Enterprise Mobility

SE5070



*Assignment I*

**Name**: B. R. M. S. R. B. Rathnayake

**Student ID**:  MS21911958

**Application Name**: *SLIITLIFY Masters*

**Technologies**: *React Native/Firebase*

Contents

[Selection of Project Area 3](#_Toc84642476)

[Overview 3](#_Toc84642477)

[Project idea 3](#_Toc84642478)

[Project features 4](#_Toc84642479)

[Application architecture 5](#_Toc84642480)

[Major Decisions 5](#_Toc84642481)

[Intent Routing 5](#_Toc84642482)

[Source code Organization 6](#_Toc84642483)

[Data Flow 6](#_Toc84642484)

[Offline access technology 7](#_Toc84642485)

[Design constraints and principles followed 8](#_Toc84642486)

[Integration hardware 9](#_Toc84642487)

[UI/UX flow 10](#_Toc84642488)

[Wireframe 1 – Main User Interface 10](#_Toc84642489)

[Wireframe 2 – Navigation Pane 10](#_Toc84642490)

[Wireframe 3 – Login Interface 11](#_Toc84642491)

[Wireframe 4 – QR Scanner Interface 11](#_Toc84642492)

[Wireframe 5 – 360 Guest View 12](#_Toc84642493)

[Usage of custom components 13](#_Toc84642494)

## Selection of Project Area

**MS21911958** = 21911958

= 2 + 1 + 9 + 1 + 1 + 9 + 5 + 8

= 36 = 3 + 6

= 9 modulo 3

= 0 -> EduTech.

## Overview

“SLIITLIFY Masters” is an react-native application built for students who are enrolled to SLIIT master’s degree.

A picture containing text

Description automatically generated

### Project idea

*User - Student*

May it be MSc, MBA, MPhil, in whichever stream this app serves the generic purposes for students and lecturers, to ease following up with attendance for lecturers, whilst making it easier for students to attend lectures and mark attendance through the app itself via a QR (quick response) code scanner. The student has direct access to Moodle (courseweb), student profile, SLIIT portal.

Students can login to the system using their MS Number as the username, and NIC as password (the same login is used to view results/grades in official SLIIT result board). This login will be validated by the backend servers. Authenticated students will login, invalid credentials will be rejected with a toast message. Student can slide the dashboard to the right, and view/select the menu options.

Main objective behind the project is to capture student’s attendance and to allow the student to view their attendance progress. ***How?*** The lecturers project a QR code during the lecture session, embedded in the middle of the lecture slide. (To make things interesting, students must pay attention to the lectures because the QR can be projected on any random slide.) For any reason if the student was unable to use the app/smartphone, the student needs to inform the lecturer regarding the technical difficulty during the lecture itself. The scanned QR code will directly update the backend, which maintains the attended lectures, total lectures (3 lectures attended out 10 total lectures) along with the enrolled module(s) details.

*User – Guest*

This mobile app is not limited for just SLIIT masters students. If any outsider from the university or any student who doesn’t follow a master’s degree wants to use the app, they can simply use the Guest login. Guest login is authenticated with Google authentication with Firebase Auth. The guests can simply login to the app and browse through limited functionality shown on the left-hand side navigator pane. The reason for this login is to ensure that the app has a broad audience, and it could help outsiders to learn more about SLIIT Culture, learn about the offered master’s degrees and explore SLIIT vicinities.

\*Assumptions: SLIIT administrates need to feed sensitive data (student profile pic, MS details, etc.) into the firebase backend, this app works assuming that data setup is already done.

### Project features

Student authentication against Firebase backend.  
Dedicated backend REST API, using cloud functions to perform CRUD operations, can be accessed over any device with an internet connection.  
Easy user Navigation.  
User preferences (dark theme/light theme)  
Native hardware used: QR code scanner, camera, vibrate, speakers. Guest users can view faculty vicinities and classroom interiors in 360-degree mode with the aid of Gyroscope. a graphical view for students to view current attendance status along with modules enrolled with a given semester/year. depending on the user login (guest /user) the left-hand side navigator pane will show/ hide certain features and functionalities available for the respective user.

