

# **MOBILE APP FOR FYP SUPERVISORS**

**TENGKU FAHAD HAFIZ**

**SESSION 2022/2023**

**FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERSITY  
JULY 2023**

# **MOBILE APP FOR FYP SUPERVISORS**

**BY**

**TENGKU FAHAD HAFIZ**

**SESSION 2022/2023**

**THE PROJECT REPORT IS PREPARED FOR  
FACULTY OF COMPUTING AND INFORMATICS  
MULTIMEDIA UNIVERSITY  
IN PARTIAL FULFILLMENT  
FOR**

**BACHELOR OF COMPUTER SCIENCE  
B.CS (HONS) SOFTWARE ENGINEERING**

**FACULTY OF COMPUTING AND INFORMATICS**

**MULTIMEDIA UNIVERSITY**

**JULY 2023**

Copyright of this report belongs to Universiti Telekom Sdn. Bhd. as qualified by Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialisation Policy. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Universiti Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2023 Universiti Telekom Sdn. Bhd. ALL RIGHTS RESERVED.

## **DECLARATION**

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.



---

Name: Tengku Fahad Hafiz Bin T. Abd Nawawi

Student Id: 1151102898

Faculty of Computing and Informatics

Multimedia University

Date: 13 July 2023

## **ACKNOWLEDGEMENT**

I am deeply thankful to my supervisor Dr. Yeoh Eng Thiam for providing me with unwavering guidance throughout my Final Year Project. Your insightful advice and recommendations have been of immense help and I am eternally grateful for that. I would also like to extend my gratitude to my family and friends for their constant support and encouragement during the project. Their belief in me kept me motivated during the tough times. I would like to give a special shoutout to my friend Tasnim for offering his valuable experience and knowledge in software engineering throughout the project. Your input was truly invaluable. Lastly, I cannot emphasize enough the role that my peers played in making this project a success. Their cooperation, support, and contributions were essential to its success.

## **ABSTRACT**

The project "Mobile app for FYP (Final Year Project) supervisors, faculty managers, moderators and coordinators" aims to improve the usability of the FYP System by studying the requirements of the app and designing a prototype based on those requirements. A background study found that most FYP systems are web-based and rely heavily on conventional communication methods. A survey concluded that the current system is moderately reliable, but the majority of the MMU staff would like a project management module added.

The proposed solution uses React Native as the framework and Android Studio and XCode for setup. It will be hosted with WebSocket and the backend and mobile coding will utilize JavaScript through the Node.js environment. The system has five (5) actors: Supervisor, Faculty Manager, Moderator, Coordinator, and Administrator. Each actor has different capabilities such as logging in, viewing announcements, creating projects, creating meetings, and managing presentations.

The software architecture includes React Native Mobile Application, AWS S3 Bucket for storing content, Node.js EC2 Backend for processing requests, sending responses, and various other tasks, and PostgreSQL for database queries.

This mobile app is designed to improve the efficiency and usability of the FYP system by providing a comprehensive solution for supervisors, faculty managers, moderators, coordinators, and administrators. The proposed solution is based on modern technology and incorporates various AWS services to ensure a reliable and scalable system.

# Table of Contents

|   |           |
|---|-----------|
| <b>DECLARATION</b>  | <b>2</b>  |
| <b>ACKNOWLEDGEMENT</b>                                    | <b>3</b>  |
| <b>ABSTRACT</b>   | <b>4</b>  |
| <b>List of Tables</b>                                     | <b>8</b>  |
| <b>List of Figures</b>                                    | <b>9</b>  |
| <b>Chapter 1: INTRODUCTION</b>                            | <b>1</b>  |
| 1.1 Project Overview:                                     | 1         |
| 1.2 Project Objectives:                                   | 1         |
| 1.3 Goals of project:                                     | 2         |
| 1.4 Project Plan  | 3         |
| 1.4.1 Milestones  | 3         |
| 1.4.2 Gantt Chart   | 5         |
| <b>Chapter 2: BACKGROUND STUDY</b>                        | <b>13</b> |
| 2.1 Existing Systems                                      | 13        |
| 2.1.1 Multimedia University (MMU)                         | 14        |
| 2.1.2 Universiti Teknologi MARA (UiTM)                    | 16        |
| 2.1.3 University of Malaya (UM)                           | 17        |
| 2.1.4 Universiti Teknologi PETRONAS (UTP)                 | 18        |
| 2.1.5 Universiti Malaysia Sarawak (UNIMAS)                | 18        |
| 2.1.6 BINUS Mobile for Lecturer                           | 19        |
| 2.1.7 Sister For Lecturer UNEJ                            | 20        |
| 2.2 Literature Review                                     | 21        |
| 2.2.1 Survey  | 24        |
| 2.2.2 Survey Analysis                                     | 27        |
| 2.2.3 Survey Conclusion                                   | 32        |
| 2.3 Problems to be solved                                 | 33        |
| 2.4 Technological Background                              | 34        |
| 2.4.1 Native Mobile App Development Tools                 | 35        |
| 2.4.1.1 Xcode   | 35        |
| 2.4.1.2 Android Studio                                    | 36        |
| 2.4.2 Cross-Platform Mobile App Development Tools         | 36        |
| 2.4.2.1 React Native                                      | 37        |
| 2.4.3 Backend   | 37        |
| 2.4.3.1 Node.js   | 37        |
| 2.4.3.2 Ruby on Rails                                     | 38        |
| 2.4.3.2 Key Differences between Node.js and Ruby on Rails | 39        |
| 2.4.4 Host  | 40        |
| 2.4.4.1 Fly.io  | 41        |
| 2.4.4.2 Backblaze B2                                      | 41        |

|  |           |
|--|-----------|
| 2.4.4.2 WebSocket                        | 42        |
| 2.4.4.2.1 AWS vs. Azure vs. Google Cloud | 43        |
| 2.5 Proposed Solution                    | 44        |
| <b>Chapter 3: REQUIREMENTS</b>           | <b>46</b> |
| 3.1 System Overview                      | 46        |
| 3.2 System minimum requirements          | 49        |
| 3.2.1 Android                            | 49        |
| 3.2.2 iOS                                | 49        |
| 3.3 Use Case Diagram                     | 50        |
| 3.3.1 Supervisor Modules                 | 51        |
| 3.3.2 Faculty Manager Modules            | 58        |
| 3.3.3 Moderator Modules                  | 60        |
| 3.3.4 Coordinator Modules                | 62        |
| 3.3.4 Administrator Modules              | 64        |
| 3.4 Class Diagram                        | 67        |
| 3.4.1 Supervisor                         | 68        |
| 3.4.2 Faculty Manager                    | 69        |
| 3.4.3 Coordinator                        | 70        |
| 3.4.4 Moderator                          | 70        |
| 3.4.5 Admin                              | 71        |
| 3.5 ER Diagram                           | 72        |
| <b>Chapter 4: DESIGN</b>                 | <b>75</b> |
| 4.1 Software Architecture                | 75        |
| 4.1.1 Supervisor                         | 76        |
| 4.1.2 Faculty Manager                    | 77        |
| 4.1.3 Coordinator                        | 78        |
| 4.1.4 Moderator                          | 79        |
| 4.1.5 Admin                              | 79        |
| 4.2 Software Model                       | 81        |
| 4.3 Sequence Diagrams                    | 82        |
| 4.3.1 Login                              | 82        |
| 4.3.2 View Announcement                  | 83        |
| 4.3.3 Create Announcement                | 84        |
| 4.3.4 View Presentation Schedule         | 85        |
| 4.3.5 Add New Project                    | 87        |
| 4.3.6 Create Meeting                     | 88        |
| 4.4 Screen Design                        | 90        |
| 4.5 Data Dictionary                      | 91        |
| <b>Chapter 5: IMPLEMENTATION</b>         | <b>97</b> |
| 5.1 Deployment                           | 97        |
| 5.1.1 System Configuration               | 100       |
| 5.1.2 Database                           | 104       |

|  |            |
|--|------------|
| 5.2 Development Environment                    | 106        |
| 5.2.1 Development IDE                          | 109        |
| 5.2.2 Development Tools                        | 110        |
| 5.3 Software Modules                           | 112        |
| 5.3.1 Authentication Algorithm                 | 112        |
| 5.3.2 Announcement Creation Algorithm          | 112        |
| 5.3.3 Project Proposal Creation Algorithm      | 117        |
| 5.3.4 Project Approval and Rejection Algorithm | 120        |
| 5.3.5 Upload and Download Report Algorithm     | 120        |
| 5.3.6 Meeting Creation Algorithm               | 124        |
| 5.3.7 Meeting Update Algorithm                 | 126        |
| 5.3.8 Presentation Creation Algorithm          | 129        |
| 5.4 Sample Outputs                             | 132        |
| 5.4.1 Login Screen                             | 132        |
| 5.4.2 Main Screen                              | 133        |
| <b>Chapter 6: TESTING</b>                      | <b>139</b> |
| 6.1 Test Plan                                  | 139        |
| 6.1.1 Test Data                                | 142        |
| 6.2 Test Results                               | 145        |
| 6.2.1 Test Case - ALUC                         | 145        |
| 6.2.2 Test Case - SUUC                         | 147        |
| 6.2.3 Test Case - FCUC                         | 153        |
| 6.2.4 Test Case - FMUC                         | 156        |
| 6.2.5 Test Case - COUC                         | 157        |
| 6.2.6 Test Case - MOUC                         | 159        |
| 6.2.7 Test Case - ADUC                         | 160        |
| <b>Chapter 7: CONCLUSION</b>                   | <b>162</b> |
| <b>REFERENCES</b>                              | <b>163</b> |
| <b>APPENDIX A: COMMERCIALISATION PROPOSAL</b>  | <b>166</b> |
| <b>APPENDIX B: MEETING LOGS (FYP1)</b>         | <b>176</b> |
| <b>APPENDIX C: MEETING LOGS (FYP2)</b>         | <b>224</b> |
| <b>APPENDIX D: PLAGIARISM CHECKING REPORT</b>  | <b>266</b> |
| <b>APPENDIX E: EXTRA CONTENTS</b>              | <b>268</b> |

## List of Tables

|  |    |
|--|----|
| Table 1.1 Milestones for Trimester 1   | 3  |
| Table 1.2 Milestones for Trimester 2   | 4  |
| Table 2.1 Modules of Each System   | 21 |
| Table 2.2 Mobile Operating System Market Share Worldwide                         | 34 |
| Table 2.3 Node.js vs. Ruby on Rails  | 39 |
| Table 3.1 Supervisor - Login/Logout Module                                       | 52 |
| Table 3.2 Supervisor - View Announcement Module                                  | 52 |
| Table 3.3 Supervisor - Add New Projects Module                                   | 53 |
| Table 3.4 Supervisor - View Projects (own) Module                                | 55 |
| Table 3.5 Supervisor - Assign Students to Project Module                         | 54 |
| Table 3.6 Supervisor - Upload Project Report Module                              | 54 |
| Table 3.7 Supervisor - Create Meeting Module                                     | 55 |
| Table 3.8 Supervisor - View Meetings Module                                      | 55 |
| Table 3.9 Supervisor - Update Meetings Module                                    | 55 |
| Table 3.10 Supervisor - View Presentation Schedule Module                        | 56 |
| Table 3.11 Supervisor - View Profile Info Module                                 | 56 |
| Table 3.12 Supervisor - View List of Students/Supervisors/Moderators Module      | 57 |
| Table 3.13 Faculty Manager - Login/Logout Module                                 | A  |
| Table 3.14 Faculty Manager - View Announcement Module                            | A  |
| Table 3.15 Faculty Manager - Create Announcements Module                         | 58 |
| Table 3.16 Faculty Manager - Update Announcements Module                         | 59 |
| Table 3.17 Faculty Manager - View Projects Module                                | A  |
| Table 3.18 Faculty Manager - View Presentation Schedule Module                   | A  |
| Table 3.19 Faculty Manager - Setup Presentation Parameters Module                | 59 |
| Table 3.20 Faculty Manager - View Profile Info Module                            | A  |
| Table 3.21 Faculty Manager - View List of Students/Supervisors/Moderators Module | A  |
| Table 3.22 Moderator - Login/Logout Module                                       | A  |
| Table 3.23 Moderator - View Announcement Module                                  | A  |
| Table 3.24 Moderator - View Assigned Students' Project Module                    | 61 |
| Table 3.25 Moderator - Download Project Report Module                            | 61 |
| Table 3.26 Moderator - View Presentation Schedule Module                         | 67 |
| Table 3.27 Moderator - View Profile Info Module                                  | A  |
| Table 3.28 Moderator - View List of Students/Supervisors/Moderators Module       | A  |
| Table 3.29 Coordinator - Login/Logout Module                                     | A  |
| Table 3.30 Coordinator - View Announcement Module                                | A  |
| Table 3.31 Coordinator - Create Announcements Module                             | A  |
| Table 3.32 Coordinator - Update Announcements Module                             | A  |
| Table 3.33 Coordinator - View Projects Module                                    | A  |
| Table 3.34 Coordinator - Approve/Reject Project Module                           | 62 |
| Table 3.35 Coordinator - Assign Moderator to Project Module                      | 63 |
| Table 3.36 Coordinator - View Presentation Schedule Module                       | 63 |
| Table 3.37 Coordinator - View Profile Info Module                                | A  |
| Table 3.38 Coordinator - View List of Students/Supervisors/Moderators Module     | A  |
| Table 3.39 Admin - Login Module  | 64 |
| Table 3.40 Admin - Create Accounts Module  | 65 |
| Table 3.41 Admin - View Accounts Module  | 65 |

|   |       |
|---|-------|
| Table 3.42 Admin - Edit Accounts Module   | 65    |
| Table 3.43 Admin - Delete Accounts Module | 66    |
| Table 4.1 Data Dictionary                 | 91    |
| Table 6.1 Test Plan                       | 139   |
| Table 6.2 Test Data                       | 142/A |
| Table 6.3 - 6.27 Test Case                | 145/A |

## List of Figures

|  |    |
|--|----|
| Figure 1.1 Gantt Chart part 1 for Trimester 1                      | 5  |
| Figure 1.2 Gantt Chart part 2 for Trimester 1                      | 7  |
| Figure 1.3 Full Gantt Chart for Trimester 1                        | 8  |
| Figure 1.4 Gantt Chart part 1 for Trimester 2                      | 9  |
| Figure 1.5 Gantt Chart part 2 for Trimester 2                      | 11 |
| Figure 1.6 Full Gantt Chart for Trimester 2                        | 12 |
| Figure 2.1 MMU FCI FYP Online System Front Page                    | 16 |
| Figure 2.2 UiTM FEE FYP Online System Front Page                   | 17 |
| Figure 2.3 UNIMAS FYP Online System Front Page                     | 19 |
| Figure 2.4 Sister for Lecturer UNEJ Home Screen (from Google Play) | 21 |
| Figure 2.5 Result for question 1 of survey                         | 27 |
| Figure 2.6 Result for question 2 of survey                         | 28 |
| Figure 2.7 Result for question 3 of survey                         | 28 |
| Figure 2.8 Result for question 4 of survey                         | 29 |
| Figure 2.9 Result for question 5 of survey                         | 29 |
| Figure 2.10 Result for question 6 of survey                        | 30 |
| Figure 2.11 Result for question 7 of survey                        | 30 |
| Figure 2.12 Result for question 8 of survey                        | 31 |
| Figure 2.13 Result for question 9 of survey                        | 32 |
| Figure 3.1 System Overview   | 46 |
| Figure 3.2 Use Case Diagram describing the FYP Mobile Application  | 50 |
| Figure 3.3 Use Case Diagram for the actor Supervisor               | 51 |
| Figure 3.4 Use Case Diagram for the actor Faculty Manager          | 58 |
| Figure 3.5 Use Case Diagram for the actor Moderator                | 60 |
| Figure 3.6 Use Case Diagram for the actor Coordinator              | 62 |
| Figure 3.7 Use Case Diagram for the actor Administrator            | 64 |
| Figure 3.8 Class Diagram   | 67 |
| Figure 3.9 Entity Relationship Diagram                             | 72 |
| Figure 4.1 Software Architecture                                   | 75 |
| Figure 4.2 Software Model  | 81 |
| Figure 4.3 Login Sequence Diagram                                  | 82 |
| Figure 4.4 View Announcement Sequence Diagram                      | 83 |
| Figure 4.5 Create Announcement Sequence Diagram                    | 84 |
| Figure 4.6 View Presentation Schedule Sequence Diagram             | 85 |
| Figure 4.7 Add New Project Sequence Diagram                        | 87 |
| Figure 4.8 Create Meeting Sequence Diagram                         | 88 |
| Figure 4.9 Login Screen  | 90 |
| Figure 4.10 Homepage/Screen  | 90 |

|  |     |
|--|-----|
| Figure 5.1 React Native in Machine                                       | 97  |
| Figure 5.2 Git in Machine  | 97  |
| Figure 5.3 Node in Machine   | 98  |
| Figure 5.4 EC2 Management Console  | 99  |
| Figure 5.5 S3 Management Console   | 100 |
| Figure 5.6 Executing React Native Project Creation in Windows PowerShell | 101 |
| Figure 5.7 Executing Node.js Project Creation in Windows PowerShell      | 101 |
| Figure 5.8 EC2 Instance  | 102 |
| Figure 5.9 EC2 Key   | 103 |
| Figure 5.10 EC2 Inbound Rules  | 103 |
| Figure 5.11 AWS S3 Bucket Overview                                       | 103 |
| Figure 5.12 IAM Users Permissions  | 104 |
| Figure 5.13 IAM Role Permissions   | 104 |
| Figure 5.14 Connection to psql through EC2 instance                      | 105 |
| Figure 5.15 Running React Native   | 106 |
| Figure 5.16 Starting Android Emulator                                    | 107 |
| Figure 5.17 Starting pm2   | 108 |
| Figure 5.18 List of Database Tables                                      | 108 |
| Figure 5.19 Visual Code Studio   | 109 |
| Figure 5.20 Nano   | 110 |
| Figure 5.21 Nginx Configuration  | 111 |
| Figure 5.22 Login Screen   | 132 |
| Figure 5.23 Faculty Manager, Coordinator, Moderator Main Container       | 133 |
| Figure 5.24 Supervisor Main Container                                    | 133 |
| Figure 5.25 Announcement Screen  | 134 |
| Figure 5.26 Project Screen   | 135 |
| Figure 5.27 Meeting Screen   | 136 |
| Figure 5.28 Presentation Screen  | 137 |
| Figure 5.29 Records Screen   | 138 |

# **Chapter 1: INTRODUCTION**

## ***1.1 Project Overview:***

This project is to develop a mobile app for lecturers who are supervising Final Year Projects (FYP). The mobile app is to complement the functionalities in the FYP System, to support remote access when the lecturers are not in front of their computers.

During the project registration phase, the app facilitates seamless communication among university personnel involved in project management.. Lecturers, as part of the university personnel, can conveniently utilize the app to perform essential functions such as confirming or cancelling projects. This streamlined process enhances efficiency and simplifies project management for university personnel.

During the project activities, the app can support the supervision of students such as uploading and downloading project reports, meeting logs and keeping an up to date with the final year presentation. The requirements for these would be determined in the project and the prototype would demonstrate all these functions.

## ***1.2 Project Objectives:***

The objectives of this project are:

- To improve the usability of the FYP System.
- To study the requirements of the app to support FYP supervisors.
- To design and develop a prototype based on the requirements.

### ***1.3 Goals of project:***

The main goal of this project is to develop an FYP mobile application specially curated towards the personnel of Multimedia University (MMU). And this includes the supervisors, faculty managers, coordinators, and moderators of Multimedia University (MMU).

In addition, the project aims to create a convenient way for the staff of Multimedia University (MMU) that is associated with the Final Year Project to access the FYP (Final Year Project) system using a mobile application that is available in two mobile operating systems, that is, iOS and Android.

Other than that, this project also aims to meet the requirements and improve on the current system as well as implementing it in a mobile platform. Overall, the goal of the project is to:

- Develop an FYP Mobile application for supervisors, faculty managers, coordinators, and moderators of Multimedia University (MMU).
- Optimization for iOS and Android.
- Design and implement complete requirements.

## ***1.4 Project Plan***

### ***1.4.1 Milestones***

Table 1.1 Milestones for Trimester 1

| Week | Milestones   | Due date   |
|------|--|--|
| 2    | Project Plan   | 07/10/2022   |
| 4    | Background Study   | 22/10/2022   |
| 8    | Requirements<br>Use Case Diagram<br>Class Diagram<br>Activity Diagram                                    | 19/11/2022<br>29/10/2022<br>05/11/2022<br>12/11/2022               |
| 9    | First Draft of Report  | 17/12/2022   |
| 10   | Design & Process Models<br>Software Architecture<br>Sequence Diagram<br>Screen Design<br>Data Dictionary | 31/12/2022<br>24/12/2022<br>24/12/2022<br>31/12/2022<br>31/12/2022 |
| 14   | Prototype  | 21/01/2023   |
| 15   | Report Submission  | 27/01/2023   |
| 16   | Presentation   | 14/02/2023   |

Table 1.2 Milestones for Trimester 2

| Week | Milestones   | Due date   |
|------|--|--|
| 3    | Project Plan<br>Review of Past Report<br>Gantt Chart<br>Update Plan and Requirements   | 07/04/2023<br>07/04/2023<br>31/03/2023<br>07/04/2023                             |
| 6    | Update System Design<br>Review Database & Software Architecture<br>Update Database & Software Architecture<br>Setup Database | 28/04/2023<br>14/04/2023<br>21/04/2023<br>28/04/2023                             |
| 9    | First Draft of Report  | 19/05/2023   |
| 12   | Development<br>Supervisor<br>Faculty Manager<br>Moderator<br>Coordinator<br>Admin  | 09/06/2023<br>12/05/2023<br>26/05/2023<br>02/06/2023<br>09/06/2023<br>09/06/2023 |
| 13   | Testing<br>Test Plan & Data<br>Testing and Evaluation  | 16/06/2023<br>16/06/2023<br>16/06/2023   |
| 14   | Final Draft of Report  | 23/06/2023   |
| 15   | Final Report Submission  | 13/07/2023   |
| 16   | Presentation   | 18/07/2023   |

## 1.4.2 Gantt Chart



Figure 1.1 Gantt Chart part 1 for Trimester 1

The figure 1.1 Gantt chart shows part 1 of the timeline for the final year project trimester 1. The main activities for part 1 are the project planning, background study, and the requirements of the project. Project planning includes the making of this gantt chart, the identifying of objectives and goals, and the final review of Chapter 1: Introduction. For the background study, this mostly consists of doing research and the writing of Chapter 2: Background Study. During the background study, questionnaire survey will partake which after the data from the survey will be collected and included in Chapter 2, Literature Survey. Following the requirements of the project, this involves the in-depth research and planning for Chapter 3: Requirements. This includes progression of multiple diagrams, essentially the use case diagram, the activity diagram, and the class diagram.



