

LectureMate

Draft Project Report

Manpreet Bance

April 2020

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Scope	2
1.3	Objectives	2
2	Technical Specification	3
2.1	Background Research	3
2.1.1	Existing Systems	3
2.2	Overview	3
2.3	Application Frameworks	4
2.4	Full Stack Development	4
2.4.1	Flutter	4
3	Project Management	6
3.1	Labour/Resource Management	6
3.2	Methodology	7
3.2.1	Agile Development Method	7
3.2.2	Kanban	7
3.3	Version Control	7
3.3.1	Git	7
4	Analysis	8
4.1	System Requirements	8
4.1.1	Functional Requirements	8
4.1.2	Non-functional Requirements	8
4.2	Stakeholders	8
4.3	Use Case	8
4.3.1	Use Case Scenarios	8
4.3.2	Use Case Diagram	8
5	Design	9
5.1	Design Models	9
5.1.1	Activity Diagram	9
5.1.2	Sequence Diagram	9
5.2	UI Design	9
6	Implementation	10
6.1	Front End - Flutter	10
6.2	Business Layer - Flutter	10
6.3	Mobile Application Development	10
6.4	Code Review	10
6.5	Disruptions and Solutions	10

7	Testing	11
7.1	Testing Criteria	11
7.2	Testing Results	11
7.3	Formative Evalutation	11
7.4	Functional Requirements Evaluation	11
7.5	Non-functional Requirements Evaluation	11
8	Conclusion	12
8.1	Summative Evaluation	12
8.2	Potential Future Development	12
9	Appendices	13
10	References	14

DRAFT

List of Figures

DRAFT

List of Tables

DRAFT

Acknowledgements

I would like to express my gratitude towards my supervisor, Golnaz Badkobeh, for her continued support and guidance throughout the course of the entire project, providing assistance from preparing the project for development and to project completion.

I would also like to thank Lahcen Ouarbya, and the Department of Computing at Goldsmiths, University of London, for the opportunity to work on this project.

DRAFT

Chapter 1

Introduction

LectureMate is designed to be a cross-platform mobile application for use in an educational setting, namely, lectures - whereby, students and other academics can use the application to make dynamic, powerful and relevant notes which are linked to a specific timestamp or set of timestamps. The type of notes which can be made are rich text, uploaded images or camera photos, links, documents/research papers - all of which would the user would deem to be relevant to the point of the audio recording which is being made. As well as this, the user has the ability to highlight certain segments in the audio which they feel are important and they would benefit from later on when coming back to review these notes, for example, during revision.

As previously mentioned, the mobile app is mainly aimed at educators and academics in the educational setting but due to its capabilities could potentially be used in a range of other settings. Primary research showed that there were no existing applications on the market which offered the functionality that this mobile application aims to do. Existing applications such as Evernote are based primarily on text-based notes with the ability to add further attachments, whereas LectureMate would be audio-based allowing for notes to be made based on the timestamp in the audio recording.

Secondary research was able to prove that educators (the primary target market) and among students at Goldsmiths College, University of London, that there was no one consistent form of making clear, relevant notes and that the target market relied on a number of various applications to make notes which was not a practical solution. These applications included the likes of Evernote, Microsoft OneNote, Google Docs, as well as traditional notes which were made with pen and paper which were later difficult to merge with existing notes on these applications.

From this, it was clear that the application would aim to provide a satisfactory solution to the needs of the market and honoring what is in need by the target market which was able to be identified through the research. As a result, upon deployment of the application we would be able to see whether this is fulfilled and is versatile enough to meet the needs of its users. This is evidence to show how an application and platform such as LectureMate is vital in filling a much needed gap in the market.

1.1 Motivation

The motivation behind this mobile application is upon research the lack of platforms which allow users to make notes based on audio recordings. As well as this, in first-hand experience, I was able to identify that I found note-taking particularly inconsistent and

upon coming back to these notes to revise I found these notes to have no context and difficult to understand. Therefore, I carried out some research to see whether any platforms existed which allowed notes to be made but with some context and found that there were none for wide use.

By this and by communicating with peers among different educational settings, I was able to recognise that most universities and lecture theatres had recording equipment. I initially upon ideation, thought of pursuing this, however, later came to the conclusion that this would not be a viable option due to the issues with obtaining the permission of those individuals who may be visible in the recording, and as a result I made the final decision to proceed with the same idea however, using audio recording instead.