## Application architecture

### Major Decisions

Maintaining a separate backend API to have server-side processing. SLIITLIFY Masters application currently several business logics and functionality, data validations written on the app itself. But in the long-term picture, these validations and processing can be gradually moved to the server side, in hopes of maintaining a dummy client as for the front end.

SLIITLIFY Masters follows three tier layered architecture.

A screenshot of a computer

Description automatically generated with medium confidence

### Intent Routing

For routing purposes, static routing has been used inside DrawerContent.tsx

Graphical user interface, text

Description automatically generated

### Source code Organization

Source code folder structure has been structured based on **Type.**

A picture containing chart

Description automatically generated

### Data Flow

User Sequence Diagram

Diagram

Description automatically generated with medium confidence

Attendance Sequence Diagram

Diagram

Description automatically generated

### Offline access technology

[react-native-offline@6.0.0](mailto:react-native-offline@6.0.0) was used to handle users that are offline

Text

Description automatically generated

### Design constraints and principles followed

Currently, the project is configured and tested to run on Android platforms inside the **Config.ts**

A screenshot of a computer

Description automatically generated with medium confidence

From a development perspective, the front-end application was built using Typescript, JavaScript and React which is then compiled to Native code to support android/iOS platforms. For iOS purposes, change the **Config.ts** file from android to iOS. Such instance is shown below to adapt to Apple smartphone display sizes.

const marginTop = Platform.OS === 'ios’? 20 : StatusBar.currentHeight;

Efforts for a thin mobile client approach can be improved in further steps, the backend server can handle most of the business logic. As an improvement, we can allocate load balancers to enhance the response times.

Data synchronization structure used - On-demand and cached for later used to prevent excessive loading.

User Agility – One Handed usage for the app, the navigator drawer can be accessed from a default thumb zone. Simple thumb gestures with easy memorization and ease of use

Buttons are fed with motion feedback, animation transition added on user picture to denote layout changes and sound emitted on screen menu tap. Simple clear icons to enhance visibility, with globally followed conventions.

Anti-pattern Hamburger menu was used for quick user navigation since there are several menus

Communication protocol used - HTTP REST with AXIOS Library

Conditional return statements used to show user a Loading screen until all data is fetched from the database.

**Other Constraints** – The application is ideal for a portrait mobile device usage. Minimum of Android version 4.4.4 above is required, 2 GB RAM (considering other applications running in the background) and bare-minimum of 100MB storage space to install the application.

## Integration hardware

QR code scanner, sound, vibration torch

## UI/UX flow

### Wireframe 1 – Main User Interface

A picture containing graphical user interface

Description automatically generated

### Wireframe 2 – Navigation Pane

Graphical user interface

Description automatically generated

### Diagram Description automatically generatedWireframe 3 – Login Interface

### Wireframe 4 – QR Scanner Interface

Graphical user interface, application

Description automatically generated

### Wireframe 5 – 360 Guest View

Graphical user interface

Description automatically generated

Hamburger menu is considered as an Anti-pattern, nevertheless for this application purpose a hamburger menu navigation seemed fit. Students might find it frustrating to swipe left and right continuously to navigate to a certain page. However, for the 360 Classroom view for the guest, a vertical scrollable view was proposed to enhance user experience.

All interfaces have eye soothing colors that were coordinated with international color pallets, since the major target audience for this application is SLIIT students, some of SLIIT color themes were integrated as well. Cool animations, icons, sound effects, vibrations that follow convention were integrated so the UX is not frustrating. User can customize the application as they wish such as the light/dark theme, this would make the student feel like they are engaging with the application in a customized preference.

* + <https://oblador.github.io/react-native-vector-icons/>

materialIcons

## Usage of custom components

* + Tag completely developed by the student himself