to learning languages and have limited sharing and/or searching options. Table on figure 1.4 represents key requirements comparison

| **Application Name Requirement** | StudyPad | Quizlet | Cram | TinyCards |
|---|---|---|---|---|
| **F6: University Selection** | Present | Absent | Absent | Absent |
| **F14: MathJax support** | Present | Absent | Absent | Limited |
| **F16: Searching and Filtering** | Present | Limited | Limited | Limited |
| **F18-F20: User commentaries** | Present | Absent | Absent | Absent |
| **F21: Save published notebook** | Present | Present | Present | Present |
| **F25: Share published notebook** | Present | Present | Limited | Present |
| **F27: Suggestions** | Present | Absent | Absent | Absent |

Table 1.1: Key requirements comparison

### 1.4.1 Quzlet - Key differences

**Quizlet** is primarily used for learning languages, from where most of the limitations come from. Closest analogy to Notebook there is Study set with Terms inside. This makes it easier for tests generation, but limits user when he/she is trying to learn anything other than new words

- **Publishing**: Content publishing process is very different to what StudyPad is trying to achieve. All study sets are visible to other users by default, which makes it hard, if not impossible, to distinct between local and shared published 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 new languages. As a consequence, the only distinctions between Study sets are its name and a language. These are the only 2 options available for when searching through published study sets.

### 1.4.2 Cram - Key differences

**Cram** is very similar to Quizlet but seems much more outdated in terms of UX and brings some sharing limitations to the table.

- **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. This makes it impossible to make local edits

### 1.4.3 TinyCards - Key Differences

**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

- **Importing:** Similar to Cram, it is not possible to edit the study set user have downloaded and saved to his library
- **Challenges:** Tests for user are generated automatically and there is no way to choose test type

Chapter 2

# Design

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