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

April 7, 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 April 7, 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, 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 it's implementation.

Keywords Android, Kotlin, MVVM, Clean architecture

Contents

List of Figures

List of Tables

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 most popular 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. StudyPad is a new solution to this problem. 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.

Goal

The goal of this thesis is to deliver StudyPad client application for Android OS. This includes following steps:

- analysis of the existing solutions,
- analysis of the functional and non-functional requirements,
- requirements design and its implementation,
- testing of the resulted solution.

Motivation

The primary motivation is to practice software development processes on the big and scalable project while applying advanced practices and technics of Android Development.

State-of-the-art

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 their 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 user has provided(hereafter Author) 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.

Chapter 1

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.[1] The iOS client was developed by the author of this thesis within the subject Introduction to iOS Development (BI-IOS). [2] The main task of this thesis is to deliver an Android OS client.

1.1.1 User Roles

Some of the use-cases 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 completes 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 that is represented by Spring Boot application. [3] *"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."* [4]

1.1.3 Android client

The detailed structure of the Android client and how it is connected with other components is presented on the component diagram on figure 1.2.

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.

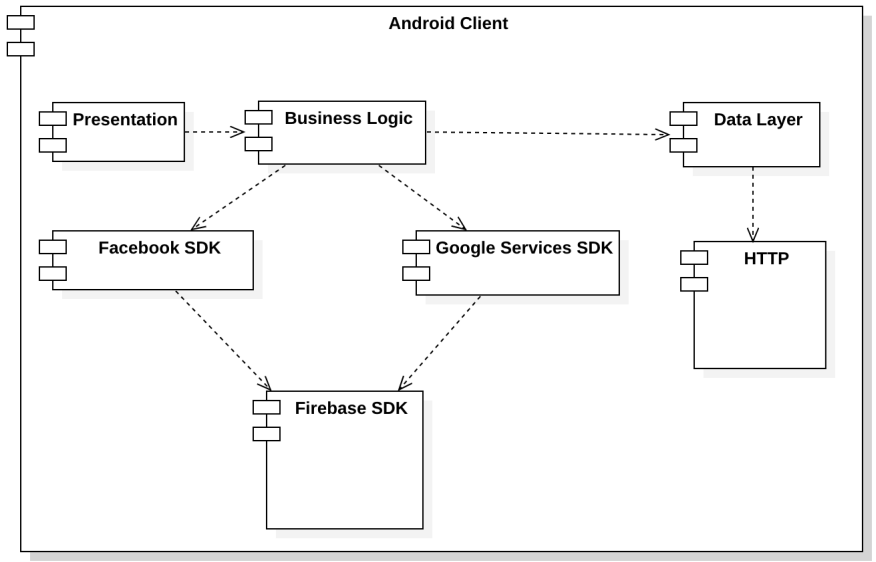


Figure 1.1: Android component Diagram

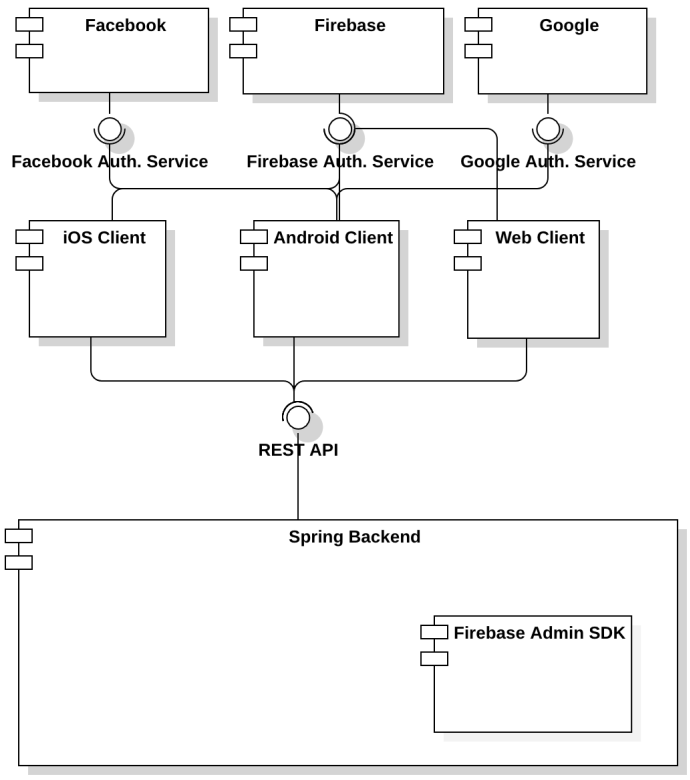


Figure 1.2: Component Diagram

1.2 Domain Description

The Class Diagram on figure 1.3 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