1.2 Scope

LectureMate aims to combat the issues that students and academics face in the note-taking and revision aspect of their learning by providing a platform where they can make clear, dynamic and consistent notes in order to tackle the existing problems faced with making notes where the notes are made across different platforms making them different to compile and are inconsistent. Therefore, making clear, consistent and dynamic notes using LectureMate are a solution to these issues.

1.3 Objectives

One of the first things which were looked at was how we can meet the objective to fulfill the need for the target market which was identified. LectureMate does this by providing a platform to satisfy the need for making clear notes.

We needed to make sure that the application met the needs that our target market fed back to us when conducting our primary and secondary research. Another objective that was indicated through executing this research was that the application would have the ability to highlight specific points within the audio where upon reviewing, would be able to identify the key, relevant sections of a lecture or recording.

The security and integrity of the notes which are made are a priority as these notes are to be kept safe whereby unauthorised changes should not be made to the notes as well as their security being ensured as to them not becoming deleted or corrupted. Therefore, considering this, the app aims to be a safe and reliable platform for its users so that they do not need to worry about the integrity of their notes. The application will do this by saving the note as a file in a folder creating regular backups each day or manually.

As the files will be stored locally on the device, the user is responsible for any data which might be subject to any tampering or corruption to the files when being manipulated through any third party application such as a file explorer. Using the application to make any changes to any notes will be a tested and safe way for users to alter notes.

Chapter 2

Technical Specification

2.1 Background Research

Here, I will focus and discuss the research I carried out prior to commencement of the design of the application. I will begin technical research looking at the potential for technologies I may use to develop the mobile application and briefly address the reasons behind this.

2.1.1 Existing Systems

In the research stage, I wanted to explore the market for existing applications which perform or complete similar or exactly the functions that I was aiming for in this application. I decided to look at features I would be able to implement from others or simply known, working models which

Soundcloud

While doing research, I thought of existing systems which I can obtain certain features from which I might be able to incorporate into the mobile application. Soundcloud came to mind whereby in their interface, they have the ability to have their users to add comments at certain time points in the audio which had been uploaded. I thought this same model would be able to be applied to this application as notes can be added at a specific point in the audio recording.

Evernote

Siri

2.2 Overview

The technical architecture of the mobile application is as simple as the layering mode. The logic of a layered architecture allows the separation of the Front-End, Middleware and Back-End, however still allowing for cohesion between these, which will be detailed further.

The uppermost layer will be the presentation layer - which will be the appearance and interaction side of the application, the middle layer - which will be responsible for the (business) logic behind the application containing the code behind the actions interacted with in the presentation layer, and the data access layer - which will be the back end code which enables the user to access and make changes to the notes, although stored locally, they will still carry out CRUD functionality similar to that used in databases.

2.3 Application Frameworks

A framework is an abstraction where software is written to provide generic function to achieve a task or tasks. This mobile application has made use of some frameworks to contribute to the functioning of it - mainly being in the Front-End or 'Presentation Layer'.

2.4 Full Stack Development

As previously mentioned, the web application was designed and structured in a way conforming to the stack structure: Front-End, Middleware, and Back-End. This section will further discuss the specification of technologies in each tier of the stack and the reasons behind their uses:

2.4.1 Flutter

The entirety of the mobile application will be using the existing Flutter framework which is an open-source UI software development kit created by Google - used to develop applications across mobile, web and desktop applications. I will detail the framework architecture and the way each of these are used in the development of the mobile application:

Dart platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features. The Dart language package allows during development for implementing features into the application which help shape the appearance and layout as it appears to be.

During the development stage, the Dart package also allows for dynamic and a live coding experience which involves hot reloading whereby any changes in code can be updated and made live without having the whole application have to be restarted for the new changes to take effect which was a welcome and proven helpful feature to the process.

Flutter engine

The flutter engine provides low-level rendering support interfacing with iOS and Android platforms which hosts the Flutter applications and manages the way they look and behave controlling and making use of its core libraries which include animation and graphics, file and network I/O, accessibility, plugin and runtime and compilation toolchain. However, these were things which during the development process were something which were not deeply explored and only utilised for basic functionality.

Using the Flutter framework allowed for a more reactive, modern framework for creating a modern and engaging mobile application which would also function well.