Figure 1.2 Gantt Chart part 2 for Trimester 1

The figure 1.2 Gantt chart shows part 2 of the timeline for the final year project trimester 1. Part 2's primary activities are the completion of the first half of the report, the design and process models, prototyping, and the final draft of the report designating that the project is finalized. Design and Process Models include the progression of the software architecture, the sequence diagrams, the early design of the screen, and the data dictionary. This all contributes to the completion of Chapter 4: Design. The first development of the mobile app is carried out during prototyping. Finally, the final draft of the report is done and turned in.

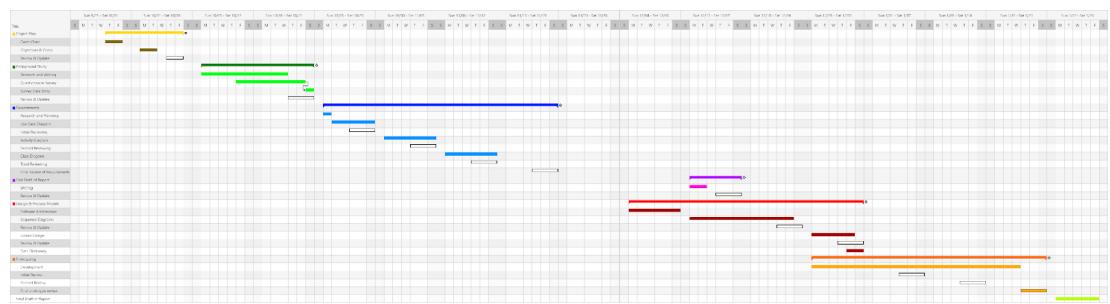


Figure 1.3 Full Gantt Chart for Trimester 1

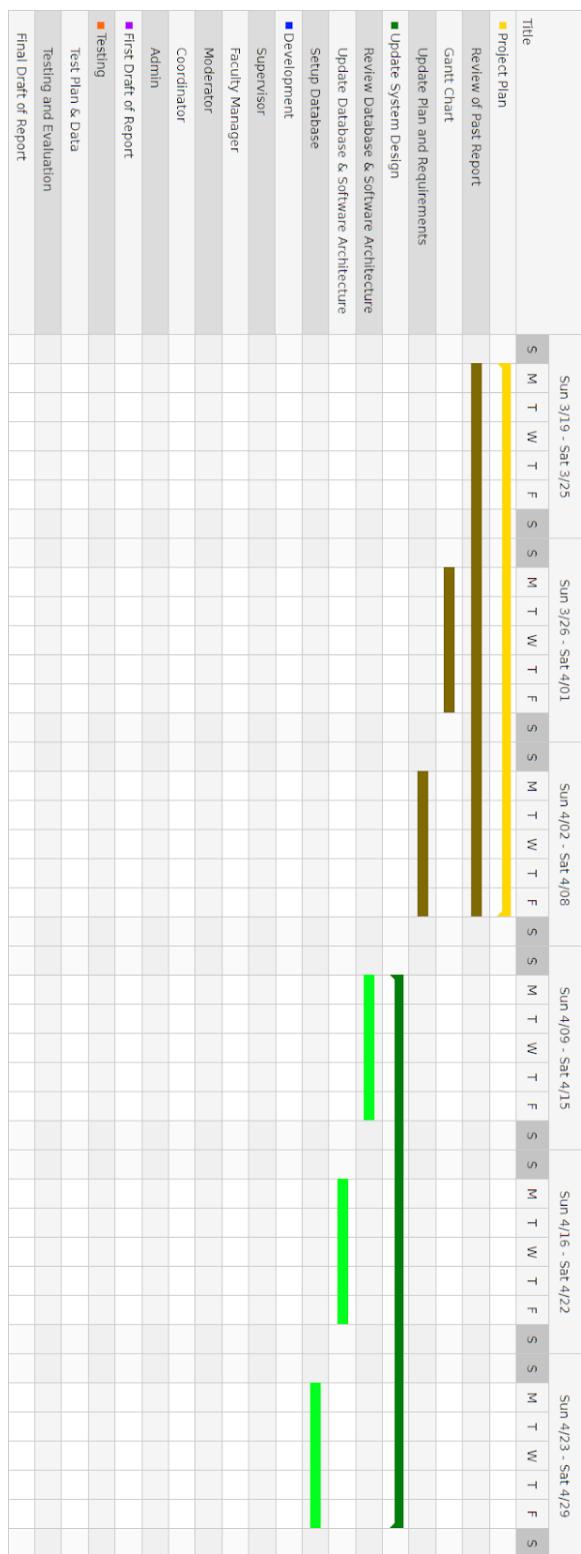


Figure 1.4 Gantt Chart part 1 for Trimester 2

The figure 1.4 Gantt chart shows part 1 of the timeline for the final year project trimester 2. The main activities for part 1 are the project planning, and the update of the system design. Project planning includes the making of this gantt chart, the review of the past interim report, and the update of Chapter 2: Background Study and Chapter 3: Requirements. For the update of system design, this mostly consists of reviewing and updating the database and software architecture in Chapter 4: Design. During this time, the database should be completed.

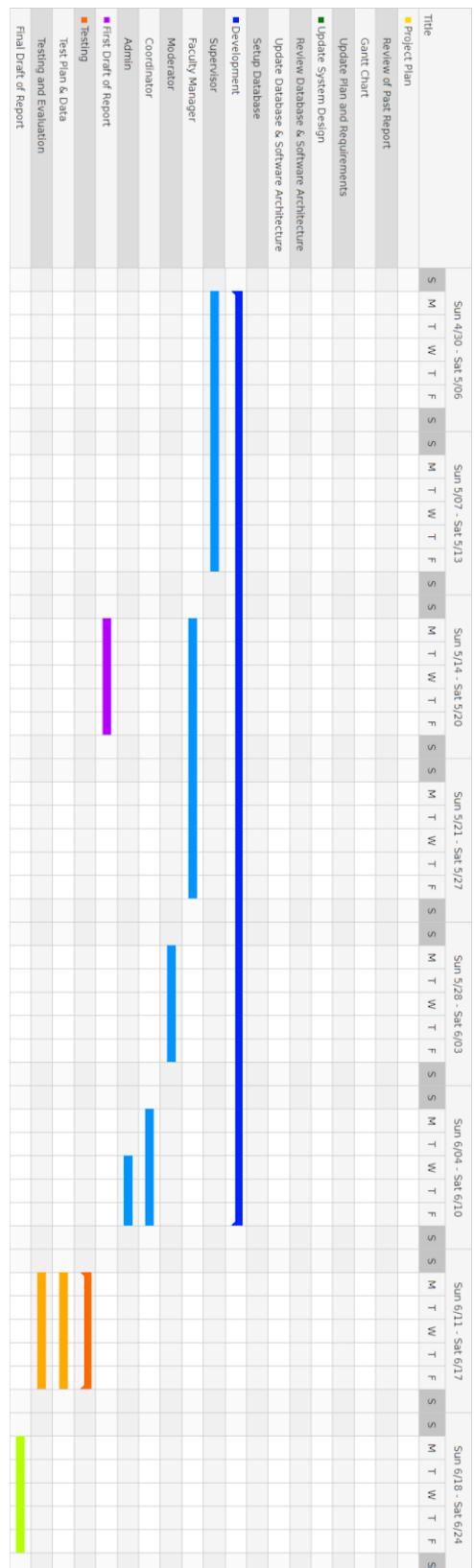


Figure 1.5 Gantt Chart part 2 for Trimester 2

The figure 1.5 Gantt chart shows part 2 of the timeline for the final year project trimester 2. Part 2's primary activities are the development of the system, completion of the first half of the report, the testing of the system, and the final draft of the report designating that the project is finalized. Development of the system includes the front-end and back-end of the system. The first draft of the report is carried out during the development process. The testing phase is done after the completion of the system and will contribute to Chapter 5: Implementation and Chapter 6: Testing, of the report. Finally, the final draft of the report is done and turned in.

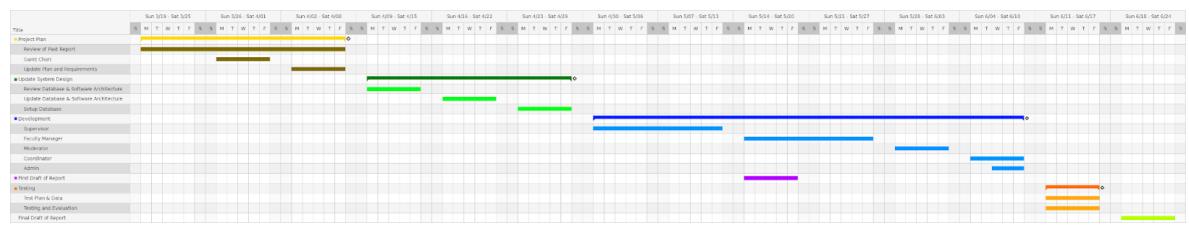


Figure 1.6 Full Gantt Chart for Trimester 2

## **Chapter 2: BACKGROUND STUDY**

### ***2.1 Existing Systems***

As of now, the main current FYP (Final Year Project) system that Multimedia University (MMU) uses for our faculty is called “FCI FYP SYSTEM v2.8” which is a web-based app. This is the main system that our University’s supervisors and faculty manager use. In addition, this web-based app is private and in order to gain access to this web, supervisors need to connect to MMU’s intranet or instead use a remote access VPN (Virtual Private Network). This is also true for student’s and faculty manager’s access. For this research in particular, we will have an in-depth look at MMU’s FYP (Final Year Project) system along with other existing Universities’ and/or institutes’ FYP (Final Year Project) systems to figure out the similarities, strengths, and flaws of every system. This includes the reviews of current FYP (Final Year Project) system of UiTM (Universiti Teknologi MARA), UM (University of Malaya), UTP (Universiti Teknologi PETRONAS), and, UNIMAS (Universiti Malaysia Sarawak). The details and data gained of other Universities’ and/or institutes’ FYP (Final Year Project) systems are shared through the feedbacks and opinions of the staffs and lecturers of the resultant Universities and/or institutes which in this paper, the identifications of said staffs and lecturers are and will be kept private. As a result of this, we will mostly study the medium of the chosen existing FYP (Final Year Project) system including its functionalities, features, and modules. In the following, we will have more detailed discussions on our (MMU) current FYP (Final Year Project) system along with the four Universities’ and/or institutes’ FYP (Final Year Project) system individually. In

addition to this, we will also have a look at two applications found in Google Play and App Store, “BINUS Mobile for Lecturer” and “Sister For Lecturer UNEJ”. These two applications were curated for University lecturers.

### ***2.1.1 Multimedia University (MMU)***

Currently, our University’s Faculty is mainly using a web-based system. As of now, the current version is v2.8. There are multiple reports of limitations of our current system, however they are still considered useful and secure. To require access to this web-based system, connection to the University’s intranet is required. Another possibility of accessing the system is through VPN. Though, the use of a third-party software called “CISCO AnyConnect” is needed for this. The guide to access the system through VPN is readily available on the internet through MMU’s (Multimedia University) website. Now the current system is considered limited due to the absence of multiple features such as a way to communicate with students without using a third party software/system/application, project management/tracker, file repository and so on. The UI is also considered visually unpleasant or “a mess” by some students and supervisors alike”. The features that the current system currently provides as a supervisor are as follows:

- Submit project proposals
- Assign students to a project
- View list of all projects
- Update projects
- Pass or fail a student on a project

The features that the current system currently provides as a student are as follows:

- View projects
- View documents
- Choose a presentation slot

The features that the system currently provides as a faculty manager ar as follows:

- Approval of proposed projects by supervisors
- Upload documents (ex. guidelines)
- Update homepage announcements
- Approve user registration
- Update the presentation slots

Based on the features depending on the role of each user, we can conclude that the current system is insufficient or limited when it comes to features. More could be done to upgrade its capabilities. However its limited capabilities as of now, it is still usable and a very secure system. Furthermore, this paper aims to support the development of a mobile application with further improvements to the current features along with some added useful features. Further inquiries to the University's FYP (Final Year Project) supervisors of the system will be done through a survey.

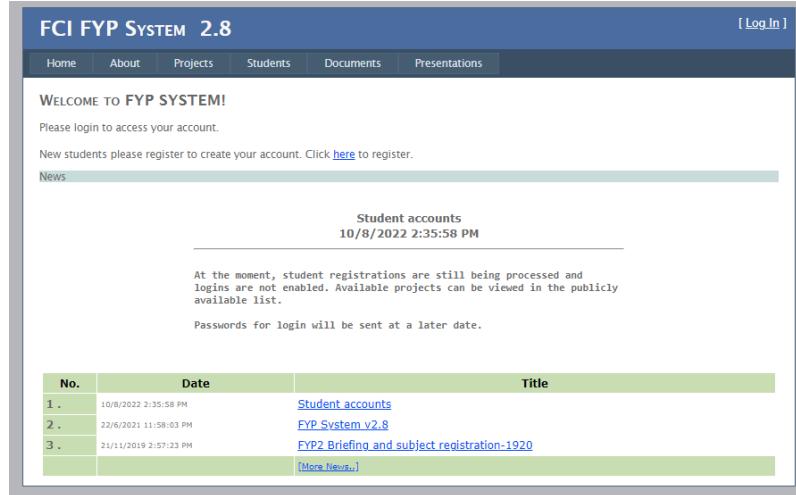


Figure 2.1 MMU FCI FYP Online System Front Page

### **2.1.2 Universiti Teknologi MARA (UiTM)**

Universiti Teknologi MARA (UiTM) is a public university based primarily in Shah Alam, Selangor. UiTM has more than 30 campuses available around Malaysia. There are many faculties available in UiTM but for this particular research, we will look at UiTM's Faculty of Electrical Engineering and how they handle their management in the FYP (Final Year Project). UiTM's system allows for students, supervisors, and coordinators to utilize its system. Their www-based system is easy to access which can be freely found on the “internet” as there is no need for intranet or VPN access. In order to access, only Staff/Student ID and password is needed along with a verification captcha. From the inquiries, it is concluded that their system's medium and features are as follows:

- www-based (no VPN needed), login required
- Project approvals, selections, and allocation
- Upload and download of relevant documents
- Submission and grading of projects

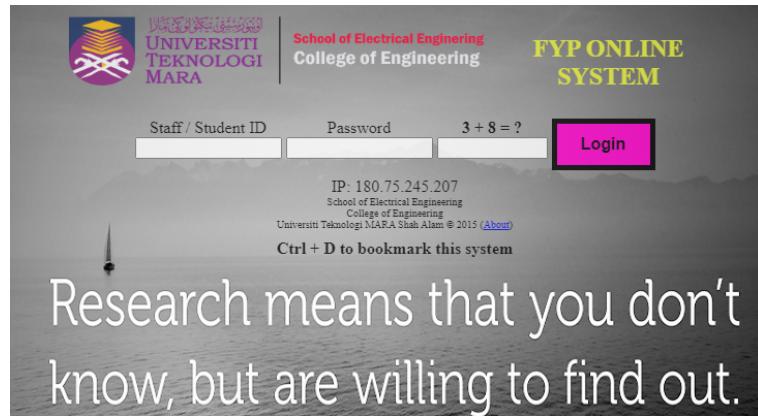


Figure 2.2 UiTM FEE FYP Online System Front Page

### ***2.1.3 University of Malaya (UM)***

The University of Malaya (UM), situated in Kuala Lumpur, Malaysia, is a publicly funded research institution that has gained recognition as the oldest higher education establishment in the country. Its reputation as a premier educational facility has been acknowledged by two respected global ranking organizations. UM's FYP (Final Year Project) system utilizes a www-based service in which you need to connect to its intranet in order to gain access. Supervisors, students, and coordinators all use this www-based system. Though, there is a mobile application available directly integrated from its www-based system, but it is now scrapped. From the inquiries, it is concluded that their current system's medium and features are as follows:

- www-based (intranet, no VPN), login required
- Project allocations
- Upload and download of relevant documents
- Submission of projects

#### ***2.1.4 Universiti Teknologi PETRONAS (UTP)***

Established on January 10, 1997, Universiti Teknologi PETRONAS is a research-focused university that is fully owned by Petroliam Nasional Berhad, a leading oil and gas company in Malaysia. The university's campus is situated in Seri Iskandar, Perak, Malaysia. UTP's FYP (Final Year Project) system is very limited according to the inquiries. Their current system is www-based and only accessible via their intranet. Their system is currently only accessible by supervisors and coordinators. Other than that, they mostly rely on physical documentation from their students. From the inquiries, it is concluded that their current system's medium and features are as follows:

- www-based (intranet, no VPN), login required
- Project allocations

#### ***2.1.5 Universiti Malaysia Sarawak (UNIMAS)***

Universiti Malaysia Sarawak (UNIMAS) is a public university located in Kota Samarahan, Sarawak. Currently, their www-based system is easily accessible from the internet. Before this, connecting to their intranet is needed to gain access. This is however not needed anymore as their system is upgraded to v2.0. Students, supervisors, and coordinators all use this current system. In order to access, username and password are required for login. From the inquiries, it is concluded that their current system's medium and features are as follows:

- www-based (no VPN needed), login required
- Announcements (only by coordinators)
- Project approvals, selections, and allocation

- Project Management (create task, schedules, milestones, and set up meetings)
- Upload and download of relevant documents
- Submission and grading of projects



Figure 2.3 UNIMAS FYP Online System Front Page

### **2.1.6 BINUS Mobile for Lecturer**

This mobile application developed by and for Bina Nusantara University, Indonesia also known as BINUS is specially designed for their lecturers. Currently this app is available for both apple and android users. According to their app description on Google Play, “BINUS Mobile for Lecturer is an all in one mobile app that will enhance lecture experience”. The app currently supports making announcements, scheduling for classes and presentations, assignments tracker, and even has book borrowing and a forum built inside the app. Though the app doesn’t focus on FYP (Final Year Project), it does have some useful features for lecturers. In short, the app’s key features are:

- Application-based

- Versatile scheduling
- Built-in forum
- Assignments tracker

### ***2.1.7 Sister For Lecturer UNEJ***

This mobile application is intended for University of Jember, Indonesia (UNEJ) Lecturers. According to the app's description in Google Play, "SISTER For Lecturer (SFL) is intended for application to a mobile UNEJ Lecturer / online can make entry value, Final guidance and guardianship. Lecturers can also monitor: academic achievement, attendance in lectures, the date of payment of fees / UKT student guidance. Besides, lecturers can also give feedback, suggestions, questions or complaints to UNEJ. Periodically UNEJ will also be able to send a notification / notification to the lecturer". The app is available on both android and apple platforms. Some of the features are:

- Application-based
- Project Management (create task, schedules, milestones, and set up meetings)
- Communication (Live chatting)
- Monitoring (Tasks, attendance, class schedules)

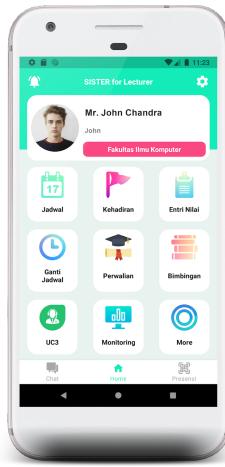


Figure 2.4 Sister for Lecturer UNEJ Home Screen (from Google Play)

## 2.2 Literature Review

Based on the brief study of other existing systems, we reached the understanding of each system's functionalities including its similarities and differences of features. We concluded that every Universities' and/or institution's FYP (Final Year Project) system has its own strengths and flaws. Table 2.1 summarizes the presence and absence of modules/functionalities of each system where "X" means it is present and blank meaning it is absent.

Table 2.1 Modules of Each System

|        | Project Allocation | Communication | Project Management | File Sharing and Repository | Submission and Grading |
|--------|--------------------|---------------|--------------------|-----------------------------|------------------------|
| MMU    | X                  |               |                    |                             |                        |
| UiTM   | X                  |               |                    | X                           | X                      |
| UM     | X                  |               |                    | X                           | X                      |
| UTP    | X                  |               |                    |                             |                        |
| UNIMAS | X                  |               | X                  | X                           | X                      |

|             |  |   |   |  |  |
|-------------|--|---|---|--|--|
| BINUS (app) |  |   | X |  |  |
| UNEJ (app)  |  | X | X |  |  |

The FYP management system is really nothing new. Several FYP management systems have been created and are utilised in educational institutions, according to the previous study. As an illustration, the FYP management system was created by (M.A. Mohamed, 2017; Syila Izawana Ismail, 2017; Abdul kareem Ademola, 2013; Marini Abu Bakar, 2011; Wahab, 2012). However, the majority of the FYP administration system was created using specific programming and gateway concepts for Degree, Master's, and Ph.D. students. Because they may be used for both business and educational activities, portals are widely used in the community. It gained popularity because it provides consolidated access to all pertinent applications and material. According to (Abdul Kareem, 2013), portals give users access to resources they require as well as collaborative tools. However, it will require someone with a strong expertise in programming to create portals. In addition to portals, mobile apps, games, and the online are other possibilities that can be used as educational tools, according to (Affandi & Omar, 2018). Mobile applications can be a suitable choice for cutting-edge educational technology since they enable learning to take place in both portals and interesting and meaningful situations. According to (Sunga, Changb, & Liu, 2015), mobile devices offer a great deal of potential to develop into a powerful instructional tool. In contrast, (Gaurav R. Sawarkar, 2019) asserted in their research that mobile applications may boost teaching and learning and increase user engagement. Though, it is thought that mobile applications are more accessible, appealing, and pedagogically constructive for users according to

(Buddhini, Gaya R., Chamali, & Malinda C.B, 2019). In accordance with this research, a google form survey was created specifically for Multimedia University's FYP (Final Year Project) supervisors of the Faculty of Computer and Informatics. The survey was proposed to gather information about the usage of the current system and assist in developing the perfect FYP (Final Year Project) mobile application.

## 2.2.1 Survey



**FCI FYP SYSTEM SURVEY**

This survey aims to get supervisor's feedback and opinion on the current FCI FYP system (v2.8) and assist in the future development of an FYP Mobile App.