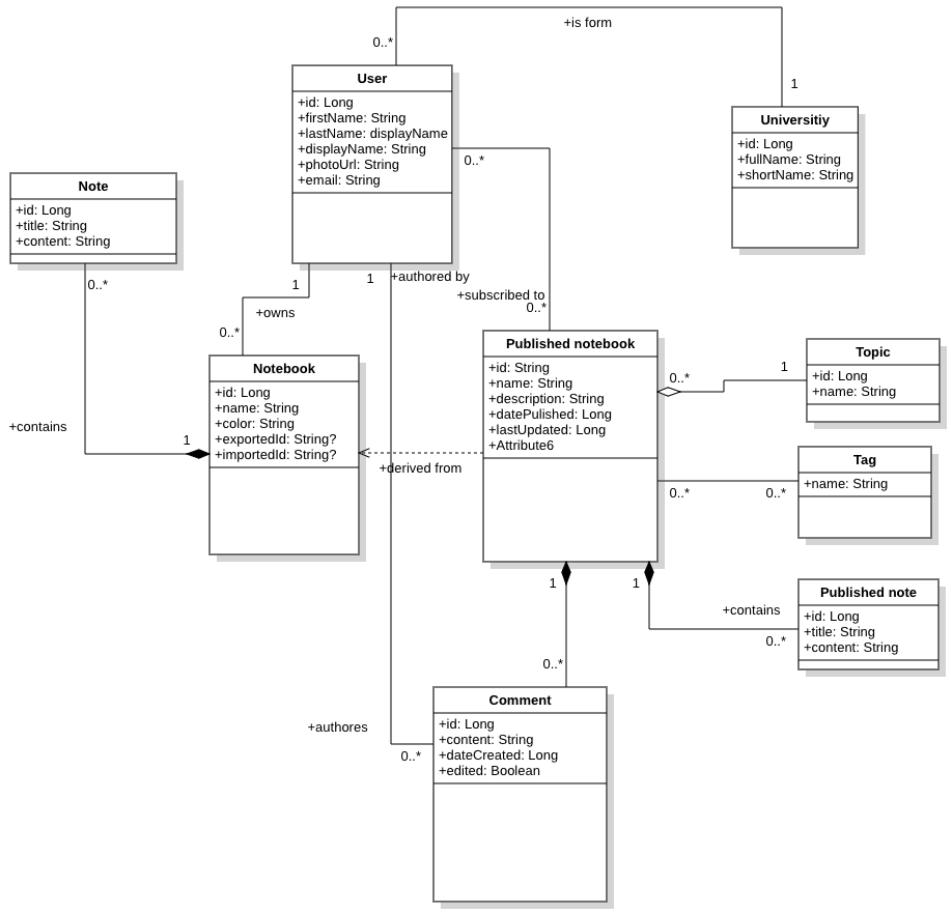


Figure 1.3: Domain Diagram

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 soul 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 properties 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 contains only one property – its name

1.2.7 Tag

Tag is a short label 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, with the only exception being content: it represents the body of the comment and assigned by the user.

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 user 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 has 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 they have subscribed to.

- **F27: Suggestions** Subscribers will be able to suggest a change or correction to the Published Notebook. The author will be able to approve it, which will result in the update of the Published Notebook, or decline it.

- **F28: Copy Published Notebook** The user will be able to copy a published notebook. This will create a standalone copy of this notebook in the user's library.

Study Hub

- **F29: Start a basic self-check** The user will be able to use an interactive method of looking through his/her notes.

- **F30: Start a written test** The user will be able to participate in a written test based on one of the notebooks to test his/her knowledge.

- **F31: Start a quiz** The user will be able to participate in a quiz challenge that will be based on one of his/her notebooks

Settings

- **F32: View Profile Information** The user will be able to view his/her profile information such as first name, last name, and university.

- **F33: Edit Profile Information** The user will be able to edit his/her profile information.

- **F34: Logout** The user will be able to logout from the application.

1.3.2 Non-functional requirements

- **N1: Native Android application** The application will be written using native Android SDK.

- **N2: Android Version** The 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** The application user interface will follow the latest Material design guidelines and best practises.

- **N4: Scalable app architecture** The application's architecture must be scalable and easy testable.

- **N5: Tablet & Phone support** The application GUI must be well suited for multiple screen sizes.
- **N6: App Localisation** The 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 those solutions are specialised in language-learning and have limited sharing and/or searching options. Each application/service will be reviewed separately and will include a 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 range 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 it does 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 (e.g. Spanish, Literature). Being a student of the university, the user can also create an interest that will be connected to his/her school – Class. Both Interests and Classes are split into two parts, shared space, where the user can find all the decks created by other users, and personal space, where the user can keep the decks he/she has created, saved or copied.