Foundation library

The foundation library written in Dart allowed for the use and implementation of basic classes and functions in the mobile application enabling access to APIs to communicate with the engine and construct the application.

Widgets

Flutter widgets are built into the package and work as the different components which make up together to build the application, such as buttons and text. These are included as standard and are used in the application for some of the components.

Design-specific widgets such as Material design using Google's design language as well as Cupertino which use Apple's Human Interface Guidelines iOS design, both make use of these respective design languages to display the application best in their respective, native environments to give them a more uniform look despite being on different platforms.

DRAFT

Chapter 3

Project Management

3.1 Labour/Resource Management

Prior to commencement of the project, I planned to allocate resources in the form of time, being the number of hours spent, on each stage of the project with ideation to conception, research, development and testing stages to ensure that this was all done in good timing and enough resources were spent on each stage so that it would be produced to a high standard. Following the allocation of resources available throughout the project, I created a timeline with set provisional dates whereby these tasks should be completed, referred to as milestones. The following table breaks down the allocation of hours spent on each part of the project:

*****TABLE*****

Taking this approach to the allocation of labour and resources means I can plan when certain tasks should be done and if these are not met, I can allocate further resources to make the project less susceptible to interruptions, which may include:

- 1. Overlap of tasks:**

The tasks were divided in smaller segments to ensure better productivity and efficiency in completing each task and to make sure that each task was completed correctly prior to moving onto the next task. This guaranteed that there was less of a feeling of a being stuck on a particular task for a prolonged amount of time and a wider variety to provide a faster workrate of tasks being completed. Due to the nature of the development process being in Flutter, the bulk of the project was the middleware and backend controlling the way functions and features were supposed to behave and whether these were being suitably met. The tasks were divided in such a way that there was no dependency of completion of a task to complete the next as such until it came to testing to see whether these worked together cohesively.

- 2. Time keeping and meeting deadlines:**

The time keeping and milestone aspect of the planning process was very important so that each of these were being met on time and no stage of the development process was falling behind so much so that it would not be complete and the application as a whole would suffer. To combat this, I created a blog so that I was keeping track of the developments made in the project each week to track whether this progress was good and steady and would meet the milestones at the correct times. Planning these milestones enables for a greater control of the project and the way it is progressing.

3.2 Methodology

3.2.1 Agile Development Method

The Agile Project Management Methodology was utilised where, by definition, an iterative process made use of at each stage of development in all areas of the development stack requiring constant effective communication. Using the Agile method meant that the project was dealt with the most efficient way possible with larger tasks being broken down and divided as to the larger and smaller tasks taking varying time to complete and complete these tasks effectively.

3.2.2 Kanban

The Kanban method is also used in order to keep track of the items which need to be completed and which are still yet to be completed as well as a section to review code and carry out ah-hoc testing as well as component testing to see whether each component is working as expected. This also provides a more broader look at the tasks to come and time to prepare for these, seeing whether any previous tasks are dependant or feed into the following task. The Kanban method is visualised using Trello.

3.3 Version Control

3.3.1 Git

The project makes strong use of Git and its features for version control purposes, with the ability to see and track any changes to files and have a history of the previous changes made and to apply any future fixes giving more of a sense of control and responsibility for each task to be completed correctly as well as the a timestamp of when this has been completed to identify whether it has been done in good time according to the milestones and personal proposed dates.

Chapter 4

Analysis

4.1 System Requirements

4.1.1 Functional Requirements

4.1.2 Non-functional Requirements

4.2 Stakeholders

4.3 Use Case

4.3.1 Use Case Scenarios

4.3.2 Use Case Diagram

Chapter 5

Design

5.1 Design Models

5.1.1 Activity Diagram

5.1.2 Sequence Diagram

5.2 UI Design

DRAFT

Chapter 6

Implementation

6.1 Front End - Flutter

6.2 Business Layer - Flutter

6.3 Mobile Application Development

6.4 Code Review

6.5 Disruptions and Solutions

Chapter 7

Testing

7.1 Testing Criteria

7.2 Testing Results

7.3 Formative Evaluation

7.4 Functional Requirements Evaluation

7.5 Non-functional Requirements Evaluation

Chapter 8

Conclusion

8.1 Summative Evaluation

8.2 Potential Future Development

DRAFT

Chapter 9

Appendices

DRAFT

Chapter 10

References

DRAFT

END OF DOCUMENT