Is the current system reliable? \*

1      2      3      4      5

Very unreliable                                    Very reliable

How difficult or easy is it to use the current system? \*

1      2      3      4      5

Very difficult                                    Very easy

What is the main device you use to access the FYP system? \*

Please choose only 1

PC  
 Laptop  
 Smartphone

How unsatisfied or satisfied are you with the current system's features? \*

1      2      3      4      5

Not at all satisfied

Extremely satisfied

What feature/module do you most likely want to see added to the system? \*

You may choose more than 1

- Communication Module (Live chatting, commenting etc.)
- Project Management Module (Task scheduler, milestone tracking etc.)
- File Sharing and Repository Module (Upload and download)
- Submission and Grading Module
- Other: \_\_\_\_\_

What third party app(s) do you use to communicate with your students on their FYP project? (ex. making announcements, setting up meetings, etc.) \*

You may choose more than 1

- Google Classroom
- Microsoft Teams
- Email
- WhatsApp
- No, I don't use any third party apps
- Other: \_\_\_\_\_

What is the main device you use to communicate with FYP students under your supervision? \*

Please choose only 1

- PC
- Laptop
- Smartphone

How effective do you think it would be to have a mobile app  
that accommodates the current FYP System? \*

1      2      3      4      5

Not at all effective

Extremely effective

What are you looking forward to the most in a mobile app of an FYP system? \*

You may choose more than 1

- To be able to communicate with students
- To be able to track student's progress
- To be able to upload and download files
- To be able to grade student's work
- To be able to more easily propose and view projects
- Other: \_\_\_\_\_

This form was created inside of Multimedia University. [Report Abuse](#)

Google Forms

## 2.2.2 Survey Analysis

The survey conducted from 28th October 2022 to 4th November 2022, resulted in 23 MMU (Multimedia University) FCI lecturers participants.

1) Is the current system reliable?

23 responses

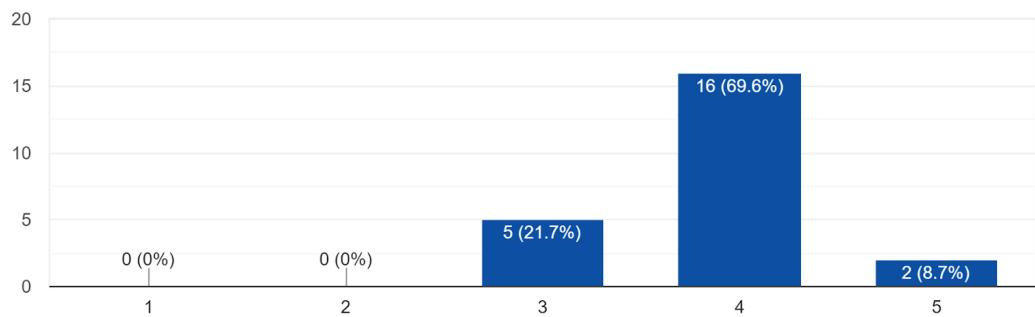


Figure 2.5 Result for question 1 of survey

Out of 23 participants, 16 (69.6%) gave a rating of 4 (reliable), 5 (21.7%) gave a rating of 3 (moderately reliable), 2 (8.7%) gave a rating of 5 (very reliable), and none (0%) gave a rating of 1 (very unreliable) and 2 (unreliable).

2) How difficult or easy is it to use the current system?

23 responses

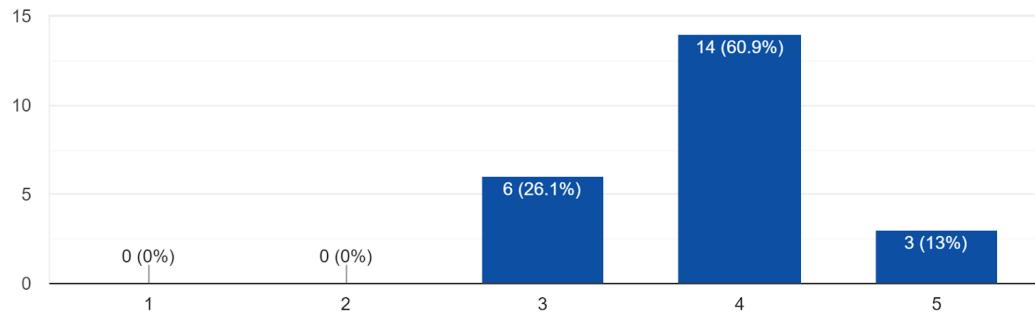


Figure 2.6 Result for question 2 of survey

14 (60.9%) out of 23 participants gave a rating of 4 (easy to use), 6 (26.1%) gave a rating of 3 (moderate), 3 (13%) gave a rating of 5 (very easy to use), and none (0%) gave a rating of 1 (very difficult) and 2 (difficult).

3) What is the main device you use to access the FYP system?

23 responses

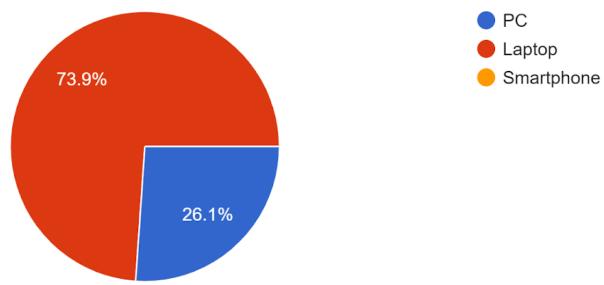


Figure 2.7 Result for question 3 of survey

17 (73.9%) out of 23 participants uses a laptop as their main device to access the FYP system, 6 (26.1%) uses a PC as their main device to access the FYP system, while none (0%) uses smartphone as their main device to access the FYP system.

4) How unsatisfied or satisfied are you with the current system's features?

23 responses

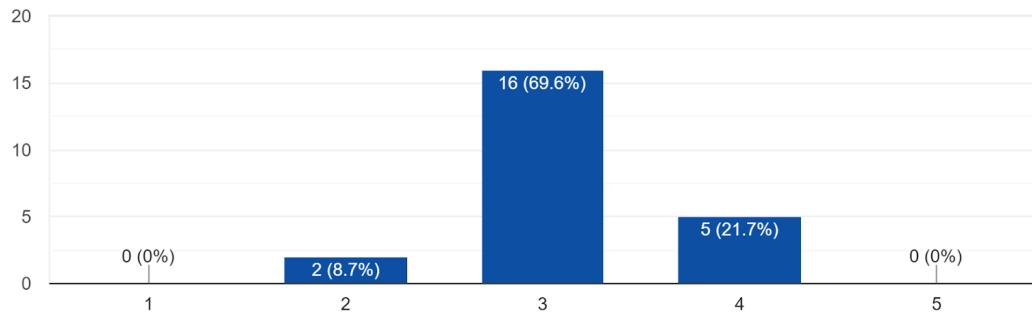


Figure 2.8 Result for question 4 of survey

16 (69.6%) out of 23 participants gave a rating of 3 (moderately satisfied), 5 (21.7%) gave a rating of 4(satisfied), 2 (8.7%) gave a rating of 2 (unsatisfied), while none (0%) gave a rating of 1 (very unsatisfied) and 5 (very satisfied).

5) What feature/module do you most likely want to see added to the system?

23 responses

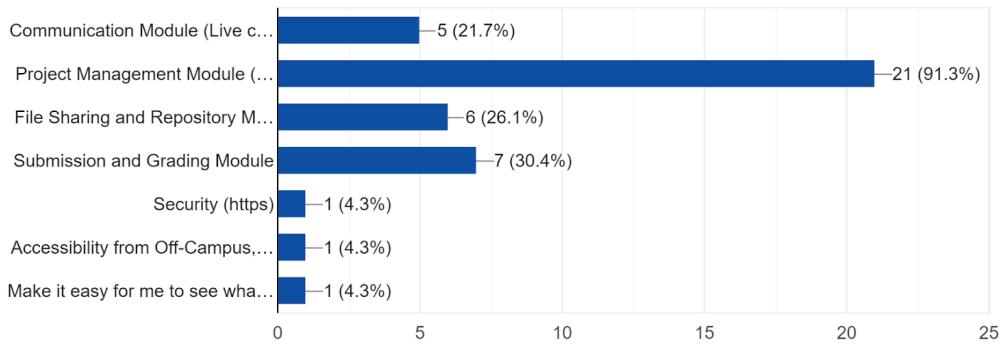


Figure 2.9 Result for question 5 of survey

21 (91.3%) out of 23 participants wants project management module, 7 (30.4%) wants submission and grading module, 6 (26.1%) wants file sharing and repository

module, 5 (21.7%) wants communication module, while 1 (4.3%) wants security, accessibility from off-campus, and an easier project proposal respectively.

6) What third party app(s) do you use to communicate with your students on their FYP project? (ex. making announcements, setting up meetings, etc.)

23 responses

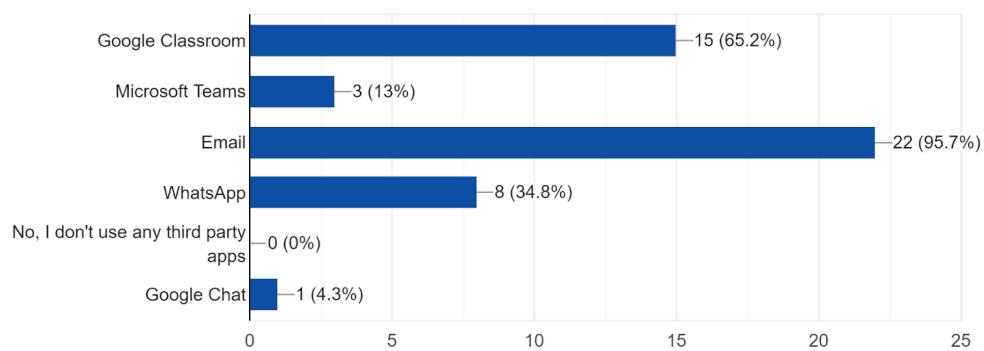


Figure 2.10 Result for question 6 of survey

22 (95.7%) out of 23 participants use email to communicate with their students, 15 (65.2%) use Google Classroom, 8 (34.8%) use WhatsApp, 3 (13%) use Microsoft Teams, while 1 (4.3%) uses Google Chat.

7) What is the main device you use to communicate with FYP students under your supervision?

23 responses

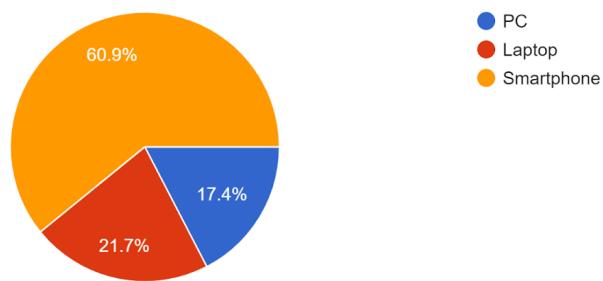


Figure 2.11 Result for question 7 of survey

14 (60.9%) out of 23 participants uses a smartphone as their main device to communicate with their students, 5 (21.7%) uses a laptop as their main device to communicate with their students, while 5 (17.4%) use a PC as their main device to communicate with their students.

8) How effective do you think it would be to have a mobile app that accommodates the current FYP System?  
23 responses

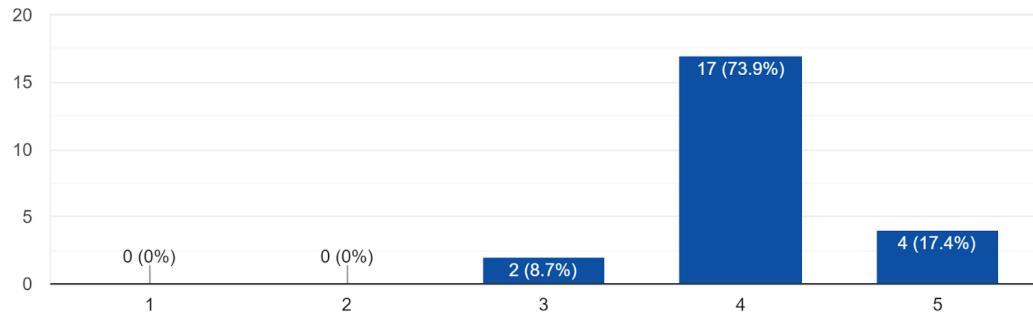


Figure 2.12 Result for question 8 of survey

17 (73.9%) out of 23 participants gave a rating of 4 (effective), 4 (17.4%) gave a rating of 5(very effective), 2 (8.7%) gave a rating of 3 (moderately effective), while none (0%) gave a rating of 1 (very ineffective) and 2 (ineffective).

9) What are you looking forward to the most in a mobile app of an FYP system?

23 responses

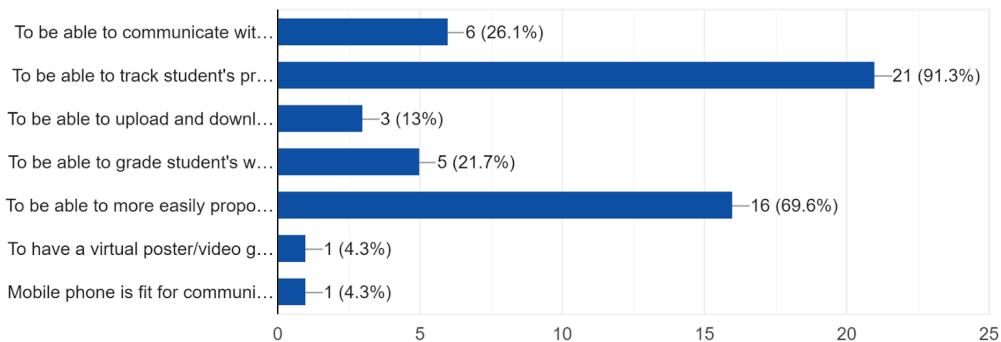


Figure 2.13 Result for question 9 of survey

21 (91.3%) out of 23 participants choose “To be able to track student’s progress, 16 (69.9%) choose “To be able to more easily propose and view projects”, 6 (26.1%) choose “To be able to communicate with students”, 5 (21.7%) choose “To be able to grade student’s work”, 3 (13%) choose “To be able to upload and download files”, while 1 (4.3%) choose other.

### 2.2.3 Survey Conclusion

From our findings of the survey, we can conclude that a majority of MMU’s (Multimedia University) personnel are somewhat satisfied with the current system but want added functionality. Majority also believe that a mobile application would be useful. The most sought out functionality in demand is the project management module, while the other modules suggested are met with minimal votes. To conclude, we will be focusing on the project management functionality for our mobile application.

### ***2.3 Problems to be solved***

Many of the FYP (Final Year Project) system activities are carried out manually and heavily rely on conventional means of communication and assessment submission, according to assessments of existing systems and literature surveys. For instance, when choosing and allocating projects, students must join groups and select their chosen project themes by emailing or texting their preferred lecturer or supervisor. Supervisors will then need to get in touch with the coordinator or faculty manager to let them know about the situation. When 20 or more students arrive, the work might become tedious and time-consuming. Many of the project deliverables, aside from the project allocation, are submitted in physical copy or by email. Supervisors typically schedule visits with students once or twice a week to assess their progress. These face-to-face meetings are limited to an hour and a half due to the supervisors' responsibility of mentoring multiple groups each year. During these sessions, students only have the opportunity to report their achievements and establish goals for the forthcoming weeks, without sufficient time to discuss or address any project challenges. Email communication with students, which is necessary for keeping the project on track, is often hindered by the time it takes to receive a response, making it challenging for supervisors to stay updated on the project's progress. Furthermore, supervisors may struggle to keep track of students' expected progress, possibly due to the absence of proper project management tools, their hectic semester schedule, or simply the hassle of turning on their own computer. Furthermore, our survey questionnaire conducted found out that the majority of our lecturers, supervisors and faculty managers prefer better tracking and monitoring of students. It is decided to

create a mobile application of the final year project management system to support the University's/Institutions' supervisors entire process and make it easier. This is done to address the various issues that were encountered.

## ***2.4 Technological Background***

The process of creating software that primarily operates on a mobile phone or other handheld digital device is known as mobile app development. It is a sector that is constantly expanding and has become essential for any enterprise we may imagine. Although there are many different smartphones on the market, Android and iOS account for the majority of sales. Let's examine the global market share of mobile operating systems:

Table 2.2 Mobile Operating System Market Share Worldwide (Statcounter, 2021)

| Mobile Operating System | Percentage (%) |
|-------------------------|----------------|
| Android                 | 71.81%         |
| iOS                     | 27.43%         |
| Samsung                 | 0.38%          |
| KaiOS                   | 0.14%          |
| Unknown                 | 0.14%          |
| Linux                   | 0.02%          |

With this kind of market, which presently accounts for 99.24% of the global market, individuals typically look for creation of mobile apps for Android and iOS. These two Operating Systems are the main targets of the modern app development industry. While conventional software development is still relevant, app development replaces them by offering functionality that standard desktop or web applications might not

have. For instance, the mobile phone is a far more potent deployment device than a PC due to the availability of GPS, Compass, Accelerometer, Sensors, Bluetooth, etc. We must select a programming language and/or a suitable framework when designing a mobile app, just as we would when constructing normal software (Rakestraw, 2013). It is important to understand the different types of mobile app development that are available, even though we might only look at Android and iOS development separately (Joorabchi, 2013).

#### ***2.4.1 Native Mobile App Development Tools***

A "native development tool" is a type of software that enables developers to create programs specifically for a particular family of operating systems, platform, or device, such as Android, iOS, or Windows (however, support for Windows Mobile ended in December 2019). By using the native programming language of the platform, a native app is crafted and coded specifically for a particular mobile platform (Jones, 2020), such as:

- iOS (Objective-C or Swift)
- Android (Java or Kotlin)

##### ***2.4.1.1 Xcode***

Xcode has revolutionized software development by providing a cutting-edge platform for developing applications. Utilizing Xcode tools and the Swift programming language, developers are able to experience a truly interactive development process. With the ability to type UI code directly into Interface Builder, changes are immediately reflected in the design canvas, creating a seamless development experience. Xcode is an all-in-one solution for developers, providing

everything necessary to create applications for Mac, iPhone, iPad, Apple TV, and Apple Watch. It offers a single, streamlined process for user interface design, coding, testing, and debugging, making it an indispensable tool for developers (Apple, 2022).

#### ***2.4.1.2 Android Studio***

Google developed Android Studio, a comprehensive developer tool for creating Android applications. It is highly valued by Android developers for its efficient implementation editor. Android Studio streamlines the development process by providing shortcuts for both coding and design, and its user-friendly layout designer makes it easy to use. The platform also features drag-and-drop tools for designing project layouts, further simplifying the development process.

#### ***2.4.2 Cross-Platform Mobile App Development Tools***

Cross-platform mobile development enables programmers to create applications for multiple platforms or mobile operating systems simultaneously, allowing them to use the same code base across many platforms. This can reduce the time and cost required for app development and result in generic apps that can be used on both iPhones and Android phones. However, there are some drawbacks to this approach, such as an increased likelihood of platform-specific bugs and potentially lower quality user-interface and performance compared to native apps. The extent of these drawbacks may vary depending on the specific platform, system, and application being used (Jones & Smith, 2019).

#### ***2.4.2.1 React Native***

React Native is an innovative cross-platform framework, designed by the Facebook community, that uses JavaScript technology to build high-quality native-like applications. With the ability to develop applications using a single code base, developers can create seamless and native-like experiences for users. This JavaScript-based framework allows developers to craft top-notch native applications, regardless of whether they were developed by a single team using React and JavaScript (React Native, n.d.).

#### ***2.4.3 Backend***

A mobile app's backend, which stores, protects, and processes the data, is essential (Pereira, 2018). A backend app functions similarly to a server for users to sort the necessary data. It might be said to relate to the processes that take place in the background as you use an app to complete a task. This can involve playing an online movie or logging into your account. Frontend apps employ backend apps to send information more specifically. Protocols are used by a front-end app to transmit data to the backend across the internet. These protocols are exclusively intended for use in app communication. The frontend app receives database information thanks to code created by backend developers. As a result, the backend programme generates all the essential output that we require.

#### ***2.4.3.1 Node.js***

The Chrome V8 engine uses Node.js, an open-source JavaScript runtime environment. The event-driven, non-blocking I/O approach it employs makes it ideal for creating quick-to-market, scalable applications. You can build extremely scalable

apps with Node.js that enable handling several concurrent user requests (Smith J., 2018). The following application types support Node.js as a backend:

- Applications for social media – Social media apps have a lot of user interactions to handle. Therefore, when developing such an app, efficiency and scalability are the most important elements to take into account. Well-known social networking applications like Twitter and LinkedIn currently employ Node.js.
- Publishing applications – Cartovera, a fantastic example of a publishing application using Node.js done in distributed servers,
- Single-page applications — These programmes just use one page of the browser to operate. As an illustration, Gmail displays all of the most current messages on a single page.
- Applications for real-time chat — Based on Node.js, React Native, MongoDB, and many other technologies, Remote help is a real-time chat application.
- E-learning platforms – Quizlet has handled over 18 million monthly and 30 million active learners in e-learning platforms.

