

Insert here your thesis' task.



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Bachelor's thesis

StudyPad - Android Client

Roman Levinzon

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

March 22, 2019

Acknowledgements

THANKS (remove entirely in case you do not wish 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 22, 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 porízkování poznámek a sociální sítě s cílem pomoci studentům zapamatovat si různé informace. Cílem práce je vyvinout aplikaci pro OS Android, která bude sloužit jako klient. Tento text uznává stávající řešení, obsahuje analýzu domén a požadavků, popis a výběr architektury aplikace a její implementace

Klíčová slova Android, Kotlin, MVVM, Clean architecture

Abstract

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

Keywords Android, Kotlin, MVVM, Clean architecture

Contents

Introduction	1
StudyPad Service	1
1 Analysis	3
1.1 System description	3
1.2 Domain Description	6
1.3 Requirements	8
1.4 Existing solutions	12
1.5 Android Platform	15
1.6 Firebase	17
2 Design	19
2.1 Wireframes	19
2.2 Application architecture	20
2.3 Platform-specific model	20
2.4 Main sequence diagrams	20
3 Implementation	21
3.1 Choice of technologies	21
3.2 Component diagram	21
3.3 Installation	21
4 Testing	23
Conclusion	25
A Acronyms	27
B Contents of enclosed CD	29

List of Figures

1.1	Component Diagram	5
1.2	Domain Diagram	6
1.3	Android Versions distribution	16

List of Tables

1.1	StudyPad & StudyBlue requirements comparison	12
1.2	StudyPad & Quizlet requirements comparison	13
1.3	StudyPad & Cram requirements comparison	14
1.4	StudyPad & TinyPads requirements comparison	15

Introduction

Each year, our smartphones get smarter and mobile applications become more advanced. The arrival of smartphones and mobile applications has completely changed our way of living, made it easier, and it is hard to come up with a single aspect of life, that 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. In addition, some of the most popular applications tend to teach us something new.

Educational applications are very popular on both mobile platforms (iOS and Android). Many of them use a flash card system in the study process – displaying small pieces of information one after another, so that the user can 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. Using Android OS, today's most popular mobile platform, StudyPad wants to give its users freedom in what they can 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. The primary focus of StudyPad is on content creation, its sharing and discovery.

The StudyPad project originated from several subjects on Faculty of Information Technology (FIT) in Czech Technical University (CTU). The first iteration of backend was introduced as a semestral project for subject Enterprise Java (BI-EJA), and subject Introduction to Android Development (BI-AND) had great influence on the first iteration of the Android Client. All of these components will be described more thoroughly in Chapter 1.

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 they 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, an optional description, topic and optional tags that serve to narrow the topic. Everything the author has provided along with their 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 makes it possible to quickly share a notebook by sending a link, which will act as a deep-link, navigating the 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. By saving Published notebook to his/her library, the subscriber will be able to make any local changes as he/she sees fit and use it as normal. The Subscriber will be notified about any updates made to the Published Notebook so that he/she can apply the changes to his/her local version, however, any local changes will be deleted. The Author of the notebook, in turn, will be notified about latest suggestions and comments left by the subscribers.

Analysis

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

1.1 System description

The StudyPad system follows classic client-server software architecture. The server part is represented by the Spring Boot application, which communicates with its clients using REST API. The client part consists of client applications for several platforms: Android, iOS and Web. The Web client is being developed alongside the Android client to serve as an Admin Panel. The iOS client was developed within the subject Introduction to iOS Development (BI-IOS). The main task of this thesis is to deliver an Android OS client.

1.1.1 User Roles

Some of the use-cases, such as university creation, requires moderation. The simplest example is a university creation use-case. Though it is possible to submit new school/university in case user haven't found it, the name, that user has submitted has to be verified first and other details, like school location and translations should be updated. This is a main purpose of the Admin Panel and it will require an introduction of different user roles: User and Admin.

The User role is a default one and will be assigned automatically to everyone who have completed the registration process. It will only be used in the client applications. Admin role is the only one that have the access to the admin panel and an ability to view and modify certain data.

