Insert here your thesis' task.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

StudyPad - Android Client

Roman Levinzon

Department of . . . (SPECIFY)
Supervisor: Ing. Miroslav Balík, Ph.D

December 16, 2018

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 December 16, 2018 .....

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, 2018.

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

Introduction

StudyPad is a combination of a note-taking service and a social network. Main purpose of the app is help users memorize different piecies of information and make it easy this information easy to exchange. To allow this, user will be able to create notes, that will act a single information piece. Notes are stored in distinct sets - Notebooks.

Chapter 1

Analysis

1.1 System description

1.2 Existing solutions

1.3 Domain model

1.4 Android Platform

1.5 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

1.5.1 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.

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.

Sharing Hub

- **F14: View published notebooks** User will be able to view notebooks published by other users.
- **F15: 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).
- **F16: Browse through published notebook** User will be able to see notes inside the notebook that's been published.
- **F17: View comments** User will be able to view others users comments discussing a notebook that's been published.
- **F18: Leave a comment** User can comment on other user published notebook.
- **F19: Delete a comment** Application will let user to delete his/her comment.
- **F20: Save published notebook** User will be able to save published notebook to his/her library.
- **F21: Publish notebook** User will be able to publish his notebook.
- **F22: Update published notebook** Author of the published notebook will be able to update it's information.
- **F23: Delete published notebook** Author of the published notebook will be able to delete the his/her notebook from shared space.
- **F24: Share notebook** User will be able to share his notebook by generating a deep-link.

Study Hub

- **F25: Start a basic self-check** User will be able to use an interactive way to look through his/her notes
- **F26: Start a written test** User will be able to participate in a written test based on one of notebooks to test his/her knowledge
- **F27: Start a quiz** User will be able to participate in quiz challenge that will be based on one of his/her notebooks

Settings

- **F28: View Profile Information** User will be able to view his/her profile information such as first name, last name and his university.
- **F29: Edit Profile Information** User will be able to edit his/her profile information.
- **F30: Logout** User will be able to logout from the application.

1.5.2 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 latest 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 Localization** Application will be able to adapt to different languages based on user locale

1.6 Existing solutions

There are several apps out there, whose goal is similar to StudyPad. However, most of the solutions are limited to learning languages and have limited sharing and discovering options. Table bellow shows requirements comparison such apps

Application Requirement	**Quizlet**	**Cram**	**TinyCards**
F1	Present	Present	Present
F2	Present	Present	Present
F3	Present	Present	Present
F4	Present	Present	Present
F5	Present	Present	Present
F6	Absent	Absent	Absent
F7	Present	Present	Present
F8	Present	Present	Present
F9	Present	Present	Present
F10	Present	Present	Present
F11	Present	Present	Present
F12	Present	Present	Present
F13	Present	Present	Present
F14	Absent	Absent	Present
F15	Limited	Limited	Limited
F16	Present	Present	Present
F17	Absent	Absent	Absent
F18	Absent	Absent	Absent
F19	Absent	Absent	Absent
F20	Present	Limited	Limited
F21	Limited	Limited	Limited
F22	Present	Present	Present
F23	Present	Present	Present
F24	Present	Limited	Limited
F25	Present	Present	Limited
F26	Present	Present	Limited
F27	Present	Present	Limited
F28	Present	Present	Limited
F29	Limted	Limted	Limited
F30	Present	Present	Limited

1.6.1 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 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 language learners. As a consequence, the only distinctions between Study sets are its name and a language. These are the only 2 options available for filtering published study sets.

1.6.2 Cram - Key differences

Cram is very similar to Quizlet but feels 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 splitted 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, it can only be automatically saved to Recent section, or manually added to Favourites. This makes it impossible to make local edits

1.6.3 TinyCards - Key Differences

TinyCards is app 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

1.7 Platform-Independent Model

Chapter 2

Design

2.1 Wireframes

2.2 Application architecture

2.3 Platform-specific model

2.4 Main sequence diagrams

Chapter 3

Implementation

3.1 Choice of technologies

3.2 Component diagram

3.3 Installation

Chapter 4

Testing

Conclusion

Appendix A

Acronyms

GUI Graphical user interface

XML Extensible markup language

Appendix B

Contents of enclosed CD

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