#### ***2.4.3.2 Ruby on Rails***

Model View Controller (MVC) architecture and the open-source Ruby programming language are the foundations of Ruby on Rails, a framework for developing server-side web applications. Ruby is renowned for being a user-friendly and simple to learn programming language. Ruby on Rails is fantastic for quick prototyping because there are so many Ruby gems available that developers can start using

numerous functionalities right away. You may use RoR (Ruby on Rails) as the backend for your mobile application in the following use cases:

- Applications for online shopping — The majority of the features, including security and authentication, are covered by a number of Ruby gems. Ruby on Rails is also used by Shopify.
- Applications with a high development priority to fulfil strict timeframes.
- Projects using Software as a Service (SaaS) that heavily utilise APIs
- Projects that make use of large databases, such as Github and all of its repositories

#### ***2.4.3.2 Key Differences between Node.js and Ruby on Rails***

Table 2.3 Node.js vs. Ruby on Rails

| Node.js  | Ruby on Rails  |
|--|--|
| Node.js is a JavaScript run-time environment framework that is written in JavaScript.  | Rails is a Ruby-based framework, which is written in Ruby language.  |
| The greatest platform for creating small-scale projects and event-based, non-blocking I/O applications is Node JS.                                 | Rails is a web application framework that works well for metaprogramming and database-backed MVC web applications.                     |
| In comparison to Rails, operating Node.js is much faster (page loads are faster throughout a site). Best advised for creating a quick application. | Node.js is faster than Rails. Best advised for creating a quick application. Ideal for creating sophisticated, extensive applications. |
| With a few archive files that must be installed on the system, Node.js is simpler to set up.   | However, in the case of Rails, you must install a lot of components as well as numerous generators.                                    |
| followed by the Node.js framework, LinkedIn, eBay, Medium, Heap, Sense, etc.   | Dribbble, Airbnb, Scribd, GitHub, Shopify, Soundcloud, Basecamp, etc. are examples of Rails-based applications.                        |
| Node.js gives us the ability to build web servers on our own that can asynchronously process all HTTP requests.                                    | Rails, however, is dependent on the WEBrick, Apache, NGINX, and Cherokee.  |

|   |  |
|---|--|
| It can be used by developers for server-side as well as client-side purposes.   | Unlike the server-side framework Ruby on Rails.  |
| massive market demand The number of users on it is growing at an exponentially faster rate.   | Not as well-liked as NodeJS at the moment.   |
| Deploying an application in Node.js takes longer since you have to search for the modules there and follow the detailed instructions for integrating them with the application. | In comparison to Node.js, Ruby on Rails is thought to be faster and lighter because processes like database migration may be completed with just a few commands.                 |
| Compared to Rails, the learning curve is rather less steep. Developers of any experience level can use it because it is adaptable and simple to comprehend.                     | Although learning Rails is simple, you may need to be familiar with and comprehend some vital and time-consuming attributes, such as objects, classes, and Ruby-level functions. |

#### **2.4.4 Host**

Writing code is insufficient when creating mobile applications. Of course, the entire backend portion will be on the user's device if you created an app like a calculator with extremely basic functions. However, if the app has a backend that performs more challenging tasks, it must be somewhere, and something must guarantee that the requests will be processed there. A service provider like Amazon Web Services (AWS) or Google Cloud Platform's app hosting enables you to execute the backend on servers or in the cloud (GCP). They offer a base on which to host your application. To avoid any confusion, the distribution of the programme is handled through Google Play and/or the AppStore. Hosting for mobile apps is similar to hosting for online apps. The user also performs an action or submits a request, which is handled by the computer hosting the mobile app. For this study, we will look at a few hosting services that are most suitable for this particular application.

#### ***2.4.4.1 Fly.io***

Fly.io is a platform for running backend and full-stack applications globally. The purpose of Fly.io, which is somewhat PaaS-oriented, is to enable developers to self-service complex infrastructure without an ops team. The company makes an effort to eliminate abstractions while minimising infrastructure ugliness. Fly.io provides a comprehensive platform for building modern cloud applications and supports a variety of popular frameworks including Phoenix, Ruby, Node, Next.js and many others (Fly.io: A Global Platform for Building Modern Cloud Applications, 2021). It also comes equipped with a range of convenient distributed services, such as multi-region Postgres.

#### ***2.4.4.2 Backblaze B2***

According to Backblaze's official documentation, B2 Cloud Storage is a straightforward, dependable, and cost-effective solution for object storage (Backblaze, 2021). Every account comes with 10 GB of unrestricted storage and 1 GB of daily unrestricted downloads with no hidden fees. To access B2 Cloud Storage, you need to have a Backblaze account, which you can create by registering on backblaze.com. Upon activation of B2 for your account, you will receive an Application Key Id and an Application Key that allow you to access the APIs. Backblaze offers two API options: S3 Compatible and B2 Native. It is recommended for developers to use the S3 Compatible API for creating new applications and integrations as it is widely supported by a larger selection of SDKs and libraries, and many developers are familiar with Amazon S3. To access features not provided by the S3 Compatible API, such as application key management and lifecycle rules, use

the Backblaze B2 web console or the B2 Native API. With the use of the Backblaze APIs, you can:

- Download, upload, and remove files
- Create and maintain the file storage buckets.
- Manage your account's configuration

#### ***2.4.4.2 WebSocket***

WebSocket is a technology that utilises the internet, as the name suggests. Because browser pages are the principal tool for online data display, the web frequently incorporates them. Non-browser programmes, however, also use online data transmission. A fresh new world of web access without the need for a web browser was made possible by the original release of the iPhone and the iPad. The new smartphone and tablet devices, on the other hand, made use of native apps' strength to provide a distinctive user experience.

Advantages:

- Allows for two-way communication
- Faster data transfer compared to HTTP and AJAX
- Supports message sharing between origins (although this presents security risks)
- Compatible with various platforms such as web, desktop, and mobile
- Has a lower overhead (2 bytes) compared to HTTP (2000 bytes)
- Uses short polling
- Supports data typing, unlike AJAX which can only transfer string data types

Disadvantages:

- Requires a fully HTML5 compatible web browser
- Lacks success methods similar to AJAX
- Does not offer intermediary or edge caching
- Can be challenging to create a simple protocol using HTTP components like statuses and bodies
- If dynamic interaction is not a requirement, HTTP is simpler to develop.

#### ***2.4.4.2.1 AWS vs. Azure vs. Google Cloud***

AWS, Azure, and Google Cloud are all reputable cloud service providers that offer a wide range of services to meet the needs of businesses of all sizes. AWS is the oldest cloud provider and has the largest market share. It offers over 175 services, including compute, storage, networking, and database services, among others. Azure is Microsoft's cloud computing platform and offers similar services to AWS, including virtual machines, databases, and storage solutions. Google Cloud, on the other hand, offers services such as machine learning, big data, and analytics, among others.

When it comes to pricing, AWS is the most cost-effective among the three cloud providers. It offers a pay-as-you-go model that enables businesses to pay only for the services they use. Azure and Google Cloud also offer similar pricing models, but their prices are relatively higher than AWS (AWS, Azure, and Google Cloud: A Comparison, n.d.).

In terms of reliability and availability, AWS has an edge over Azure and Google Cloud. AWS has a global infrastructure that spans across 24 regions and 77 availability zones, making it the most reliable cloud provider. Azure and Google

Cloud have a smaller infrastructure compared to AWS, which can affect their reliability and availability (AWS, Azure, and Google Cloud: A Comparison, n.d..

Another factor that sets AWS apart from Azure and Google Cloud is its security features. AWS has a wide range of security services that provide end-to-end security for applications and data. Its security features include identity and access management, encryption, and threat detection. Azure and Google Cloud also offer similar security features, but AWS is considered the most secure cloud provider (AWS, Azure, and Google Cloud: A Comparison, n.d..

In conclusion, while Azure and Google Cloud are reputable cloud service providers, AWS is the best to use. It offers a wide range of services, is cost-effective, reliable, and has robust security features. These factors make AWS the most preferred cloud provider among businesses of all sizes.

## ***2.5 Proposed Solution***

Based on our technological studies, we know that using React Native which has a cross-platform framework built on JavaScript is the easiest route. Though, Android studio is still needed to set it up. With React Native, deploying for iOS is also possible. The best backend for this would be Ruby on Rails as this will allow for hosting using Fly.io. Fly.io is very cost effective as it offers a good free tier. Other than that, Fly.io also provides backend api plus databases like postgres for free. Though Fly.io sounds good, it does have a big drawback which is that it doesn't support file hosting. Backblaze B2 is another option as host. Backblaze B2's api should be compatible with AWS (Amazon) S3 cloud object storage but the pricing could be a hurdle. Another drawback is that Fly.io and Ruby on Rails' default

architecture can't handle high concurrent load. This is substandard for the communication module which will have a high concurrent load. Another alternative for this is actionable for live connection that is a websocket. This should allow for live chatting (communication module). And with React Native and JavaScript, there is a native support for websocket. Lastly, Ruby on Rails doesn't handle database migration, so because of this, Node.js is the better option as it handles database migration. Using Node.js, backend and mobile coding will only use JavaScript. In conclusion, our best possible solution are:

- React Native as framework.
- Android Studio and XCode to setup.
- Host with WebSocket (JavaScript has native support) - AWS
- Node.js environment (backend and mobile coding fully utilize JavaScript)

# Chapter 3: REQUIREMENTS

## 3.1 System Overview

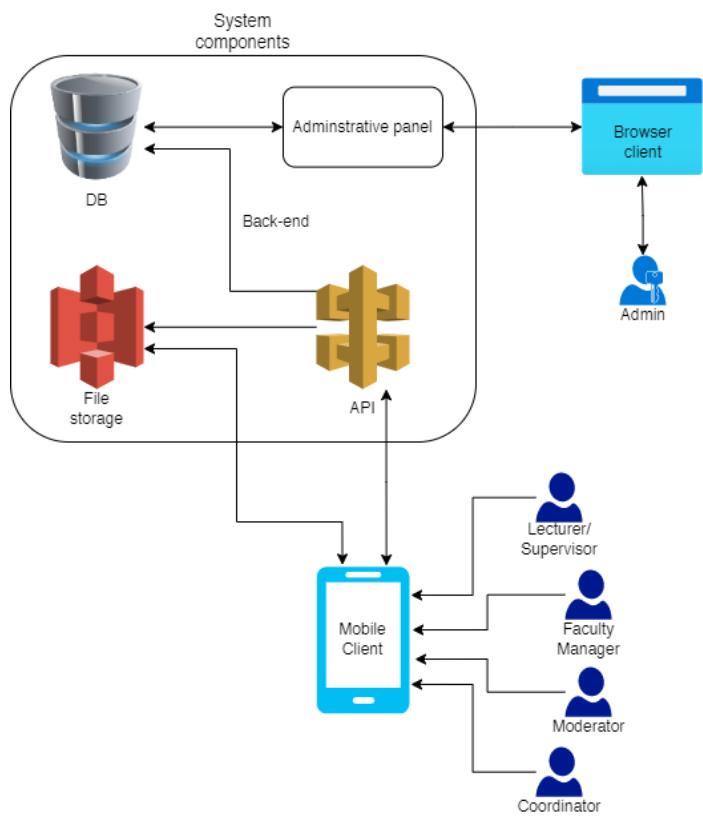


Figure 3.1 System Overview

Fig. 3.1 illustrates the proposed system for an FYP (Final Year Project) Mobile Application mainly curated towards the lecturers of Multimedia University (MMU).

**API:** An API (Application Programming Interface) is a set of protocols, routines, and tools for building software applications. In this context, the API provides a way for other software applications to interact with the system, such as by sending requests for data or performing actions.

File storage: This component is responsible for storing files, such as images, videos, or documents. It provides a way for the system to access and retrieve these files when needed.

DB: A database is a collection of data that is organized in a way that allows for efficient storage, retrieval, and manipulation of that data. In this context, the database stores information related to the system, such as user data, file metadata, or system settings.

An administrative panel is a web-based interface that allows authorized users to manage and configure the system. In this context, the administrative panel provides a way for system administrators to manage user accounts and perform other administrative tasks.

The main users of the system's main mobile client are supervisors, faculty managers, moderators, and coordinators. Both of them will access the system through their personal mobile phones/smartphones. Though, an administrator will also be in place and will access the system through a browser client.

The Supervisor can:

- |                              |                               |                            |
|------------------------------|-------------------------------|----------------------------|
| • Log-in                     | • View announcements          | • View projects(own)       |
| • Add new projects           | • Assign students for project | • Upload project reports   |
| • View meeting               | • Create meeting              | • Update meeting           |
| • View presentation schedule | • View profile info           | • View list of supervisors |
| • View list of moderators    | • View list of students       | • Log-out                  |

The Faculty Manager can:

- |  |                         |                              |
|--|-------------------------|------------------------------|
| • Log-in                                     | • View announcements    | • Create announcements       |
| • Update announcements                       | • View projects         | • View presentation schedule |
| • Setup presentation parameters (date,venue) | • View profile info     | • View list of supervisors   |
| • View list of moderators                    | • View list of students | • Log-out                    |

The Moderator can:

- |                            |                              |                                    |
|----------------------------|------------------------------|------------------------------------|
| • Log-in                   | • View announcements         | • View assigned students' projects |
| • Download project report  | • View presentation schedule | • View profile info                |
| • View list of supervisors | • View list of moderators    | • View list of students            |
| • Log-out                  |                              |                                    |

The Coordinator can:

- |                               |                              |                          |
|-------------------------------|------------------------------|--------------------------|
| • Log-in                      | • View announcements         | • Create announcements   |
| • Update announcements        | • View projects              | • Approve/Reject project |
| • Assign moderator to project | • View presentation schedule | • View profile info      |
| • View list of supervisors    | • View list of moderators    | • View list of students  |
| • Log-out                     |                              |                          |

The Administrator can:

- |                |                   |                 |
|----------------|-------------------|-----------------|
| • Log-in       | • Create account  | • View accounts |
| • Edit account | • Delete accounts |                 |

## ***3.2 System minimum requirements***

### ***3.2.1 Android***

- Android 5.1.1 (Lollipop) or higher
- CPU: Intel Dual-Core
- RAM: 1GB
- Storage: 100MB
- Internet access

### ***3.2.2 iOS***

- iOS 9.0 or higher
- CPU: Armv8 Dual-Core
- RAM: 1GB
- Storage: 100MB
- Internet access

### 3.3 Use Case Diagram

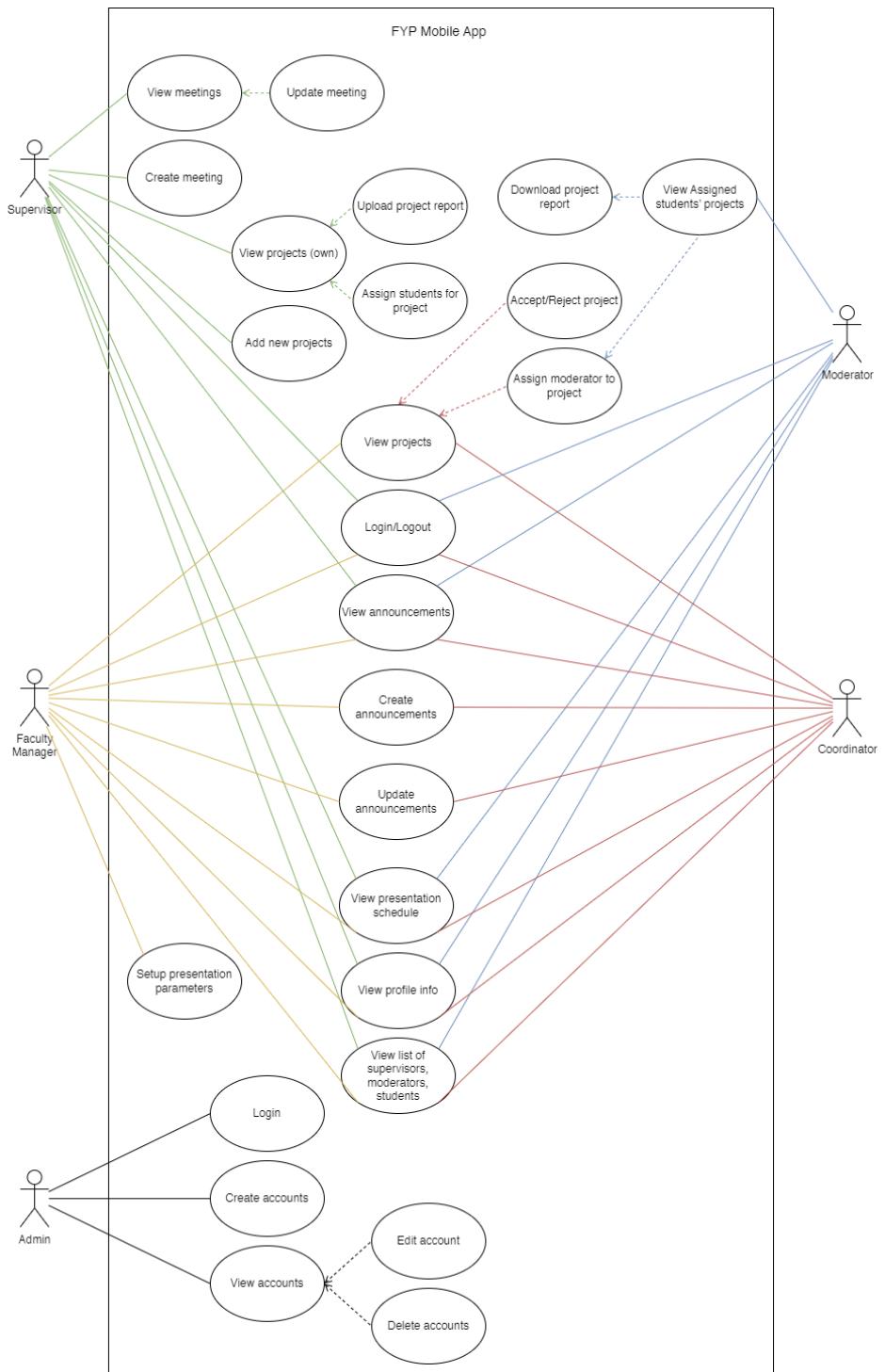


Figure 3.2 Use Case Diagram describing the FYP Mobile Application

In figure 3.2, we have five (5) actors and twenty-five (25) use cases. The five actors in this use case diagram are the supervisor, faculty manager, moderator, coordinator, and administrator.

### **3.3.1 Supervisor Modules**



Figure 3.3 Use Case Diagram for the actor Supervisor

The figure 3.3 shows the use case diagram of Lecturer/Supervisor. It consists of twelve(12) use cases, which each will be explained in detail by the tables 3.1 to 3.12.

**Table 3.1 Supervisor - Login/Logout Module**

|                |   |               |                     |
|----------------|---|---------------|---------------------|
| Use Case ID    | ALUC-01   | Use Case Name | Login/Logout Module |
| Pre-Condition  | User must be registered by the Administrator  |               |                     |
| Post-Condition | Valid login or invalid login   Logout   |               |                     |
| Scenario:      | 1) User inputs email and password.<br>2) If successful, the user is redirected to the home screen, the “Announcement” screen.<br>3) If unsuccessful, “Invalid email and/or password” alert will appear.<br>4) To logout, the user must navigate to the “Records” screen.<br>5) If the user is logged out, the user is set to null, thus ending the session. |               |                     |
| Extends List:  | None  |               |                     |

**Table 3.2 Supervisor - View Announcement Module**

|                |   |               |                          |
|----------------|---|---------------|--------------------------|
| Use Case ID    | ALUC-02   | Use Case Name | View Announcement Module |
| Pre-Condition  | User must be logged in  |               |                          |
| Post-Condition | Display announcement  |               |                          |
| Scenario:      | 1) Upon successful login, the user is redirected to the “Announcement” screen, where the announcement posted is displayed.<br>2) The user can also click the “Announcement” button at the bottom navigation bar or the “MMU” logo on top of the screen to open this page. |               |                          |
| Extends List:  | None  |               |                          |

**Table 3.3 Supervisor - Add New Project Module**

|                |   |               |                        |
|----------------|---|---------------|------------------------|
| Use Case ID    | SUUC-01   | Use Case Name | Add New Project Module |
| Pre-Condition  | User must be logged in as Supervisor  |               |                        |
| Post-Condition | Projects will successfully be proposed  |               |                        |
| Scenario:      | 1) Upon successful login as Supervisor, the user must click the “Project” button on the bottom navigation bar.<br>2) This will display the Projects screen where the user can see the status of their proposed project.<br>3) To propose a project, the user must click the “+” button located at the bottom right of the Projects screen.<br>4) After clicking the “+” button, this will redirect to the “Propose a Project” screen where the user must input:<br>a) Project Title<br>b) Project Type <ul style="list-style-type: none"> <li>i) Research based</li> <li>ii) Application based</li> </ul> c) Specialization <ul style="list-style-type: none"> <li>i) Information System</li> <li>ii) Software Engineering</li> </ul> |               |                        |

|               |   |
|---------------|---|
|               | <ul style="list-style-type: none"> <li>iii) Data Science</li> <li>iv) Game Development</li> <li>d) Focus</li> <li>e) Project Description</li> <li>f) Project Objectives</li> <li>g) Project Scope</li> <li>h) No. of Students           <ul style="list-style-type: none"> <li>i) 1</li> <li>ii) 2</li> </ul> </li> <li>i) *If 2 students:           <ul style="list-style-type: none"> <li>i) Student 1 Subtitle</li> <li>ii) Student 1 Work Distribution</li> <li>iii) Student 2 Subtitle</li> <li>iv) Student 2 Work Distribution</li> </ul> </li> <li>j) Industry Collaboration           <ul style="list-style-type: none"> <li>i) Yes</li> <li>ii) No</li> </ul> </li> <li>k) If “Yes” Industry Collaboration           <ul style="list-style-type: none"> <li>i) Company</li> <li>ii) Contact Name</li> <li>iii) Contact No.</li> </ul> </li> <li>5) After inputting, the user must click submit to submit the proposal. If there’s missing input, an alert will appear.</li> <li>6) If successful, a successful alert will appear and the proposal will be sent to the Coordinator to be reviewed.</li> </ul> |
| Extends List: | None  |

Table 3.4 Supervisor - View Projects (own) Module

| Use Case ID    | SUUC-02   | Use Case Name | View Projects (own) Module |
|----------------|---|---------------|----------------------------|
| Pre-Condition  | User must be logged in as Supervisor  |               |                            |
| Post-Condition | Show own proposed project's list  |               |                            |
| Scenario:      | 1) Upon successful login as Supervisor, the user must click the “Project” button on the bottom navigation bar.<br>2) This will display the Projects screen where the user can see the status of their proposed project: <ul style="list-style-type: none"> <li>a) Under Review</li> <li>b) Accepted</li> <li>c) Rejected</li> </ul> 3) Page will be blank if there is no project proposed/created.<br>4) The user can click/select the project to redirect to the “Project Info” screen, where the selected project's information will be displayed |               |                            |
| Extends List:  | None  |               |                            |

Table 3.5 Supervisor - Assign Students to Project Module

|                |  |               |                                   |
|----------------|--|---------------|-----------------------------------|
| Use Case ID    | SUUC-03  | Use Case Name | Assign Students to Project Module |
| Pre-Condition  | User must be logged in as Supervisor<br>- Project must be approved by Coordinator  |               |                                   |
| Post-Condition | Projects successfully assigned to student  |               |                                   |
| Scenario:      | 1) Once in the “Project Info” screen, the user can click on the “Assign Student” button.<br>2) User must then input the:<br>a) Student Name<br>b) Student ID<br>c) Student Contact No.<br>3) If 2 students project:<br>a) Student 1 Name<br>b) Student 1 ID<br>c) Student 1 Contact No.<br>d) Student 2 Name<br>e) Student 2 ID<br>f) Student 2 Contact No.<br>4) After inputting, the user must click the “Submit” button to assign the student to the project. If there’s missing input, an alert will appear.<br>5) If successful, a successful alert will appear and the student is assigned to the project. |               |                                   |
| Extends List:  | View projects (own) Module   |               |                                   |