Table 1.1: StudyPad & StudyBlue requirements comparison

Requirements / Application name	StudyPad	StudyBlue
F6: University Selection	★	★
F13: Show Notes	★	⯪
F14: MathJax /ASCII Math support	★	⯪
F16: Browse Published Notebooks	★	★
F17: Searching Through Published Notebooks	★	★
F18-F21: User Commentaries	★	☆
F27: User Suggestions	★	☆

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

- **Publishing:** The Publishing process is quite different compared to what StudyPad is trying to achieve. When creating a deck, the user can choose whether they 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 their personal space, but they will not be able to make any edits without making a copy.
- **Discovering:** The fact that all decks are either assigned to an Interest or a Class makes searching quite simple. The 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 the user has to add this Interest to his/her "Backpack".
- **UI & UX**: StudyBlue has been in development roughly since 2009, and as 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 come from. The 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 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 to modify a study set.

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

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

1.4.4 TinyCards

TinyCards is a flash-card application made by Duolingo, one of the biggest applications for learning languages. TinyCard is meant to be more generic as it

alows users to create custom study sets, often not limited to languages. Table 1.4 shows how StudyPad requirements 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 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 the user has downloaded and saved to their library

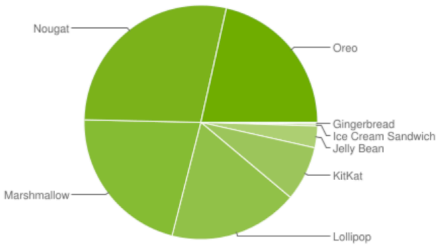
- **Challenges:** Tests are generated automatically and there is no way to choose the test type.

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 the 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."* [5]

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 1.4: Android Versions distribution

When developing applications for Android, the developer has to consider which OS versions the 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 have 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 a high compatibility rate of 85% and Material Design Support, which is one of the non-functional requirements.

1.6 Firebase

"Firebase lets you build more powerful, secure and scalable apps, using world-class infrastructure." [6] Firebase provides a high number of services, some of which are a necessity in today's world of software development.
StudyPad will be using the 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 mobile applications, Firebase Auth and FCM are essential for some of the functional requirements.

Storing user credentials, and most importantly, storing them 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, as well as credentials storing and encryption.

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

Chapter 2

Design

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

2.1 Material Design

Material Design is a design language that Google developed in 2014. Later in 2018, it was revamped with a goal to provide more flexibility for designers to create more custom themes. Material Design was chosen as a main source of inspiration when creating StudyPad UI due following reasons:

- Material Design has very wide support on Android Platform. Every component that is described in Material Design Guidelines is implemented by Google and can be used by developers.
- Material Design is cross-platform, meaning that its guidelines can easily be applied when creating design for other platforms like iOS or Web.

2.1.1 Used Components

This section contains a list of Material Design specific Components (not including widely used elements such as Buttons and Input Fields) used when creating design for StudyPad.

- **Floating Action Button (FAB)**: According to Material Design Guidelines: *"A floating action button (FAB) performs the primary, or most common, action on a screen. It appears in front of all screen content, typically as a circular shape with an icon in its center."* [7] FAB is widely used in creating content, hence it found its place in the note-taking part of the application.

- **Bottom App Bar**: *"Bottom app bars provide access to a bottom navigation drawer and up to four actions, including the floating action button."* [8]. In StudyPad it is primarily used in the details screens with one primary action (FAB) and two or more secondary actions.
- **Chips**: *"Chips are compact elements that represent an input, attribute, or action."*[9]. In StudyPad it is used mainly for displaying Tags and in Searching flow. In most of the cases it will be placed in the ChipGroup – a special container to store dynamic amount of Chips

2.2 UI & UX

This section contains most important screens description from UI & UX standpoint, wireframes and it's iterations

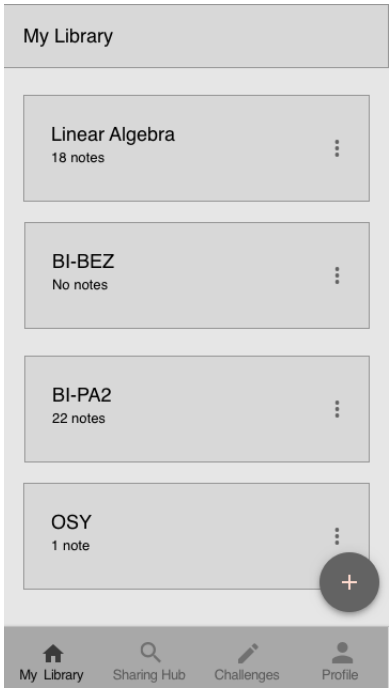
2.2.1 Library Managent

Library management requirements (F7-F14) are covered by the first top-level destination: **My Library**. This part of the application serves as the main entry point for the returning users. Wireframes of this sections can be viewed on Figure 2.1.