1.1.2 StudyPad REST API

StudyPad has its own REST API is represented by Spring Boot application. The Spring Framework is an application framework and inversion of control

container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform.

1.1.3 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: the Data layer, Domain Layer and Presentation layer. The Data layer is only responsible for downloading and storing data. The 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 also connected be to Facebook SDK, Google Auth SDK and Firebase SDK to provide authentication options. Firebase SDK will also enable an analytics service, crash reporting and push notifications.

1.1. System description

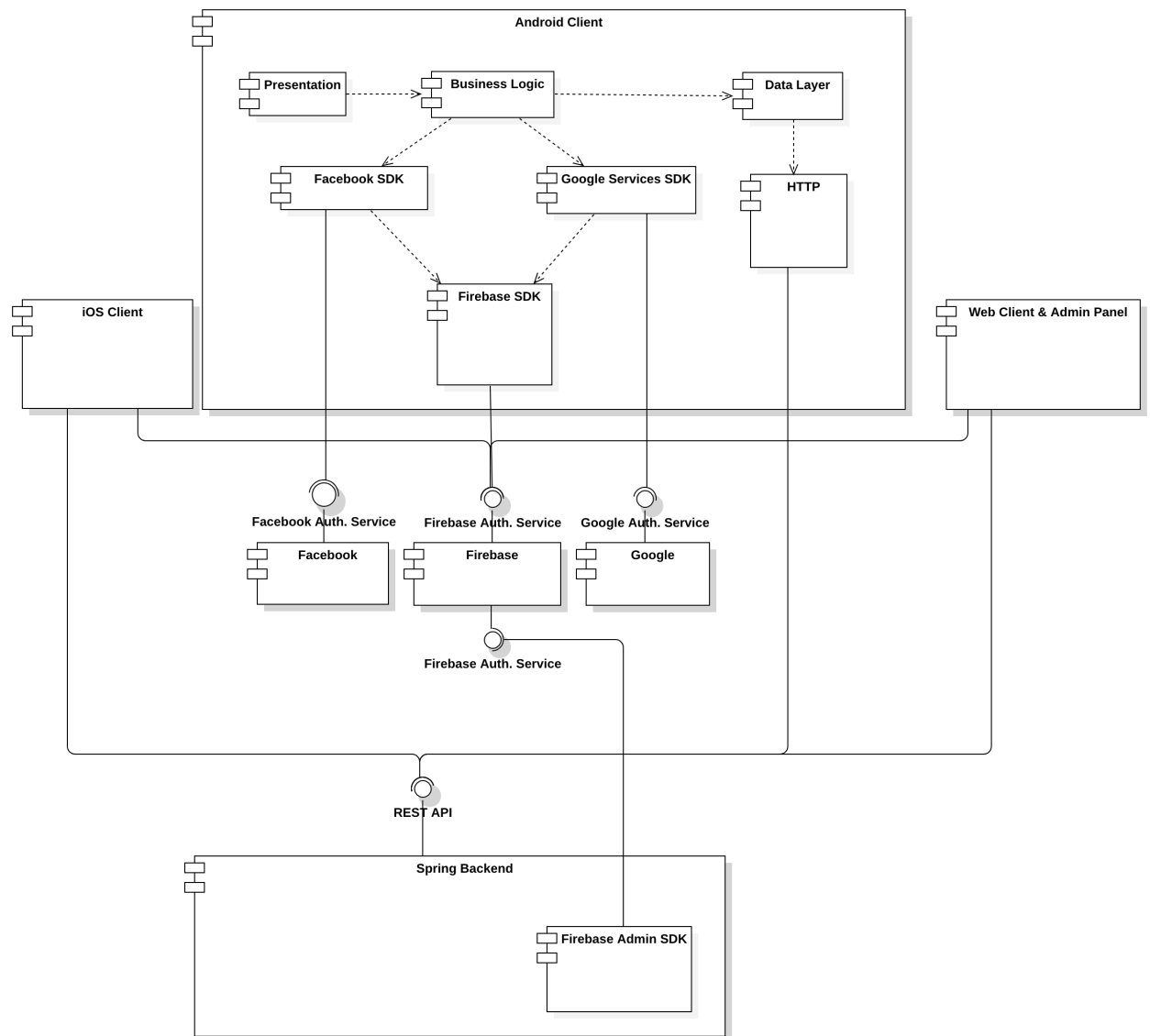


Figure 1.1: Component Diagram

1.2 Domain Description

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

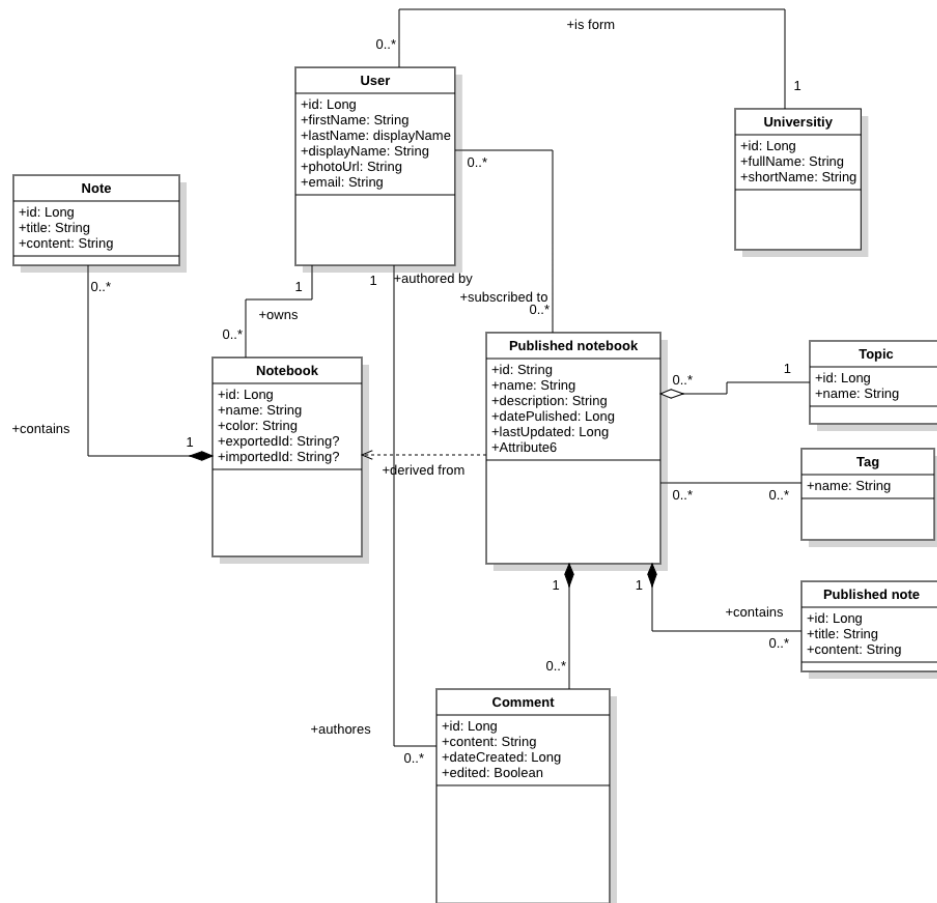


Figure 1.2: Domain Diagram

1.2.1 User

The User entity represents someone who has completed registration flow using one of the client applications or created manually (in case of Admin role). This entity contains such properties as: role, first name, last name, email address, password, university. Due to the fact, that StudyPad provides several ways for the 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. Each note must be assigned to one of the notebooks, hence there is 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. The main purpose of the Notebook is to store Notes and serve as a source for a Published Notebook. Properties name and colour are used to help users distinguish between different Notebooks.

1.2.4 University

University represents a school, where the 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 Topic

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

1.2.7 Tag

Tag is a short label that is 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 the name.

1.2.8 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. The section bellow contains all requirements designated 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 an email address/password combination.
- **F2: Registration/Login using Facebook** The User will be able to use his/her Facebook account to access StudyPad.
- **F3: Registration/Login using Google** The ser will be able to use his/her Google account to access StudyPad.
- **F4: Store OAuth token** API Authentication Token will be stored in a device memory.
- **F5: Token refreshment** API Token will be refreshed when needed, so the user will not have to login again.
- **F6: University selection** As part of user registration flow, the user will be able to select his/her university.
- **F7: University Addition** The user will be able to add his/her school, in case of not finding it in a StudyPad database.