Table 3.6 Supervisor - Upload Project Report Module

|                |  |               |                              |
|----------------|--|---------------|------------------------------|
| Use Case ID    | SUUC-04  | Use Case Name | Upload Project Report Module |
| Pre-Condition  | User must be logged in as Supervisor<br>- Project must be approved by Coordinator<br>- Student must be assigned to the project   |               |                              |
| Post-Condition | Project report will be uploaded  |               |                              |
| Scenario:      | 1) Once in the “Project Info” screen, the user can click on the “Upload Report” button.<br>2) If there are two students assigned to the project, two buttons will appear instead of one.<br>3) The user can then select a file to upload.<br>4) After the file is selected, the file will be uploaded. If there’s something wrong, an alert will appear.<br>5) If successful, a successful alert will appear and the file is uploaded. |               |                              |
| Extends List:  | View projects (own) Module   |               |                              |

**Table 3.7 Supervisor - Create Meeting Module**

|                |   |               |                       |
|----------------|---|---------------|-----------------------|
| Use Case ID    | SUUC-05   | Use Case Name | Create Meeting Module |
| Pre-Condition  | User must be logged in as Supervisor  |               |                       |
| Post-Condition | Meeting successfully created  |               |                       |
| Scenario:      | 1) Upon successful login as Supervisor, the user must click the “Meeting” button on the bottom navigation bar.<br>2) This will display the “Meetings” screen where the user can see the list of meetings created (page is blank if there’s no meeting created).<br>3) To create a meeting, the user must click the “+” button located at the bottom right of the Meetings screen.<br>4) After clicking the “+” button, this will redirect to the “Create a Meeting” screen where the user must input:<br>a) Project (approved project)<br>b) Student (assigned student)<br>c) Meeting Date<br>d) Meeting Time<br>5) If there’s missing input, an alert will appear.<br>6) If successful, a successful alert will appear and a meeting will be created.<br>7) A list of fourteen meetings will be created. |               |                       |
| Extends List:  | None  |               |                       |

**Table 3.8 Supervisor - View Meetings Module**

|                |   |               |                      |
|----------------|---|---------------|----------------------|
| Use Case ID    | SUUC-06   | Use Case Name | View Meetings Module |
| Pre-Condition  | User must be logged in as Supervisor  |               |                      |
| Post-Condition | Able to view created meetings   |               |                      |
| Scenario:      | 1) Upon successful login as Supervisor, the user must click the “Meeting” button on the bottom navigation bar.<br>2) This will display the Meetings screen where the user can see the meetings created (page is blank if there’s no meeting created). |               |                      |
| Extends List:  | None  |               |                      |

**Table 3.9 Supervisor - Update Meeting Module**

|                |  |               |                       |
|----------------|--|---------------|-----------------------|
| Use Case ID    | SUUC-07  | Use Case Name | Update Meeting Module |
| Pre-Condition  | Meeting must be created  |               |                       |
| Post-Condition | Meeting successfully updated   |               |                       |
| Scenario:      | 1) Upon successful meeting creation, the user must click the “Meeting” button on the bottom navigation bar.<br>2) This will display the “Meetings” screen where the user can see the list of meetings created (page is blank if there’s no meeting created).<br>3) Click on a meeting that’s been created (Will display Student Name).<br>4) A list of 14 weeks will be shown. |               |                       |

|               |   |
|---------------|---|
|               | <p>5) Choose a week and the user can update the meeting for the week by inputting:</p> <ul style="list-style-type: none"> <li>a) Meeting Mode</li> <li>b) Work Done</li> <li>c) Work to be Done</li> <li>d) Comments</li> </ul> <p>6) After inputting, the user can click the “Save” button.</p> <p>7) If successful, an alert will appear and the meeting details are saved and updated.</p> |
| Extends List: | View Meeting Module   |

Table 3.10 Supervisor - View Presentation Schedule Module

| Use Case ID    | ALUC-03   | Use Case Name | View Presentation Schedule Module |
|----------------|---|---------------|-----------------------------------|
| Pre-Condition  | User must be logged in.   |               |                                   |
| Post-Condition | Able to view presentation slots   |               |                                   |
| Scenario:      | <p>1) Upon successful login, the user must click the “Presentation” button on the bottom navigation bar.</p> <p>2) This will display the “Presentation” screen where the user can see the presentations created (page will be blank if presentations are not yet set by the Faculty Manager).</p> |               |                                   |
| Extends List:  | None  |               |                                   |

Table 3.11 Supervisor - View Profile Info Module

| Use Case ID    | ALUC-04  | Use Case Name | View Profile Info Module |
|----------------|--|---------------|--------------------------|
| Pre-Condition  | User must be logged in   |               |                          |
| Post-Condition | Able to view profile information   |               |                          |
| Scenario:      | <p>1) Upon successful login, the user must click the “Records” button on the bottom navigation bar.</p> <p>2) There will be a selectable list of buttons:</p> <ul style="list-style-type: none"> <li>a) List of Supervisors</li> <li>b) List of Moderators</li> <li>c) List of Students</li> </ul> <p>3) Depending on what is chosen, the user will be redirected to a list of Supervisors, Moderators, or Students that are registered.</p> |               |                          |
| Extends List:  | None   |               |                          |

Table 3.12 Supervisor - View List of Students/Supervisors/Moderators Module

|                |  |               |   |
|----------------|--|---------------|---|
| Use Case ID    | ALUC-05  | Use Case Name | View List of Students/Supervisors/Moderators Module |
| Pre-Condition  | User must be logged in   |               |   |
| Post-Condition | Able to view list of supervisors, moderators, and students   |               |   |
| Scenario:      | 1) Upon successful login, the user must click the “Records” button on the bottom navigation bar.<br>2) This will display the Records screen where profile information is displayed:<br>a) Avatar<br>b) Name<br>c) Email<br>d) Role |               |   |
| Extends List:  | None   |               |   |

### 3.3.2 Faculty Manager Modules



Figure 3.4 Use Case Diagram for the actor Faculty Manager

The figure 3.4 shows the use case diagram of the Faculty Manager. It consists of nine (9) use cases, which each will be explained in detail by the tables 3.13 to 3.21. Refer to Appendix E (E - 3.2.2) for the explanations on tables 3.13, 3.14, 3.17, 3.18, 3.20, and 3.21

Table 3.15 Faculty Manager - Create Announcements Module

| Use Case ID    | FCUC-01  | Use Case Name | Create Announcements Module |
|----------------|--|---------------|-----------------------------|
| Pre-Condition  | User must be logged in as Faculty Manager or Coordinator           |               |                             |
| Post-Condition | Announcement successfully created                                  |               |                             |
| Scenario:      | 1) Upon successful login as Faculty Manager, on the "Announcement" |               |                             |

|               |   |
|---------------|---|
|               | <p>screen, the user can click the “+” button located at the bottom right of the Announcement screen.</p> <p>2) After clicking the “+” button, this will redirect to the “Create Announcement” screen where the user must input:</p> <ul style="list-style-type: none"> <li>a) Title</li> <li>b) Description</li> <li>c) Attachment (if any)</li> </ul> <p>3) After inputting, the user must click submit to submit the announcement. If there's missing input, an alert will appear.</p> <p>4) If successful, a successful alert will appear and the announcement will be created.</p> <p>5) Announcements created will be displayed on the “Announcement” Screen. (latest post will be on the topmost)</p> |
| Extends List: | None  |

Table 3.16 Faculty Manager - Update Announcements Module

| Use Case ID    | FCUC-02   | Use Case Name | Update Announcements Module |
|----------------|---|---------------|-----------------------------|
| Pre-Condition  | Announcements must be created by the user   |               |                             |
| Post-Condition | Announcement successfully updated   |               |                             |
| Scenario:      | <p>1) Upon successful login as Faculty Manager, on the “Announcement” screen, the user can click the “...” button located at the top right of any of the Announcements created.</p> <p>2) The User can then pin the announcement, or delete it.</p> |               |                             |
| Extends List:  | None  |               |                             |

Table 3.19 Faculty Manager - Setup Presentation Parameters Module

| Use Case ID    | FMUC-01   | Use Case Name | Setup Presentation Parameters Module |
|----------------|---|---------------|--------------------------------------|
| Pre-Condition  | User must be logged in as Supervisor  |               |                                      |
| Post-Condition | Successfully setup presentation parameters (date,time,venue)  |               |                                      |
| Scenario:      | <p>1) Upon successful login as Faculty Manager, user must click the “Presentation” button on the bottom navigation bar, user can then click the “+” button located at the bottom right of the Presentation screen.</p> <p>2) After clicking the “+” button, this will redirect to the “Create Presentation” screen where the user must input:</p> <ul style="list-style-type: none"> <li>a) Presentation Type <ul style="list-style-type: none"> <li>i) Online</li> <li>ii) Physical</li> <li>iii) Poster</li> </ul> </li> <li>b) For Specialization <ul style="list-style-type: none"> <li>i) Any</li> <li>ii) Information System</li> <li>iii) Software Engineering</li> <li>iv) Data Science</li> <li>v) Game Development</li> </ul> </li> <li>c) Presentation Venue (if any)</li> <li>d) Presentation Date</li> </ul> |               |                                      |

|  |   |
|--|---|
|  | <p>e) Presentation Time</p> <p>3) After inputting, the user must click submit to “Create Presentation”. If there’s missing input, an alert will appear.</p> <p>4) If successful, a successful alert will appear and the presentation parameters will be available at the presentation screen.</p> |
|--|---|

### 3.3.3 Moderator Modules

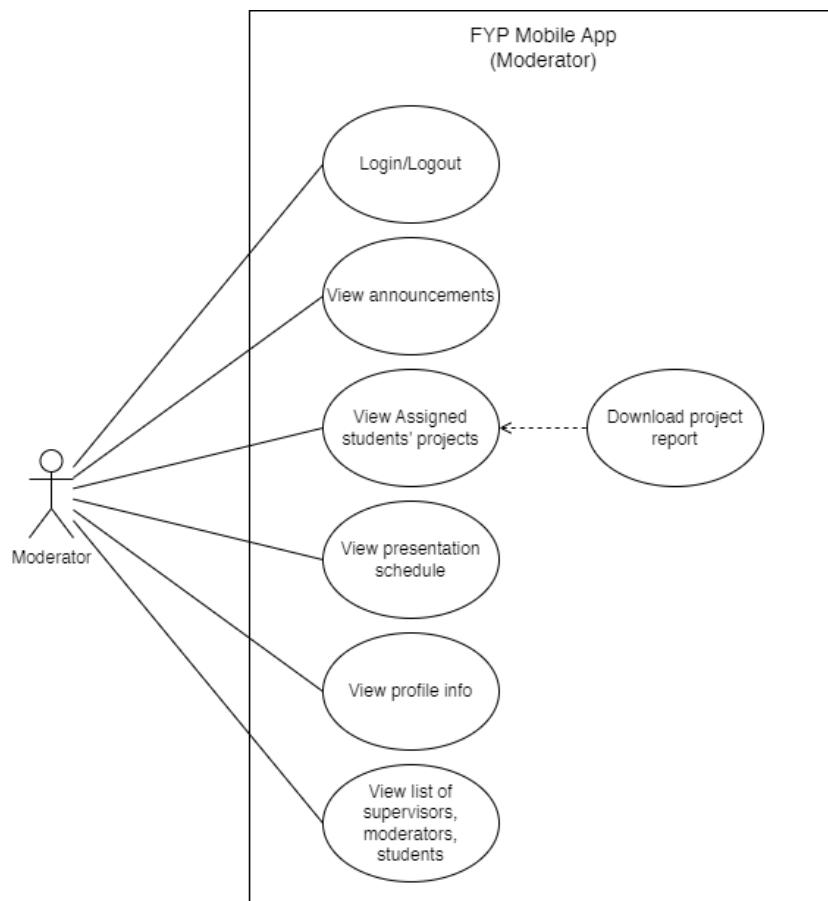


Figure 3.5 Use Case Diagram for the actor Moderator

The figure 3.5 shows the use case diagram of the Moderator. It consists of seven (7) use cases, which each will be explained in detail by the tables 3.22 to 3.28. Refer to Appendix E (E - 3.3.3) for the explanations on tables 3.22, 3.23, 3.26, 3.27, and 3.28

**Table 3.24 Moderator - View Assigned Students' Project Module**

|                |  |               |  |
|----------------|--|---------------|--|
| Use Case ID    | MOUC-01  | Use Case Name | View Assigned Students' Project Module |
| Pre-Condition  | User must be logged in as Moderator  |               |  |
| Post-Condition | Able to view projects (assigned by Coordinator)  |               |  |
| Scenario:      | 1) Upon successful login as Moderator, the user must click the “Project” button on the bottom navigation bar.<br>2) This will display the Projects screen where the user can see the assigned projects.<br>3) Page will be blank if there is no assignment of projects.<br>4) The user can click/select the project to redirect to the “Project Info” screen, where the selected project’s information will be displayed |               |  |
| Extends List:  | None   |               |  |

**Table 3.25 Moderator - Download Project Report Module**

|                |   |               |                                |
|----------------|---|---------------|--------------------------------|
| Use Case ID    | MOUC-02   | Use Case Name | Download Project Report Module |
| Pre-Condition  | User must be logged in as Moderator<br>- Project must be assigned to user by Coordinator<br>- Student must be assigned to the project<br>- Project report must be uploaded by Supervisor  |               |                                |
| Post-Condition | Project report will be downloaded   |               |                                |
| Scenario:      | 1) Once in the “Project Info” screen, the user can click on the “Download Report” button.<br>2) If there are two students assigned to the project, two buttons will appear instead of one.<br>3) The user can then download the file.<br>4) If there’s something wrong, an alert will appear.<br>5) If successful, a successful alert will appear and the file is downloaded. |               |                                |
| Extends List:  | View Assigned Student's Project Module  |               |                                |

### 3.3.4 Coordinator Modules

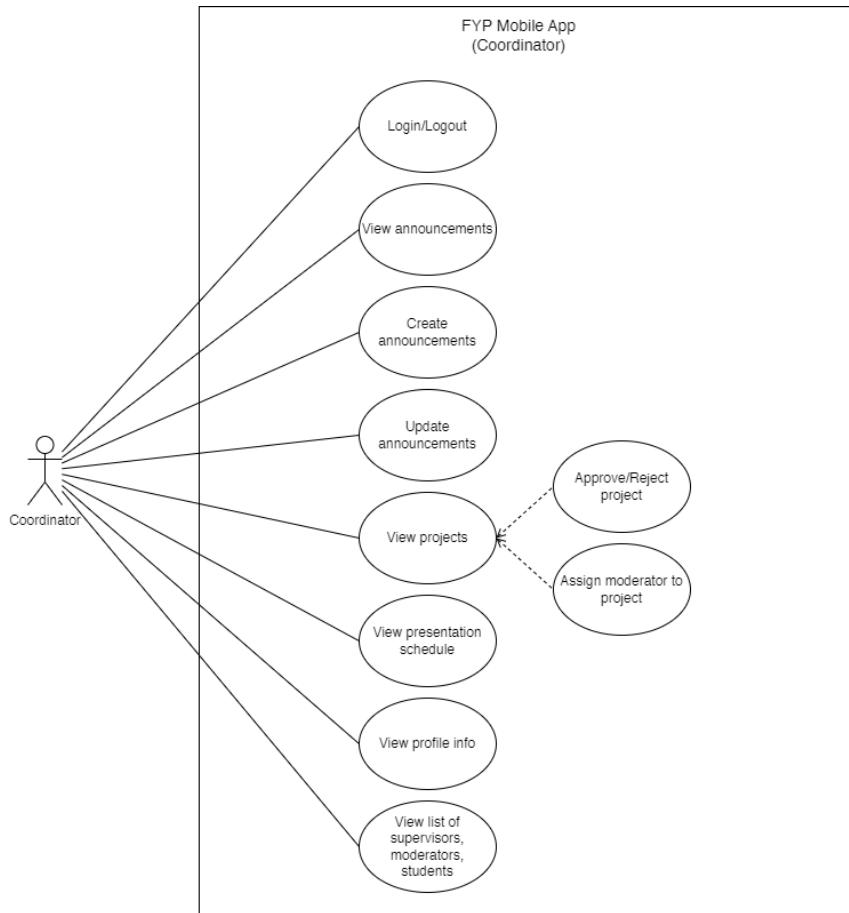


Figure 3.6 Use Case Diagram for the actor Coordinator

The figure 3.6 shows the use case diagram of the Coordinator. It consists of ten (10) use cases, which each will be explained in detail by the tables 3.29 to 3.38. Refer to Appendix E (E - 3.3.4) for the explanations on tables 3.29, 3.30, 3.31, 3.32, 3.33, 3.36, 3.37, and 3.38.

Table 3.34 Coordinator - Approve/Reject Project Module

| Use Case ID    | COUC-01   | Use Case Name | Approve/Reject Project Module |
|----------------|---|---------------|-------------------------------|
| Pre-Condition  | User must be logged in as Coordinator<br>- Project status must be “pending” |               |                               |
| Post-Condition | Able to approve or reject a project   |               |                               |
| Scenario:      | 1) Once in the “Project Info” screen, the user can click on the “Approve”   |               |                               |

|               |  |
|---------------|--|
|               | <p>button to approve the project or click on the “Reject” button to reject the project.</p> <ol style="list-style-type: none"> <li>2) If it is approved, an alert will appear and project status will be set to “approved”.</li> <li>3) If it is rejected, an alert will appear and project status will be set to “rejected”.</li> </ol> |
| Extends List: | View Projects Module   |

Table 3.35 Coordinator - Assign Moderator to Project Module

| Use Case ID    | COUC-02   | Use Case Name | Assign Moderator to Project Module |
|----------------|---|---------------|------------------------------------|
| Pre-Condition  | User must be logged in as Coordinator<br>- Student must be assigned to the project (by Supervisor)  |               |                                    |
| Post-Condition | Successfully assign a moderator to a student  |               |                                    |
| Scenario:      | <ol style="list-style-type: none"> <li>1) Once in the “Project Info” screen, the user can click on the “Assign Moderator” button.</li> <li>2) User must then input:           <ol style="list-style-type: none"> <li>a) Moderator</li> </ol> </li> <li>3) After inputting, the user must click the “Submit” button to assign the moderator to the project. If there’s missing input, an alert will appear.</li> <li>4) If successful, a successful alert will appear and the moderator is assigned to the project.</li> </ol> |               |                                    |
| Extends List:  | View Projects Module  |               |                                    |

### 3.3.4 Administrator Modules

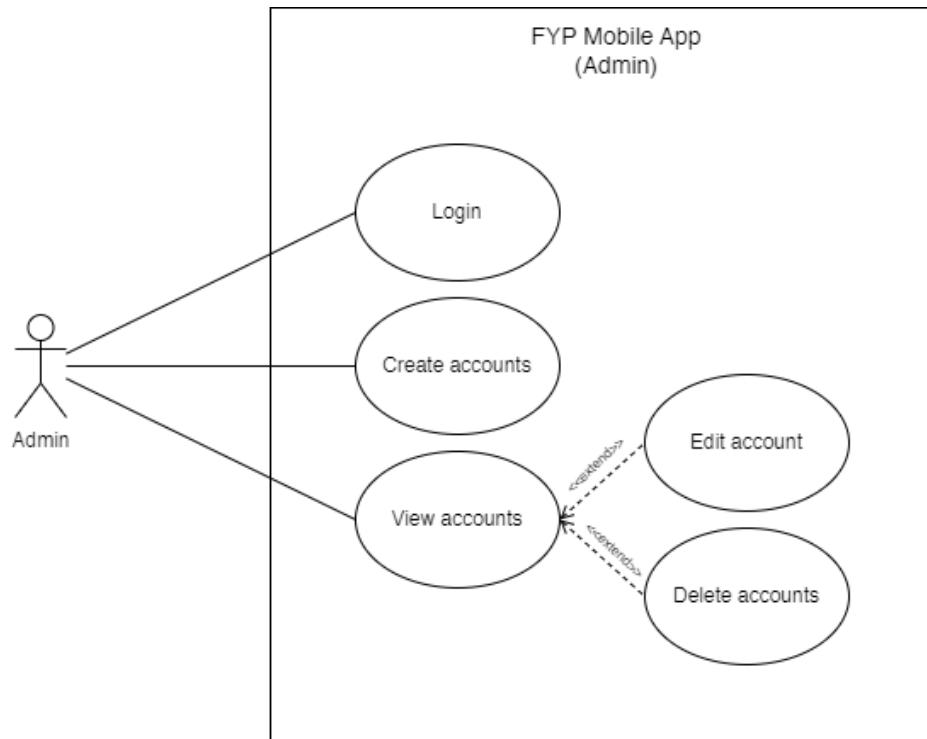


Figure 3.7 Use Case Diagram for the actor Administrator

The figure 3.7 shows the use case diagram of the Administrator. It consists of five (5) use cases, which each will be explained in detail by the tables 3.39 to 3.43.