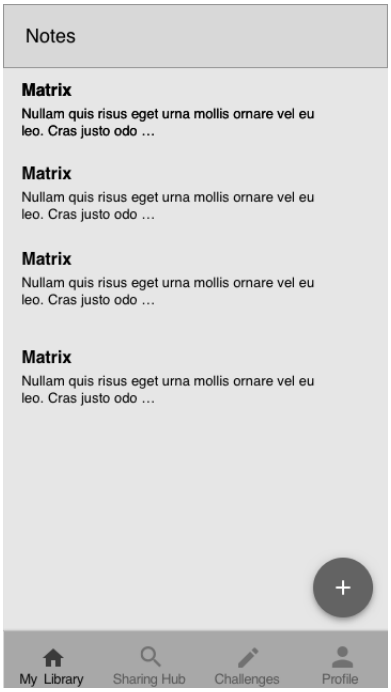
Its first screen, as shown on Figure 2.1a, contains a scrollable list of Notebooks and provides access to the various notebook-management related actions such as: Notebook creation(triggered by FAB), edition, deletion and sharing

By clicking on one of the notebooks in the list, user will be transferred to the Notes List Screen (Figure 2.1b). Similar to Notebooks List Screen, this screen contains a scrollable list of notes from this notebook and covers all the note management related actions and interactions such as Note creation, edition and deletion. Once again, FAB is used for to trigger create action.

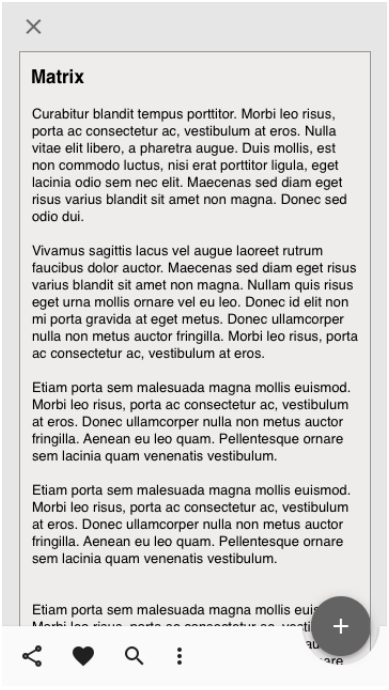
Finally, by choosing a note, the user will be able to view its details. As shown of Figure 2.1c, this screen utilises Bottom app bar to provide access for key actions and FAB.



(a) Notebooks List Screen



(b) Notes List Screen



(c) Note Detail Screen

Figure 2.1: My Library Section Wireframes

2.2.2 Notebook Publishing

The fact the all the content in StudyPad is made by its community makes Notebook Publishing Flow is one of the core functionalities of the application. The flow it self acts as complex form with different types of inputs:

- Notebook name – Single-line input field, will be pre-filled with the name of local Notebook that is being published.
- Language of the Notebook – Selector, will be pre-filled with the current user locale.
- University/School associated with this Notebook – Selector, will be pre-filled with the User's university, this field is optional.
- Topic/Category of the Notebook – Selector, this field is mandatory
- Tags – ChipGroup, chosen tags will be represented by Chips. The field is optional
- Description/Message for other users – Multi-line input field, this field is optional.

Considering an amount of input needed, its variety, the decision was made to split this form into several steps. It will allow make the form more dynamic and put more emphasis on some inputs by grouping them.

First step will include all fields that can be pre-filled: Notebook's name, language and University. This will allow to skip this step in most cases, making it only about controlling pre-filled information. Second step will require users to provide search-relevant information, such as Tags and Topic/Category. Third step will allow user to provide a message or a more detailed of description of the Notebook. Fourth step will be all about confirmation, in which the user can double-check information he/she entered and submit it.

Though the choice of UI elements in different steps is well defined, they way we present these steps is not. Simplified wireframes with different approaches of how the steps can be presented are shown on Figure 2.2. In the end, decision to go with the Stepper solution was made as it gives a number of advantages:

- Stepper is a standardised element of Material Design
- In every step, Stepper gives an overview of all steps available and its completion state
- Stepper allows to quickly navigate between different steps.