Library Management (Notes & Notebooks)

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

- **F12: Show Notebooks:** The user will be able to view all the notebooks he/she created.
- **F13: Show Notes:** By clicking on notebook item, the 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** The user will be able to view notebooks published by other users.
- **F16: Search through published books** The 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 have been published.
- **F18: View comments** User will be able to view others users comments discussing a notebook that have been published.
- **F19: Leave a comment** The user can comment on an other user published notebook.
- **F20: Delete a comment** The application will allow the user to delete his/her comment.
- **F21: Edit a Comment** The application will allow the 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** The Author of the published notebook will be able to update its information.
- **F24: Delete published notebook** The Author of the published notebook will be able to delete the his/her notebook from shared space.
- **F25: Share notebook** The user will be able to share his/her notebook by generating a deep-link.
- **F26: Notification on update** The 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 one 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 find 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

1. ANALYSIS

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

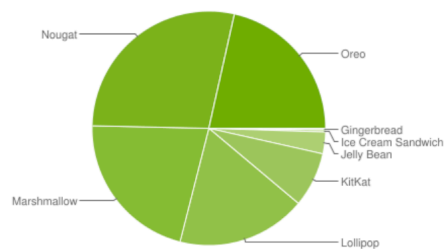
1.5 Android Platform

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.

1. ANALYSIS

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.1%
4.2.x		17	1.5%
4.3		18	0.4%
4.4	KitKat	19	7.6%
5.0	Lollipop	21	3.5%
5.1		22	14.4%
6.0	Marshmallow	23	21.3%
7.0	Nougat	24	18.1%
7.1		25	10.1%
8.0	Oreo	26	14.0%
8.1		27	7.5%



Data collected during a 7-day period ending on October 26, 2018 (update coming soon: data feed under maintenance).

Figure 1.3: Android Versions distribution

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.

Design

This chapter describes how StudyPad will comply with its requirements. This includes reviewing of application UI & UX standpoint and choice of applications architecture

2.1 Wireframes

2.1.1 Navigation Principle

How user will be able to navigate through your applications different screens and section is one of most important choices when developing a mobile applications. Material Design describes 3 types of navigation hierarchies that can be used for a top-level navigation

- Bottom navigation bar – Bottom navigation bars provide access to 2-5 top-level destinations on mobile devices. Their location, visibility, and persistence across screens allow quick pivoting between destinations.
- Navigation drawer – Navigation drawers are appropriate for five or more top-level destinations, and can be used across device sizes for a consistent navigation experience.
- Tabs – Tabs can be used at any level of an application's hierarchy to present two or more peer sets of data across screen sizes.

StudyPad application has 4 main independent sections: My Library, Sharing Hub, Challenges and Profile making the Bottom Navigation bar the perfect choice. It will enable quick and persistent navigation between different sections.

2.1.2 Top-level sections

Following application sections are used for top level navigation. They must provide access to application core functionalities

My Library section covers note-taking service part of the application. Its entry point contains a list of Notebooks and provides different actions related to the notebook management, such as: Notebook creation, edition, deletion and sharing. By clicking on one of the notebooks in the list, user will be transferred to the Notes subsection, that will contain a list of notes from this notebook. Like Notebooks section, Notes section covers all the note management related actions and interactions such as Note creation, edition and deletion.

Sharing Hub section features list of Published Notebooks. One of the main responsibilities of this section is to enable Searching & Filtering flow.

Challenges section serves as an entry point for setting up different tests based on the user notebooks

Profile sections provides various user-important information and allows user to modify them.

2.2 Application architecture

2.3 Platform-specific model

2.4 Main sequence diagrams

Implementation

- 3.1 Choice of technologies
- 3.2 Component diagram
- 3.3 Installation

Testing

Conclusion

Acronyms

GUI Graphical user interface

XML Extensible markup language

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