Table 3.39 Admin - Login Module

| Use Case ID    | ADUC-01  | Use Case Name | Login Module |
|----------------|--|---------------|--------------|
| Pre-Condition  | None   |               |              |
| Post-Condition | Valid login or invalid login   |               |              |
| Scenario:      | 1) User inputs email and password.<br>2) If successful, the user is sent to the admin dashboard.<br>3) If unsuccessful, “Invalid email and/or password” message will appear. |               |              |
| Extends List:  | None   |               |              |

**Table 3.40 Admin - Create Accounts Module**

|                |   |               |                        |
|----------------|---|---------------|------------------------|
| Use Case ID    | ADUC-02   | Use Case Name | Create Accounts Module |
| Pre-Condition  | User must be logged in as Administrator   |               |                        |
| Post-Condition | Successfully created an account (Supervisor, Faculty Manager, Coordinator, or Moderator)  |               |                        |
| Scenario:      | 1) Upon successful login as administrator, user can select the “Create an Account”<br>2) After selecting, user will need to input:<br>a) Account Type:<br>i) Supervisor<br>ii) Faculty Manager<br>iii) Coordinator<br>iv) Moderator<br>b) Name<br>c) Staff ID<br>d) Staff Email<br>e) Phone No.<br>f) Address<br>g) Password<br>h) Confirm Password<br>3) After inputting, the user must click submit to submit the account creation. If there's missing input, a message will display “Error! Missing [input]”.<br>4) If successful, a message will display “Account Created!” and the account will be created successfully. |               |                        |
| Extends List:  | None  |               |                        |

**Table 3.41 Admin - View Accounts Module**

|                |  |               |                      |
|----------------|--|---------------|----------------------|
| Use Case ID    | ADUC-03  | Use Case Name | View Accounts Module |
| Pre-Condition  | User must be logged in as Administrator  |               |                      |
| Post-Condition | Able to view all the accounts  |               |                      |
| Scenario:      | 1) Upon successful login as administrator, user can select the “Account”<br>2) A list of accounts created will appear. |               |                      |
| Extends List:  | None   |               |                      |

**Table 3.42 Admin - Edit Accounts Module**

|                |  |               |                      |
|----------------|--|---------------|----------------------|
| Use Case ID    | ADUC-04  | Use Case Name | Edit Accounts Module |
| Pre-Condition  | Account must be created  |               |                      |
| Post-Condition | Able to edit accounts  |               |                      |
| Scenario:      | 1) Upon successful login as administrator, user can select the “Account”<br>2) A list of accounts created will appear.<br>3) User can select an account to edit. |               |                      |

|               |                      |
|---------------|----------------------|
| Extends List: | View Accounts Module |
|---------------|----------------------|

Table 3.43 Admin - Delete Accounts Module

| Use Case ID    | ADUC-05  | Use Case Name | Delete Accounts Module |
|----------------|--|---------------|------------------------|
| Pre-Condition  | Account must be created  |               |                        |
| Post-Condition | Able to delete accounts  |               |                        |
| Scenario:      | 1) Upon successful login as administrator, user can select the “Account”<br>2) A list of accounts created will appear.<br>3) User can select an account to delete. |               |                        |
| Extends List:  | View Accounts Module   |               |                        |

### 3.4 Class Diagram

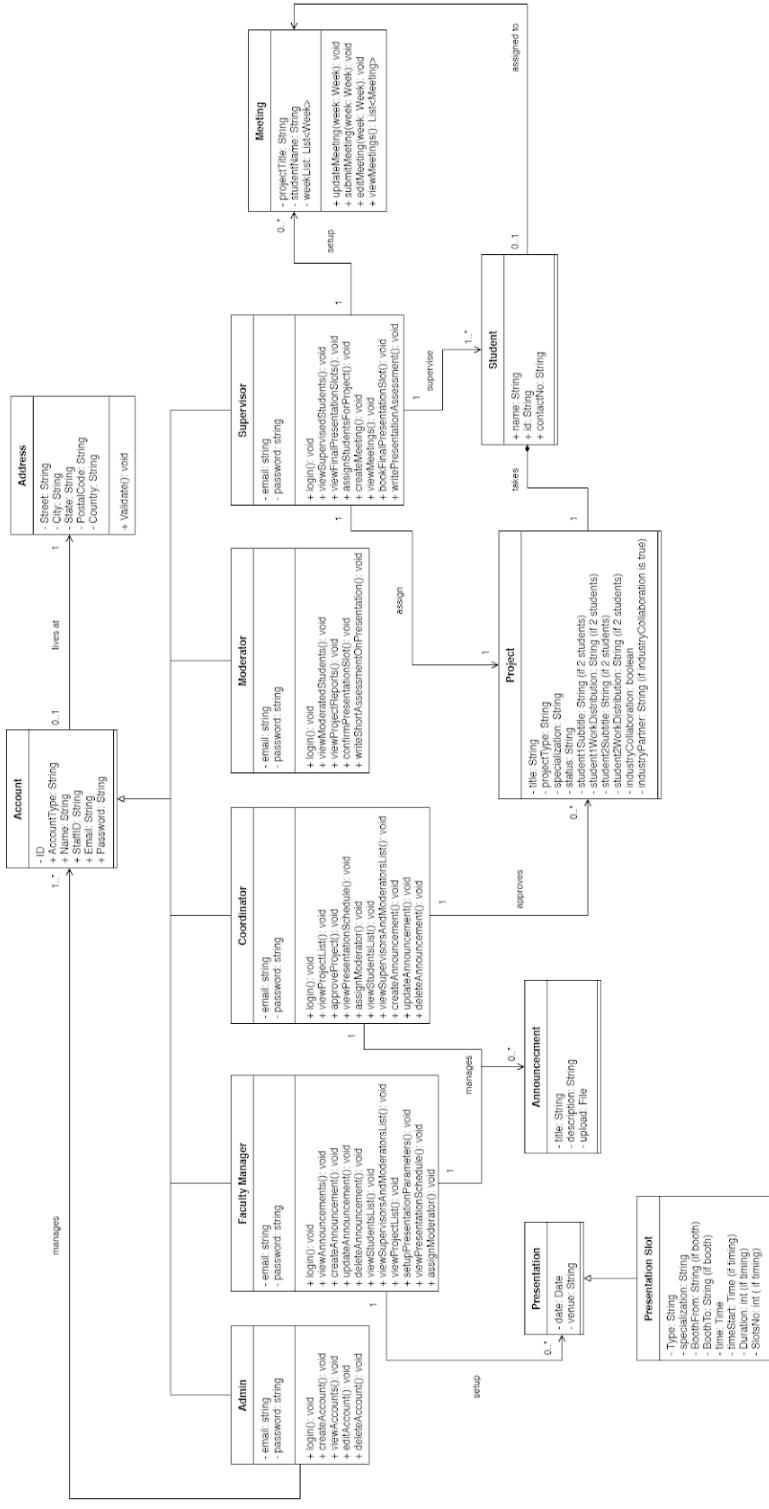


Figure 3.8 Class Diagram

Figure 3.8 shows the class diagram of FYP Mobile Application. The whole project consists of 13 classes in total. Five of them named Admin, Faculty Member, Coordinator, Moderator and Supervisor are inherited from the Account class. The Account class consists of several members including ID, Name, Phone, Email, Password and is associated with Address which contains the address of the account member. Additional information about the child members is explained below along with their functions:

#### ***3.4.1 Supervisor***

Supervisor is inheriting all the members of Account class. The Supervisor class represents a supervisor who has access to various modules in the system. The Supervisor class would contain attributes such as email and password for login, as well as a list of Student objects representing the students that the supervisor is supervising. The login method would allow the supervisor to log in to the system. The viewSupervisedStudents method would allow the supervisor to view a list of all students that they are supervising. The viewMeeting method would allow the supervisor to view a specific meeting for a given week, while the viewAllMeetings method would return a list of all meetings for the supervisor. The assignStudents method would allow the supervisor to assign a student to a specific project. The createMeeting method would allow the supervisor to create a new meeting for a specific week. The viewMeetings method would allow the supervisor to view a list of all meetings created. The writePresentationAssessment method would allow the supervisor to write an assessment for a specific presentation. The

`bookFinalPresentationSlot` method would allow the supervisor to book a final presentation slot for a specific presentation and timeslot.

### ***3.4.2 Faculty Manager***

The Faculty Manager is inheriting all the members of the Account class. The Faculty Manager class represents a faculty manager who has access to various modules in the system. The Faculty Manager class would contain attributes such as email and password for login. The login method would allow the faculty manager to log in to the system. The `createAnnouncement` method would allow the faculty manager to create a new announcement by providing a title, description, and optional file upload. The `updateAnnouncement` method would allow the faculty manager to update an existing announcement by providing a new title, description, and optional file upload. The `deleteAnnouncement` method would allow the faculty manager to delete an existing announcement. The `viewAnnouncements` method would allow the faculty manager to view a list of all announcements created. The `viewStudentsList` method would allow the faculty manager to view a list of all FYP students. The `viewSupervisorsAndModeratorsList` method would allow the faculty manager to view a list of all supervisors and moderators. The `viewProjectList` method would allow the faculty manager to view a list of all approved projects. The `setupPresentationParameters` method would allow the faculty manager to set up the presentation parameters such as date, time, and venue. The `viewPresentationSchedule` method would allow the faculty manager to view the presentation schedule. The `assignModerator` method would allow the faculty manager to assign a moderator to a specific student.

### ***3.4.3 Coordinator***

The Coordinator class represents a coordinator who has access to various modules in the system. The Coordinator class would contain attributes such as email and password for login. The login method would allow the coordinator to log in to the system. The viewProjectList method would allow the coordinator to view a list of all projects proposed by supervisors. The approveProject method would allow the coordinator to approve or reject a specific project by providing the project object and the status (Accept, Reject, Pending). The viewStudentsList method would allow the coordinator to view a list of all FYP students. The viewSupervisorsAndModeratorsList method would allow the coordinator to view a list of all supervisors and moderators. The viewPresentationSchedule method would allow the coordinator to view the presentation schedule. The assignModerator method would allow the coordinator to assign a moderator to a specific student.

### ***3.4.4 Moderator***

The Moderator class represents a moderator who has access to various modules in the system. The Moderator class would contain attributes such as email and password for login. The login method would allow the moderator to log in to the system. The viewModeratedStudents method would allow the moderator to view a list of all students that they are moderating. The viewProjectReports method would allow the moderator to view a list of project reports uploaded by the supervisor for a specific project. The confirmPresentationSlot method would allow the moderator to confirm a presentation slot that has been booked by the supervisor. The writeShortAssessment

method would allow the moderator to write a short assessment for a specific presentation slot.

### ***3.4.5 Admin***

Admin is inheriting all the members of Account class. A single admin manages 1 or more accounts. The Admin class represents an administrator who has access to various modules in the system. The Admin class would contain attributes such as email and password for login. The login method would allow the administrator to log in to the system. The createAccount method would allow the administrator to create a new account for a Supervisor, Faculty Manager, Coordinator, or Moderator by providing the necessary information such as account type, name, staff ID, staff email, phone number, address, password, and confirm password. The viewAccounts method would allow the administrator to view a list of all accounts created. The editAccount method would allow the administrator to edit an existing account. The deleteAccount method would allow the administrator to delete an existing account.

### 3.5 ER Diagram

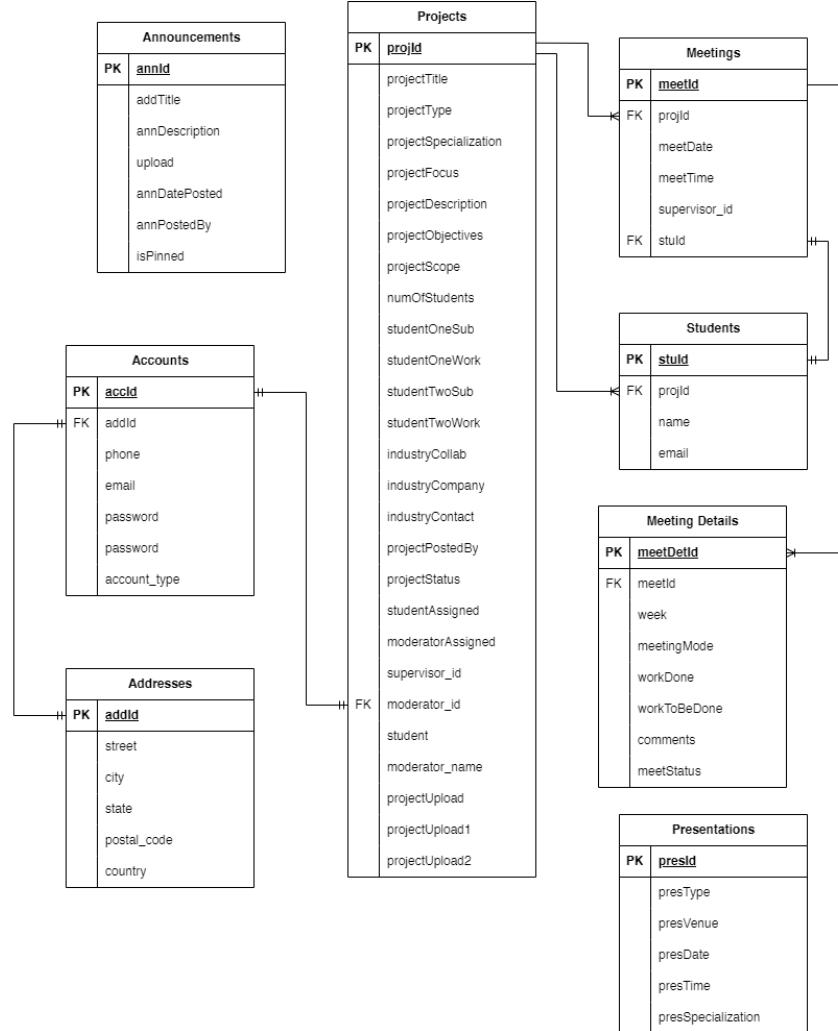


Figure 3.9 Entity Relationship Diagram

As shown in Figure 3.9, there are in total 8 entities in the Entity Relationship Diagram (ERD). An entity relationship diagram (ERD) is a graphical representation of the relationships between entities in a database. It is used to model the data requirements for a system and to illustrate the relationships between different entities in the system.

The Accounts entity has seven attributes. accId with integer data type and not null restriction is the primary key of the entity whereas addId with integer data type is a foreign key. This foreign key is used to reference the address of the account member which is stored in the address table. Additionally, it has name, phone, email, password, address and account\_type all of which are character data types. account\_type is an important attribute as it defines the role of the account either supervisor, coordinator, moderator, faculty member or administrator.

The Address entity has a total of six attributes describing an address. Address\_id with integer data type and not null restriction is the primary key. Additionally, it has street, city, state, postal\_code and country all of which are character except postal\_code which is an integer. The Addresses entity is referenced in Accounts.

The Students entity has four attributes. stuId with integer data type and not null restriction is the primary key of the entity whereas projId, and stuId with integer data type is a foreign key referencing Projects and Students entity respectively. Additionally, it has a name, and email, of which are character data types.

The Projects entity has twenty-seven attributes. projId with integer data type and not null restriction is the primary key of the entity whereas supervisor\_id, and moderator\_id with integer data type is a foreign key referencing Accounts entity. Additionally, it has projectTitle, projectType, projectDescription, projectStatus and much more, all of which are character data types. The projectStatus attribute shows the current status of the project which could be ‘pending’, ‘approved’, or ‘rejected’.

The Announcements entity has seven attributes. annId with integer data type and not null restriction is the primary key of the entity. Additionally, it has an annTitle,

annDescription, upload, annDatePosted, annPostedBy, and isPinned with character data type, jsonb for the upload, and announcement data with timestamp data type.

The Meetings entity has six attributes. meetId with integer data type and not null restriction is the primary key of the entity whereas projId, and stuId with integer data type is a foreign key referencing Projects and Students entity respectively. Additionally, it has meetDate and meetTime with a timestamp data type.

The Meeting Details entity has four attributes. meetDetId with integer data type and not null restriction is the primary key of the entity. Additionally, it has meetId with integer data type that is a foreign key referencing Meeting entity.

The Presentations entity has six attributes. presId with integer data type and not null restriction is the primary key of the entity whereas presType, presVenue, presDate, presTime, and presSpecialization are attributes.

# Chapter 4: DESIGN

## 4.1 Software Architecture

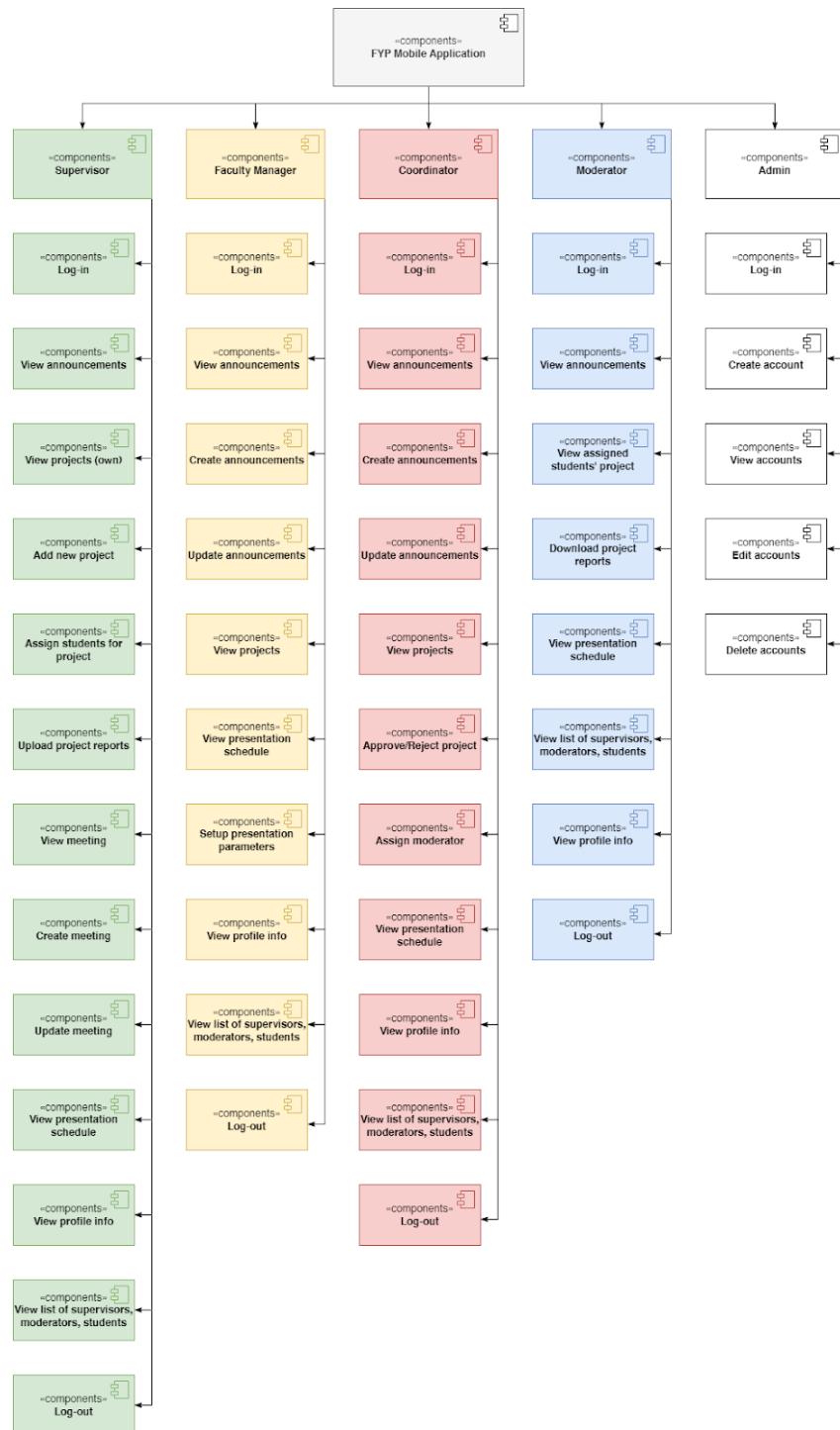


Figure 4.1 Software Architecture

Figure 4.1 shows the software architecture diagram for the FYP Mobile application.

This software architecture consists of five main components which are:

#### ***4.1.1 Supervisor***

The Supervisor component is a self-contained unit of functionality that is responsible for managing and supervising the activities of students in a system. It consists of the following sub-components:

- Login: This sub-component allows a supervisor to authenticate and access the system.
- View Meetings: This sub-component allows a supervisor to view a list of meetings that they are scheduled to attend.
- Add New Projects: This sub-component allows a supervisor to add new projects to the system.
- View Announcements: This sub-component allows a supervisor to view and stay updated on announcements and updates related to the system.
- View Projects: This sub-component allows a supervisor to view a list of projects in the system.
- Create Meeting: This sub-component allows a supervisor to schedule and create a new meeting.

Finally, the Supervisor component provides a range of functions and tools for managing and supervising the activities of students in the system. It is an essential component for ensuring the smooth operation and success of the system.

#### ***4.1.2 Faculty Manager***

The Faculty Manager component is a self-contained unit of functionality that is responsible for managing and coordinating the activities of students and other faculty members in a system. It consists of the following sub-components:

- Login: This sub-component allows a faculty member to authenticate and access the system.
- Setup Presentation Parameters: This sub-component allows a faculty member to specify the parameters and requirements for student presentations.
- Create Announcements: This sub-component allows a faculty member to create and publish announcements related to the system.
- View Student List: This sub-component allows a faculty member to view a list of students in the system.
- View Supervisor and Moderator List: This sub-component allows a faculty member to view a list of supervisors and moderators in the system.
- View Project List: This sub-component allows a faculty member to view a list of projects in the system.

Finally, the Faculty Member component provides a range of functions and tools for managing and coordinating the activities of students and other faculty members in the system. It is an essential component for ensuring the smooth operation and success of the system.

#### ***4.1.3 Coordinator***

The Coordinator component is a self-contained unit of functionality that is responsible for managing and coordinating the activities of students and other faculty members in a system. It consists of the following sub-components:

- Login: This sub-component allows a coordinator to authenticate and access the system.
- Assign Moderator: This sub-component allows a coordinator to assign a moderator to a student or a project.
- View Presentation Schedule: This sub-component allows a coordinator to view the schedule of presentations in the system.
- View Student List: This sub-component allows a coordinator to view a list of students in the system.
- View Supervisor and Moderator List: This sub-component allows a coordinator to view a list of supervisors and moderators in the system.
- View Project List: This sub-component allows a coordinator to view a list of projects in the system.
- Create Announcements: This sub-component allows a coordinator to create and publish announcements related to the system.
- View Announcements: This sub-component allows a coordinator to view and stay updated on announcements and updates related to the system.

Finally, the Coordinator component provides a range of functions and tools for managing and coordinating the activities of students and other faculty members in

the system. It is an essential component for ensuring the smooth operation and success of the system.

#### ***4.1.4 Moderator***

The Moderator component is a self-contained unit of functionality that is responsible for managing and moderating the activities of students in a system. It consists of the following sub-components:

- Login: This sub-component allows a moderator to authenticate and access the system.
- View Moderated Students: This sub-component allows a moderator to view a list of students that they are responsible for moderating.
- View Project Reports: This sub-component allows a moderator to view and review project reports submitted by students.
- View Announcements: This sub-component allows a moderator to view and stay updated on announcements and updates related to the system.

Finally, the Moderator component provides a range of functions and tools for managing and moderating the activities of students in the system. It is an essential component for ensuring the smooth operation and success of the system.