More detailed wireframes for this flow using Stepper are shown of Figure 2.3

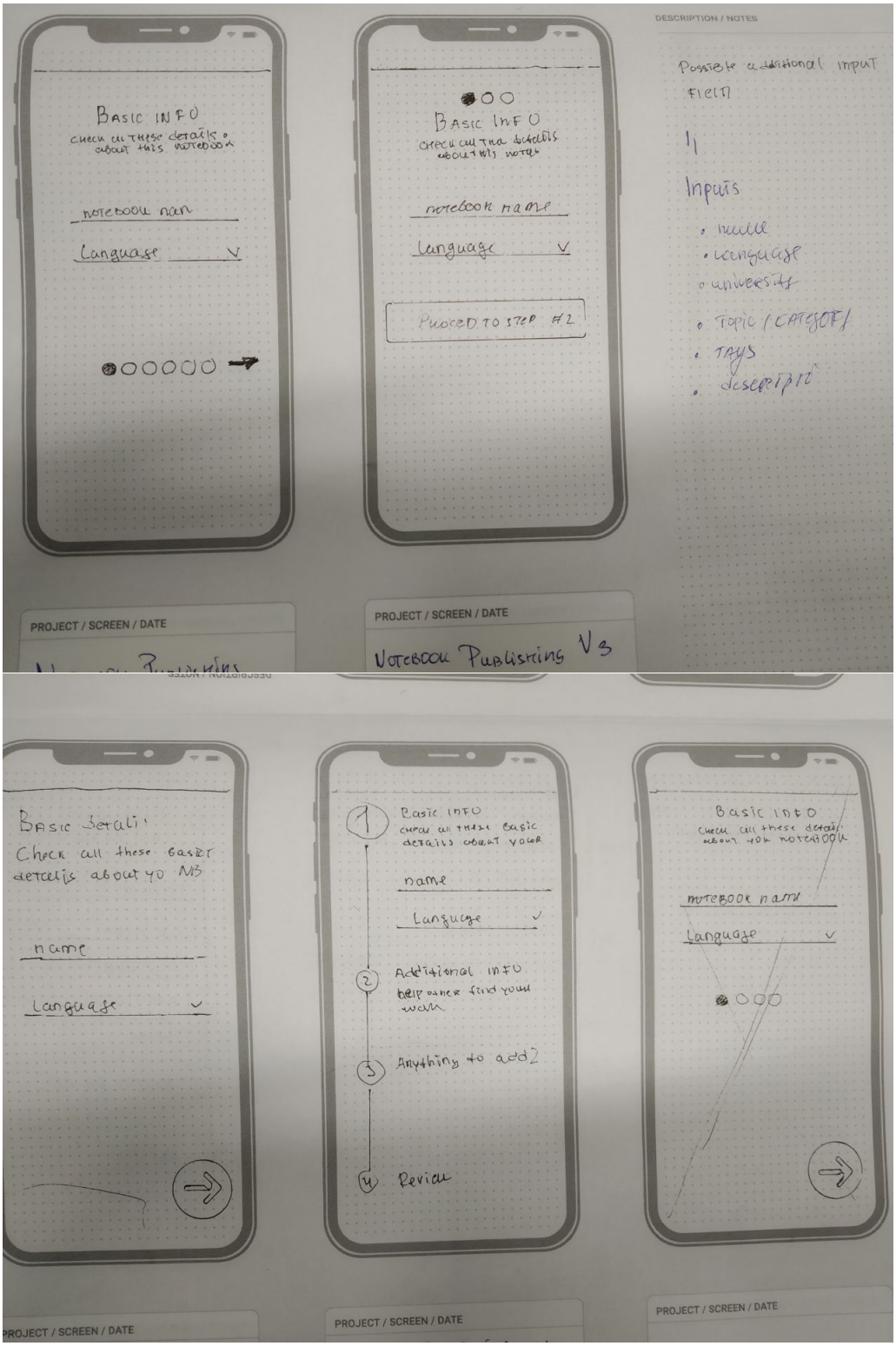
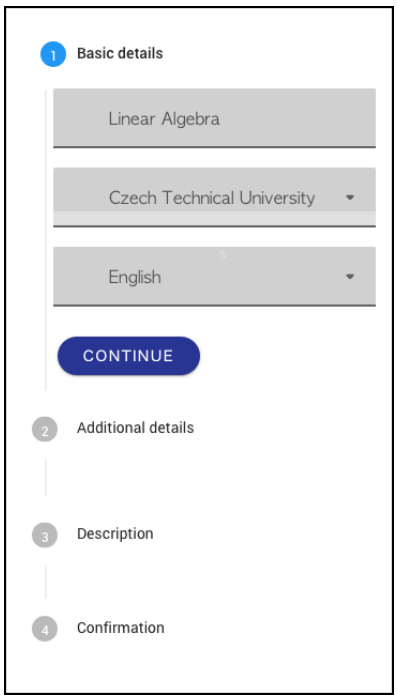
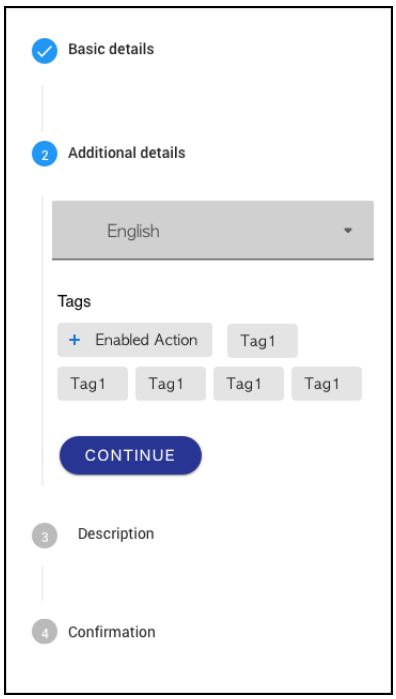


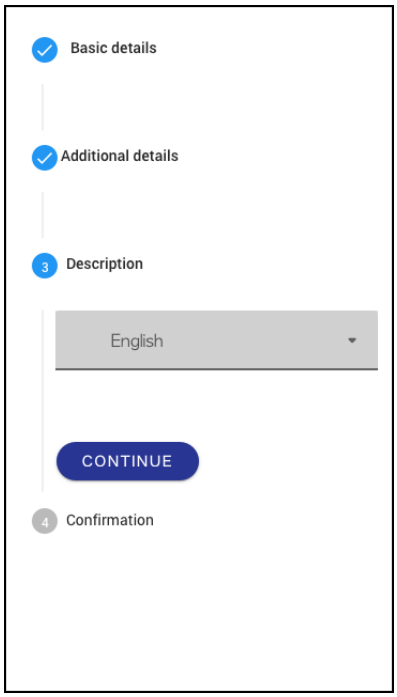
Figure 2.2: Notebook Publishing Flow - Sketch Phase



(a) First Step of Notebook Publishing



(b) Second Step of Notebook Publishing



(c) Third Step of Notebook Publishing

Figure 2.3: Notebook Publishing Wireframes

2.2.3 Sharing Hub

Sharing Hub is another top-level destination, that allows users to access various Notebooks published by other users.

In the early stage of the StudyPad development, Sharing Hub root screen (Figure 2.4) contained only one list with the Published Notebooks, with the idea of showing the most recently added Notebooks. Later on, the decision was made to make this screen more personalised by introducing Sections and Search Shortcuts (Figure 2.5)

Search Shortcut represent one of the previously entered queries during Search Flow. By clicking the Shortcuts, new Search Flow will be started with all the previously used Search Options, such as tags, pre-filled. Section is small list of Published Notebooks of fixed size, that can be scrolled horizontally. By introducing these small lists it will be possible to group Published Notebooks into personalised sets(Notebooks from the user's university for example). By its nature, Sections are simple queries, so it can be used as Search Shortcuts too. So, by clicking See All in Notebooks From my University section, user will be navigated to the Search Flow with the pre-filled University selection.