#### ***4.1.5 Admin***

The Admin component is a self-contained unit of functionality that is responsible for managing and maintaining user accounts in a system. It consists of the following sub-components:

- Login: This sub-component allows an administrator to authenticate and access the system.
- Create Accounts: This sub-component allows an administrator to create new user accounts in the system.
- View Accounts: This sub-component allows an administrator to view a list of all user accounts in the system.
- Edit Account: This sub-component allows an administrator to edit the details of an existing user account.
- Delete Account: This sub-component allows an administrator to delete a user account from the system.

Finally, the Admin component provides a range of functions and tools for managing and maintaining user accounts in the system. It is an essential component for ensuring the smooth operation and security of the system.

## 4.2 Software Model

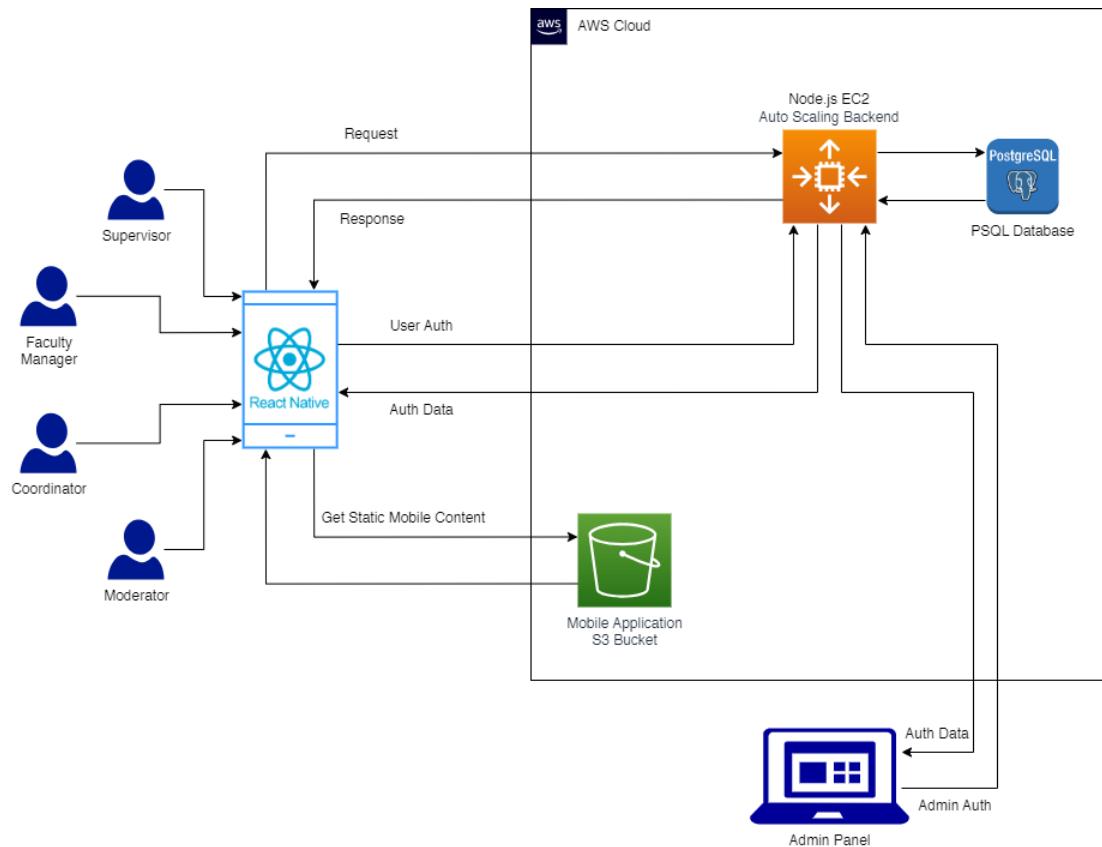


Figure 4.2 Software Model

Figure 4.2 shows the model diagram for the FYP Mobile application. This architecture consists of several components which are:

- **AWS S3 Bucket** – The S3 bucket is used to store static content for mobile users. The mobile users' S3 bucket stores content that is specific to the mobile application.
- **Node.js EC2 Backend** – The Node.js backend, which is hosted on an EC2 Auto Scaling instance, receives and responds to requests from React Native Mobile App. The backend performs various tasks such as processing data, or integrating with other services.

## 4.3 Sequence Diagrams

### 4.3.1 Login

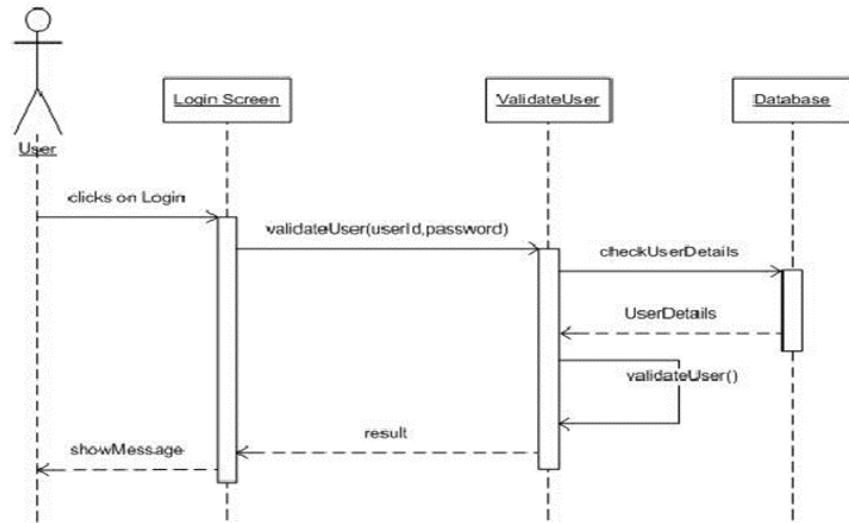


Figure 4.3 Login Sequence Diagram

In a login sequence diagram, the objects involved in the login process are represented as boxes or rectangles, and the interactions between these objects are represented by arrows or lines. These arrows represent the messages or method calls that are exchanged between the objects, and the sequence in which they are exchanged.

The login process typically involves the following steps:

- The user enters their login credentials (e.g. username and password) into a login form.
- The login form sends a request to the server to authenticate the user's credentials.
- The server checks the user's credentials against a database of registered users to verify their authenticity.

- If the credentials are valid, the server sends a message to the login form indicating that the user has been authenticated.
- The login form redirects the user to the main application or dashboard.
- The user is now logged into the system and can access the various features and functions available to them.

#### **4.3.2 View Announcement**

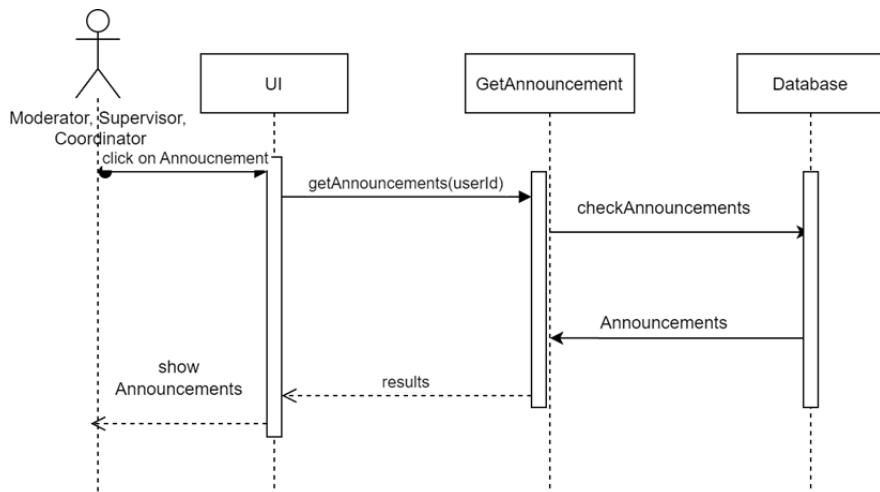


Figure 4.4 View Announcement Sequence Diagram

In a view announcement sequence diagram, the objects involved in the process are represented as boxes or rectangles, and the interactions between these objects are represented by arrows or lines. These arrows represent the messages or method calls that are exchanged between the objects, and the sequence in which they are exchanged.

The process of viewing an announcement typically involves the following steps:

- The user clicks on a link or button to view an announcement.

- The system sends a request to the server to retrieve the details of the selected announcement.
- The server retrieves the announcement details from a database or other data store.
- The server sends the announcement details to the system.
- The system displays the announcement to the user.

#### 4.3.3 Create Announcement

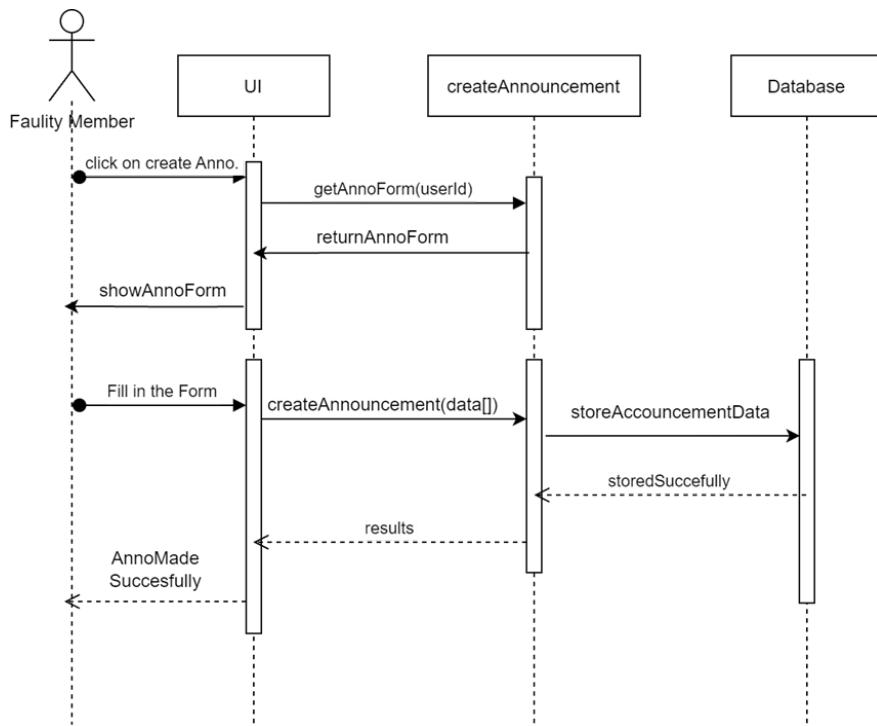


Figure 4.5 Create Announcement Sequence Diagram

In a create announcement sequence diagram, the objects involved in the process are represented as boxes or rectangles, and the interactions between these objects are represented by arrows or lines. These arrows represent the messages or method calls

that are exchanged between the objects, and the sequence in which they are exchanged.

The process of creating an announcement typically involves the following steps:

- The user enters the details of the announcement (e.g. title, content, target audience) into a form.
- The system sends a request to the server to create the new announcement.
- The server stores the announcement details in a database or other data store.
- The server sends a message to the system indicating that the announcement has been created successfully.
- The system displays a confirmation message to the user.

#### **4.3.4 View Presentation Schedule**

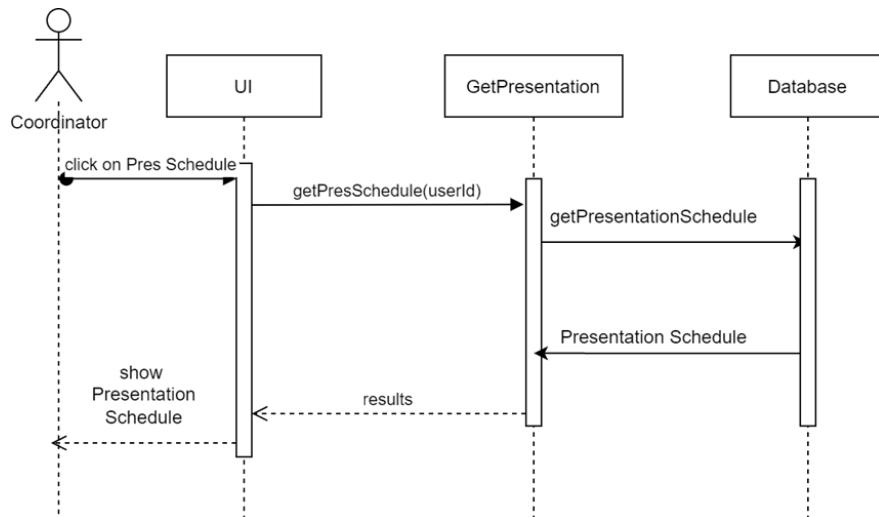


Figure 4.6 View Presentation Schedule Sequence Diagram

In a view presentation schedule sequence diagram, the objects involved in the process are represented as boxes or rectangles, and the interactions between these

objects are represented by arrows or lines. These arrows represent the messages or method calls that are exchanged between the objects, and the sequence in which they are exchanged.

The process of viewing the presentation schedule typically involves the following steps:

- The user clicks on a link or button to view the presentation schedule.
- The system sends a request to the server to retrieve the details of the presentations.
- The server retrieves the presentation details from a database or other data store.
- The server sends the presentation details to the system.
- The system displays the presentation schedule to the user.

#### 4.3.5 Add New Project

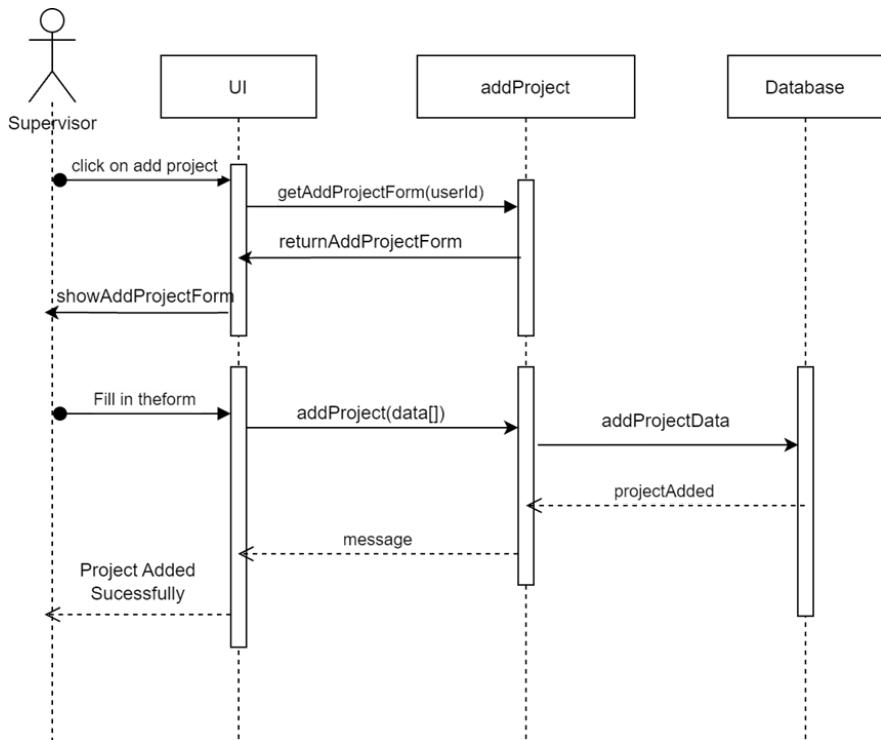


Figure 4.7 Add New Project

In add new project sequence diagram, the objects involved in the process are represented as boxes or rectangles, and the interactions between these objects are represented by arrows or lines. These arrows represent the messages or method calls that are exchanged between the objects, and the sequence in which they are exchanged.

The process of adding a new project typically involves the following steps:

- The user enters the details of the project (e.g. title, description, deadlines) into a form.
- The system sends a request to the server to create the new project.
- The server stores the project details in a database or other data store.

- The server sends a message to the system indicating that the project has been added successfully.
- The system displays a confirmation message to the user.

#### **4.3.6 Create Meeting**

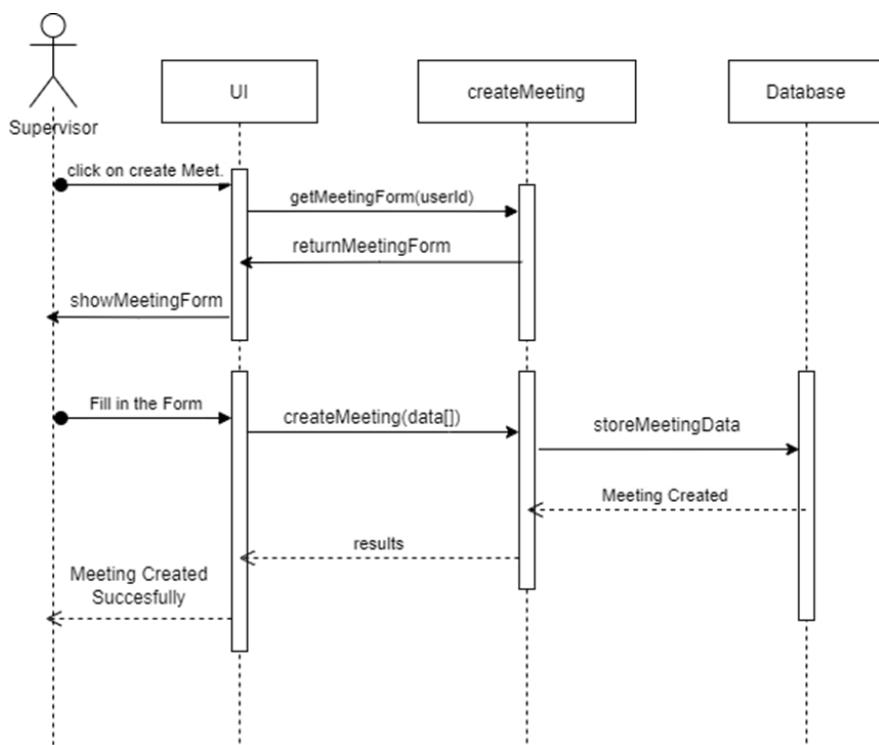


Figure 4.8 Create Meeting Sequence Diagram

In a create meeting sequence diagram, the objects involved in the process are represented as boxes or rectangles, and the interactions between these objects are represented by arrows or lines. These arrows represent the messages or method calls that are exchanged between the objects, and the sequence in which they are exchanged.

The process of creating a meeting typically involves the following steps:

- The user enters the details of the meeting (e.g. title, date, time, location) into a form.
- The system sends a request to the server to create the new meeting.
- The server stores the meeting details in a database or other data store.
- The server sends a message to the system indicating that the meeting has been created successfully.
- The system displays a confirmation message to the user.

#### 4.4 Screen Design

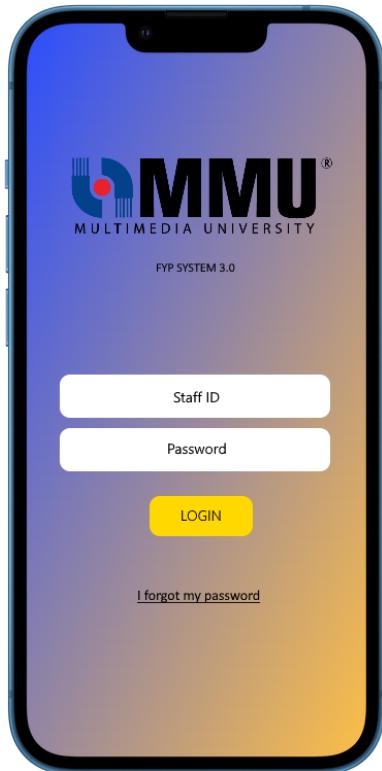


Figure 4.9 Login screen



Figure 4.10 Homepage/screen

## 4.5 Data Dictionary

Table 4.1 Data Dictionary

| Field Name             | Data Type | Description  | Example               | Data Format |
|------------------------|-----------|--|-----------------------|-------------|
| <b>Accounts Table</b>  |           |  |                       |             |
| acctId                 | int       | Unique identifier for each account                 | 1                     | Integer     |
| addId                  | int       | References the address associated with the account | 1                     | Integer     |
| name                   | varchar   | The name of the account holder                     | “John Doe”            | String      |
| phone                  | varchar   | Phone number of the account holder                 | “+60 123456789”       | String      |
| email                  | varchar   | Email address of the account holder                | “johndoe@example.com” | String      |
| password               | varchar   | Encrypted password of the account holder           | “mypassword”          | String      |
| account_type           | varchar   | The role of the account holder                     | “Supervisor”          | String      |
| <b>Addresses Table</b> |           |  |                       |             |
| addId                  | int       | Unique identifier for each address                 | 1                     | Integer     |
| street                 | varchar   | Street of the address                              | “123 Main Street”     | String      |
| city                   | varchar   | City of the address                                | “New York City”       | String      |
| state                  | varchar   | State of the address                               | “New York”            | String      |
| postal_code            | int       | Postal code of the address                         | 12345                 | Integer     |

|                            |             |   |   |           |
|----------------------------|-------------|---|---|-----------|
| country                    | varchar(50) | Country of the address                  | "United States"   | String    |
| <b>Announcements Table</b> |             |   |   |           |
| annId                      | int         | Unique identifier for each announcement | 1   | Integer   |
| annTitle                   | varchar     | Title of the announcement               | "Important Notice: Upcoming Maintenance"                                      | String    |
| annDescription             | varchar     | Description of the announcement         | "Please be informed that there will be scheduled maintenance on our website." | String    |
| upload                     | jsonb       | Attachment of the announcement          | {"file_name": "test.pdf", "file_size": "2.5 MB"}                              | Object    |
| annDatePosted              | time        | Date posted of the announcement         | "2023-06-20 09:00:00"   | Timestamp |
| annPostedBy                | text        | Name of the poster of the announcement  | "John Doe"  | String    |
| isPinned                   | varchar     | Pinned status of the announcement       | "Yes"   | String    |
| <b>Projects Table</b>      |             |   |   |           |
| projId                     | int         | Unique identifier of each project       | 1   | Integer   |
| projectTitle               | varchar     | Title of the project                    | "Website Redesign"  | String    |
| projectType                | varchar     | Type of the project                     | "Application Based"   | String    |
| projectSpecialization      | varchar     | Specialization of the project           | "Information System"  | String    |
| projectFocus               | varchar     | Focus of the project                    | "User Experience"   | String    |