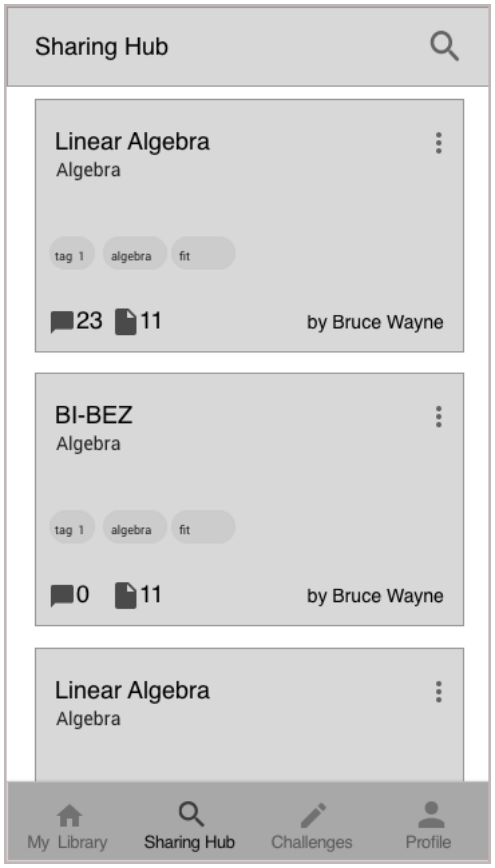


Figure 2.4: First Iteration of Sharing Hub Landing Page

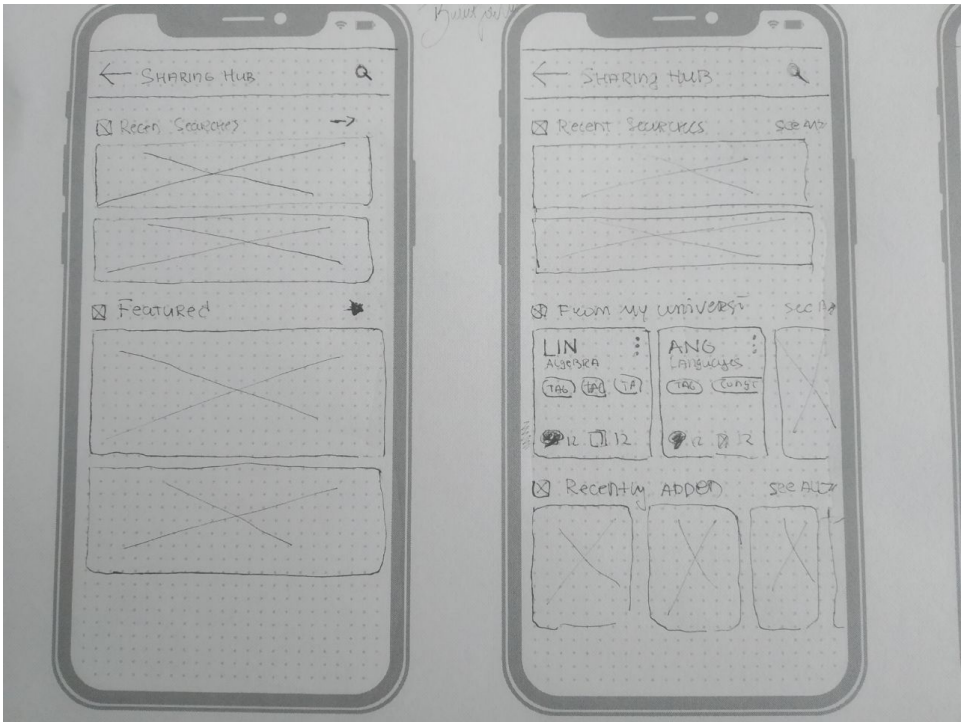


Figure 2.5: Second Iteration of Sharing Hub Landing Page

One of the main responsibilities of this screen is to start Discovery flow. Discovery flow includes various tools with goal to help find user new notebooks, on of them being Searching flow. Searching flow key requirements are following:

- There are should several primary searching options available: Notebooks name, University/School, Topic/Category. There are also several secondary options to narrow the search result: Tags and a language. Name of the notebook should be the only option that can be typed manually by user, all the other options, such as University or Topic, are well defined by the backed and can be chosen via Dialog for example.
- When none of searching options are active, empty state should be presented
- Previously used Searching options (if any) will be saved locally and presented as a shortcut when next discovery flow starts (i.e Empty state)

2.2.4 Published Notebook Details

Published Notebook, as an entity, contains a large amount properties and there are a lot of actions associated with it. This screen will cover all actions regarding user comments (requirements F18-F21) and various Published

Notebook actions such as saving, sharing, and copying. Including all these actions and flows into one screen would make UI very crowded, resulting in bad UX. *Tabs* can help solve this issue quite easy.

According to Material Design Guidelines, Tabs can organise content across different screens, data sets, and allow navigation between groups of content that are related and at the same level of hierarchy. [10]. By using Tabs, this screen can be split into two distinct sections: Detail Tab and Comments Tab, thus dividing responsibilities and making this screen more responsive.

Details Tab (Figure 2.6a) features key information about this Notebook. Due to the fact that there are a lot of details to display, all the properties are divided into several groups. First group contains simple properties such as Notebook's name, its author, category/topic, language and associated university. Second group contains more dynamic content such as description, that can take multiple lines, and a ChipGroup with tags. And the last group will serve a preview for the Notes.

Comments tab, as shown on Figure 2.6b, contains a scrollable list of the user comments and an Input Field, that will be used to create a new comment, or edit the old one. Also, every comment contains an options button that will trigger either an edit or a delete action.

2.2.5 Challenges Section

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

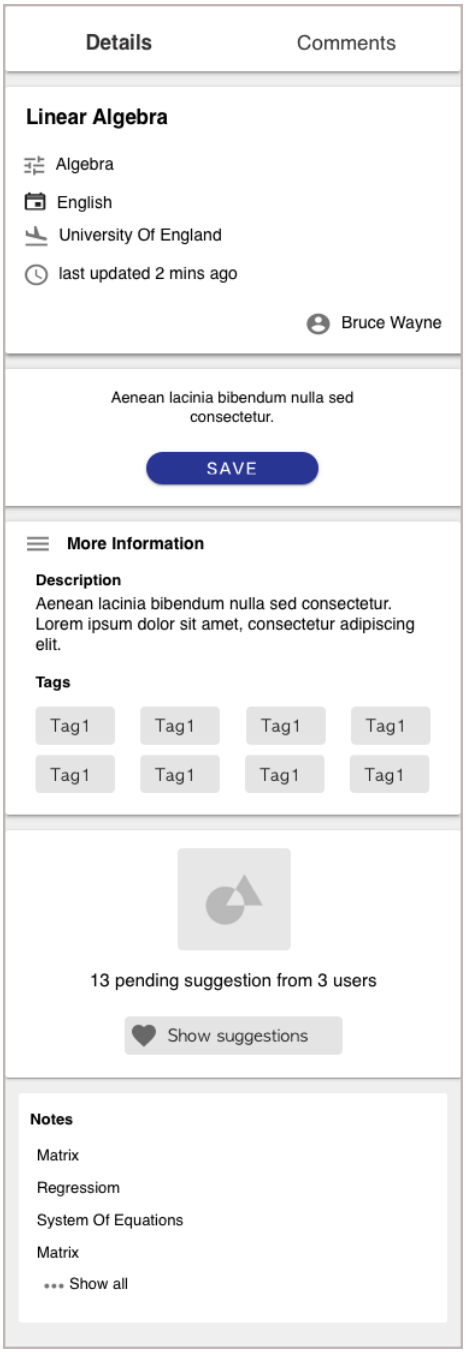
2.2.6 Profile Section

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

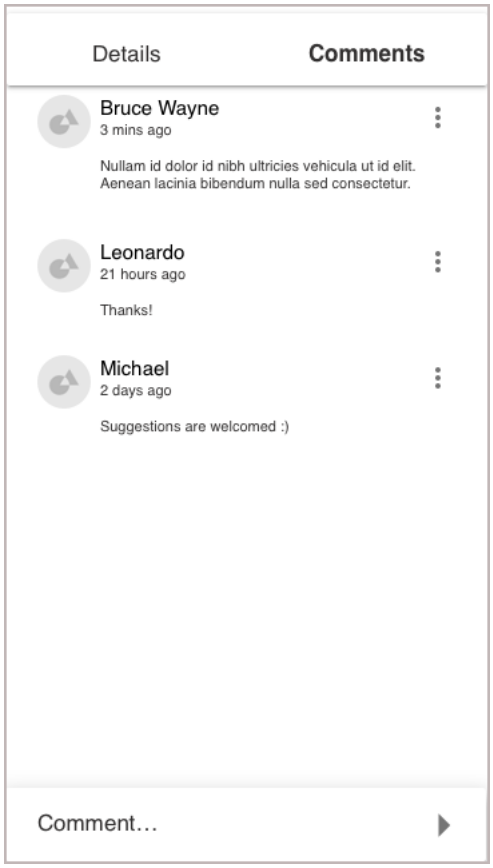
2.3 Application architecture

2.4 Platform-specific model

2.5 Main sequence diagrams



(a) Details Tab



(b) Comments Tab

Figure 2.6: Published Notebook Details Wireframes

Chapter 3

Implementation

3.1 Choice of technologies

3.2 Component diagram

3.3 Installation

[?]

Chapter 4

Testing

Conclusion

Bibliography

[1] Roman, L. *StudyPad Admin Panel*. Available from: https://github.com/levinzonr/studypad-web

[2] Roman, L. *StudyPad iOS Client*. Available from: https://github.com/levinzonr/studypad-ios

[3] Roman, L. *StudyPad Backend*. Available from: https://github.com/levinzonr/studypad-spring

[4] Wikipedia. Spring Framework — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Spring%20Framework&oldid=888321753, 2019, [Online; accessed 23-March-2019].

[5] Google. *AndroidX Overview [online]*. Accessed on 2019-07-07. Available from: https://developer.android.com/jetpack/androidx

[6] Roman, L. *Firebase*. Available from: https://firebase.google.com/products

[7] Floating Action Button. https://material.io/design/components/buttons-floating-action-button.html#, accessed: 2019-03-23.

[8] Bottom App Bar. ://material.io/design/components/app-bars-bottom.htmlbehavior, accessed: 2019-03-23.

[9] Understanding navigation. https://material.io/design/components/chips.html, accessed: 2019-03-23.

[10] Understanding navigation. https://material.io/design/components/tabs.html#usage, accessed: 2019-03-23.

Appendix A

Acronyms

GUI Graphical user interface

XML Extensible markup language

FAB Floating Action Button

UI User Interface

UX User Expirience

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

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