|                    |         |   |  |         |
|--------------------|---------|---|--|---------|
| projectDescription | varchar | Description of the project                      | "The project aims to redesign the company website to improve user experience and enhance the visual appeal." | String  |
| projectObjectives  | varchar | Objectives of the project                       | "1. Improve website usability, 2. Enhance the visual design, 3. Optimize page load speed"                    | String  |
| projectScope       | varchar | Scope of the project                            | "The project will focus on redesigning the homepage and key landing pages."                                  | String  |
| numOfStudents      | int     | Number of student of the project                | 2  | Integer |
| studentOneSub      | varchar | Student one subtitle of the project             | "Front-end Developer"  | String  |
| studentOneWork     | varchar | Student one work of the project                 | "Responsible for implementing the new design using HTML, CSS, and JavaScript."                               | String  |
| studentTwoSub      | varchar | Student two subtitle of the project             | "UI/UX Designer"   | String  |
| studentTwoWork     | varchar | Student two work of the project                 | "Responsible for creating wireframes, prototypes, and visual assets for the redesigned website."             | String  |
| industryCollab     | text    | Status of industry collaboration of the project | "Yes"  | String  |
| industryCompany    | varchar | Industry company name of the project            | "ABC Corporation"  | String  |
| industryContact    | varchar | Industry contact number of the project          | "John Smith (123-456-7890)"  | String  |

|                            |         |  |   |         |
|----------------------------|---------|--|---|---------|
| projectPostedBy            | text    | Name of the poster of the project                  | "Jane Doe"  | String  |
| projectStatus              | varchar | Status of the project                              | "Accepted"  | String  |
| studentAssigned            | varchar | Student assignment status of the project           | "Yes"   | String  |
| moderatorAssigned          | varchar | Moderator assignment status of the project         | "Yes"   | String  |
| supervisor_id              | int     | Id of poster of the project                        | 1   | Integer |
| moderator_id               | int     | References the account associated with the project | 2   | Integer |
| student                    | json    | Student information of the project                 | {"studentOne": {"name": "John", "email": "john@example.com"}, "studentTwo": {"name": "Emma", "email": "emma@example.com"} } | Object  |
| moderator_name             | varchar | Name of the moderator assigned to the project      | "David Johnson"   | String  |
| projectUpload              | jsonb   | Student report of the project                      | {"file_name": "final_report.pdf", "file_size": "5.2 MB"}  | Object  |
| projectUpload1             | jsonb   | Student one report of the project                  | {"file_name": "student1_report.pdf", "file_size": "3.1 MB"}   | Object  |
| projectUpload2             | jsonb   | Student two report of the project                  | {"file_name": "student2_report.pdf", "file_size": "2.1 MB"}   | Object  |
| <b>Presentations Table</b> |         |  |   |         |
| presId                     | int     | Unique identifier for each presentation            | 1   | Integer |

|                              |         |  |                       |           |
|------------------------------|---------|--|-----------------------|-----------|
| presType                     | varchar | Presentation type for each presentation            | "Poster Presentation" | String    |
| presVenue                    | varchar | Presentation venue for each presentation           | "Grand Ballroom"      | String    |
| presDate                     | date    | Presentation date for each presentation            | "2023-07-15"          | Datetime  |
| presTime                     | time    | Presentation time for each presentation            | "14:00:00"            | Timestamp |
| presSpecialization           | varchar | Specialization for each presentation               | "Software Engineer"   | String    |
| <b>Meetings Table</b>        |         |  |                       |           |
| meetId                       | int     | Unique identifier for each meeting                 | 1                     | Integer   |
| projId                       | int     | References the project associated with the meeting | 1                     | Integer   |
| meetDate                     | date    | Meeting date for each meeting                      | "2023-08-05"          | Datetime  |
| meetTime                     | time    | Meeting time for each meeting                      | "10:30:00"            | Timestamp |
| supervisor_id                | int     | Id of the creator of the meeting                   | 1                     | Integer   |
| stuId                        | int     | References the student associated with the meeting | 1                     | Integer   |
| <b>Meeting Details Table</b> |         |  |                       |           |
| meetDetId                    | int     | Unique identifier for each meeting details         | 1                     | Integer   |

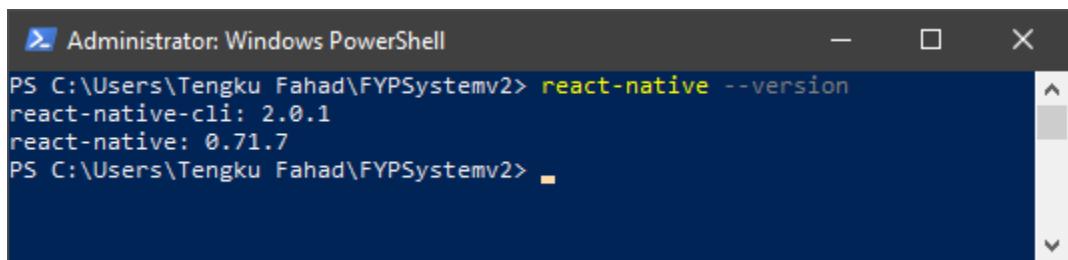
|                       |         |  |   |         |
|-----------------------|---------|--|---|---------|
| meetId                | int     | References the meeting associated with the meeting details | 1   | Integer |
| week                  | int     | Meeting week for each meeting details                      | 4   | Integer |
| meetingMode           | varchar | Meeting mode for each meeting details                      | "Online"  | String  |
| workDone              | varchar | Work done for each meeting details                         | "Reviewed project progress and addressed design issues"                 | String  |
| workToBeDone          | varchar | Work to be done for each meeting details                   | "Implement user authentication feature and conduct performance testing" | String  |
| comments              | varchar | Comments for each meeting details                          | "Good progress made, need to focus on improving load times"             | String  |
| meetStatus            | varchar | Status for each meeting details                            | "Done"  | String  |
| <b>Students Table</b> |         |  |   |         |
| stuId                 | int     | Unique identifier for each student                         | 1   | Integer |
| name                  | varchar | Name for each student                                      | "John Doe"  | String  |
| email                 | varchar | Email for each student                                     | "johndoe@example.com"   | String  |
| projId                | int     | References project associated with the student             | 1   | Integer |

## Chapter 5: IMPLEMENTATION

### 5.1 Deployment

The deployment process of the solution system consists of a React Native frontend, Node.js backend, EC2 cloud computing service, PostgreSQL database hosted on EC2, and S3 storage service. Deployment plays a critical role in making the system accessible to users and ensuring its proper functioning. Additionally, best practices that ensure a seamless and efficient deployment process will be discussed.

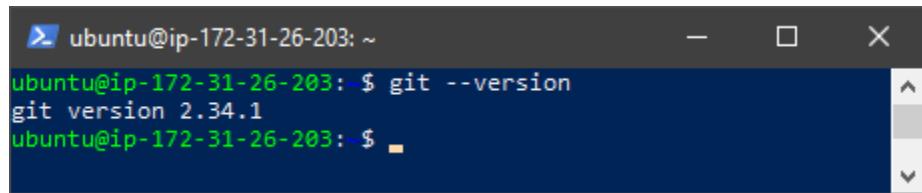
The frontend of the solution system is built using React Native, enabling the development of cross-platform mobile applications.



```
Administrator: Windows PowerShell
PS C:\Users\Tengku Fahad\FYPSystemv2> react-native --version
react-native-cli: 2.0.1
react-native: 0.71.7
PS C:\Users\Tengku Fahad\FYPSystemv2>
```

Figure 5.1 React Native in Machine

To deploy the frontend, it is essential to set up a version control system such as Git to manage the codebase effectively.

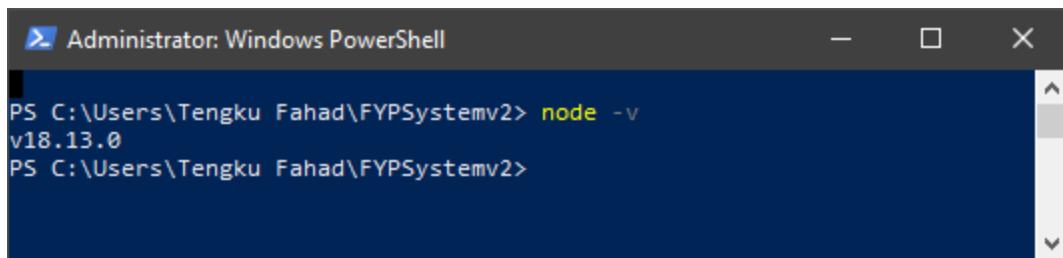


```
ubuntu@ip-172-31-26-203: ~
ubuntu@ip-172-31-26-203:~$ git --version
git version 2.34.1
ubuntu@ip-172-31-26-203:~$
```

Figure 5.2 Git in Machine

Additionally, implementing a continuous integration and continuous deployment (CI/CD) pipeline streamlines the deployment process by automating tasks such as building, testing, and deploying the frontend code to the production environment.

The backend of the solution system is developed using Node.js, a versatile JavaScript runtime for server-side development. To deploy the backend, it is crucial to organize the code into modules, ensuring proper separation of concerns and maintainability.



```
Administrator: Windows PowerShell
PS C:\Users\Tengku Fahad\FYPSystemv2> node -v
v18.13.0
PS C:\Users\Tengku Fahad\FYPSystemv2>
```

Figure 5.3 Node in Machine

A robust web framework like Express.js can be used to handle HTTP requests, define routes, and implement middleware for authentication and authorization. The deployment process involves setting up the production environment with necessary configurations, dependencies, and security measures.

For cloud computing needs, the solution system utilizes Amazon EC2.

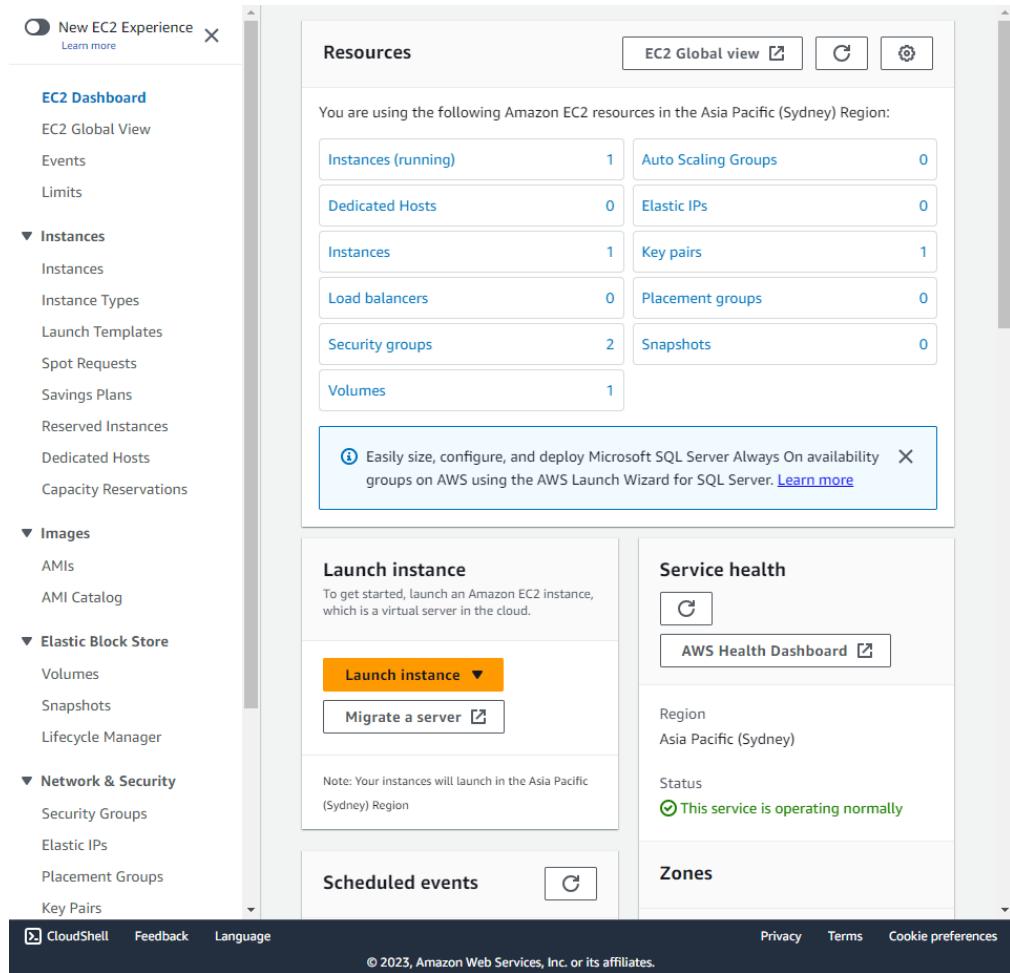


Figure 5.4 EC2 Management Console

Deploying on EC2 involves provisioning instances based on system requirements, considering factors like instance type, storage capacity, and scalability needs. Security groups are configured to control inbound and outbound traffic, allowing access only to necessary ports and protocols. Once the instances are provisioned, software dependencies such as Node.js and PostgreSQL are installed and configured, ensuring compatibility and smooth operation of the system.

Amazon S3 is used as the storage service for managing static assets like documents, images, and files in the solution system.

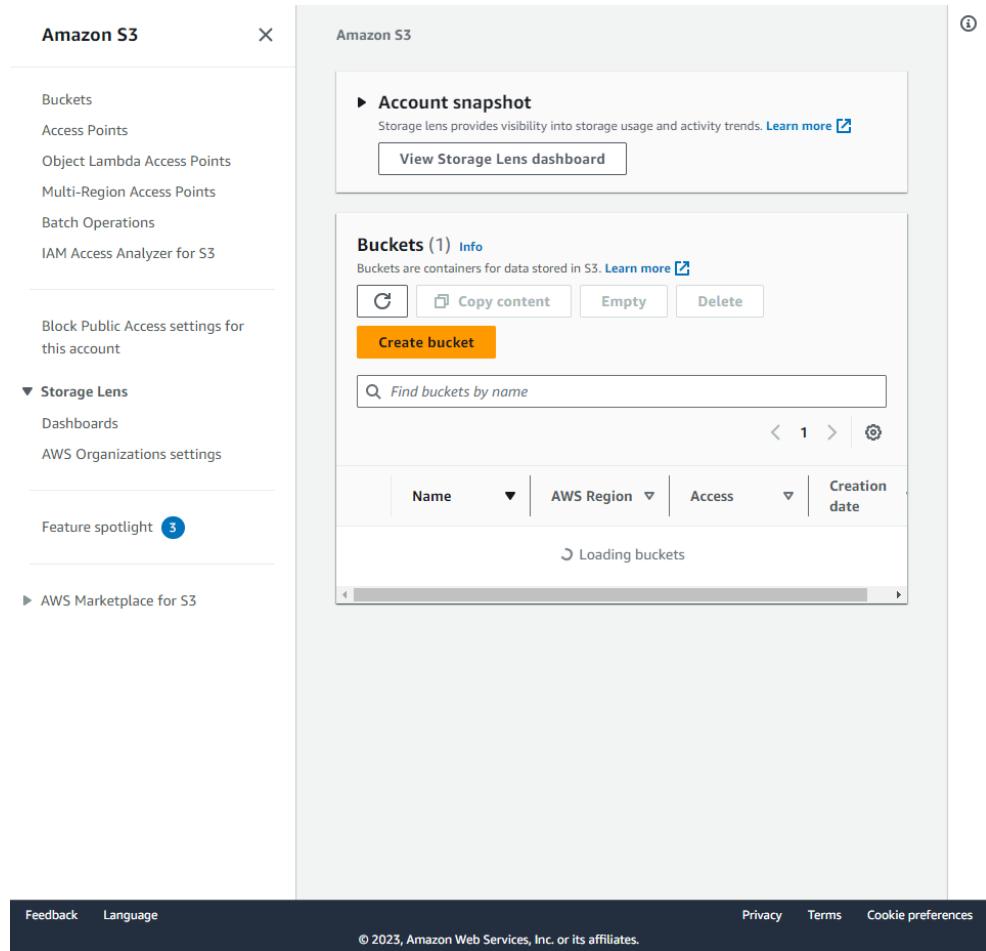
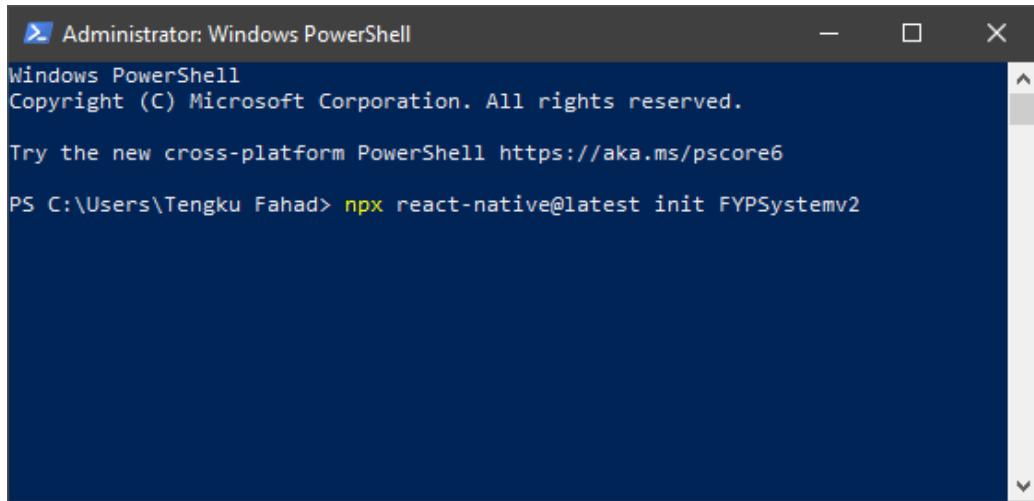


Figure 5.5 S3 Management Console

Deploying S3 involves creating buckets and configuring appropriate access permissions to secure the data stored within. Integration with both the frontend and backend is implemented to facilitate file uploads, downloads, and management.

### 5.1.1 System Configuration

During the system configuration phase, the server was set up, and network connections were established to ensure optimal performance and connectivity. The React Native project was created by executing the command "npx react-native@latest init FYPSystemv2" in Windows PowerShell.



A screenshot of an Administrator: Windows PowerShell window. The command entered is `npx react-native@latest init FYPSystemv2`. The output shows standard PowerShell startup information followed by the confirmation of the new project creation.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

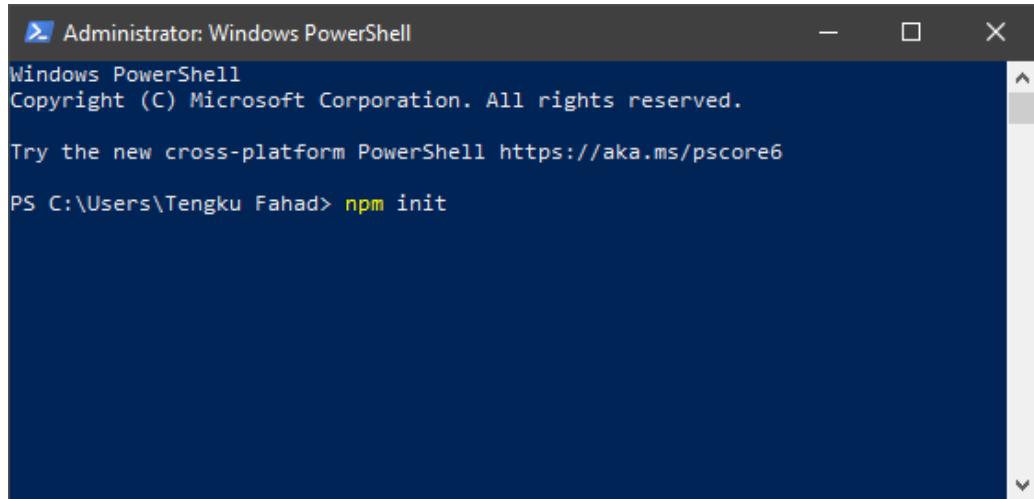
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tengku Fahad> npx react-native@latest init FYPSystemv2
```

Figure 5.6 Executing React Native Project Creation in Windows PowerShell

This command initialized a new React Native project called "FYPSystemv2". To facilitate testing, the Android Studio Android Virtual Device (AVD) emulator was used to run and test the application on a virtual Android device.

The backend development utilized Node.js, where the backend was initialized by executing the command "npm init" to create a new Node.js project.



A screenshot of an Administrator: Windows PowerShell window. The command entered is `npm init`. The output shows standard PowerShell startup information followed by the confirmation of the new project creation.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Tengku Fahad> npm init
```

Figure 5.7 Executing Node.js Project Creation in Windows PowerShell

Visual Studio Code served as the preferred text editor and Integrated Development Environment (IDE) for coding the project.

For hosting the solution system, an Amazon EC2 instance was set up in the ap-southeast-2 region.

| Instance summary for i-038be763a0bd650ba <a href="#">Info</a>  |   |   |
|--|---|---|
| Updated less than a minute ago   |   |   |
| <a href="#">C</a> <a href="#">Connect</a> <a href="#">Instance state ▾</a> <a href="#">Actions ▾</a> |   |   |
| Instance ID<br><a href="#">i-038be763a0bd650ba</a>   | Public IPv4 address<br><a href="#">3.26.23.172</a>   <a href="#">open address</a>                   | Private IPv4 addresses<br><a href="#">172.31.26.203</a>   |
| IPv6 address<br>-  | Instance state<br><a href="#">Running</a>   | Public IPv4 DNS<br><a href="#">ec2-3-26-23-172.ap-southeast-2.compute.amazonaws.com</a>   <a href="#">open address</a>                |
| Hostname type<br>IP name: ip-172-31-26-203.ap-southeast-2.compute.internal                           | Private IP DNS name (IPv4 only)<br><a href="#">ip-172-31-26-203.ap-southeast-2.compute.internal</a> | Elastic IP addresses<br>-   |
| Answer private resource DNS name<br>IPv4 (A)   | Instance type<br>t2.micro   | AWS Compute Optimizer finding<br><a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a><br>  <a href="#">Learn more</a> |
| Auto-assigned IP address<br><a href="#">3.26.23.172</a> [Public IP]                                  | VPC ID<br><a href="#">vpc-0b1609575f628e056</a>   | Auto Scaling Group name<br>-  |
| IAM Role<br><a href="#">fyp-admin</a>  | Subnet ID<br><a href="#">subnet-0015303056535931d</a>   |   |
| IMDSv2<br>Optional   |   |   |

Figure 5.8 EC2 Instance

Additionally, a key pair was created to enable secure SSH access. The key (pem file) is then stored in the React Native project folder.

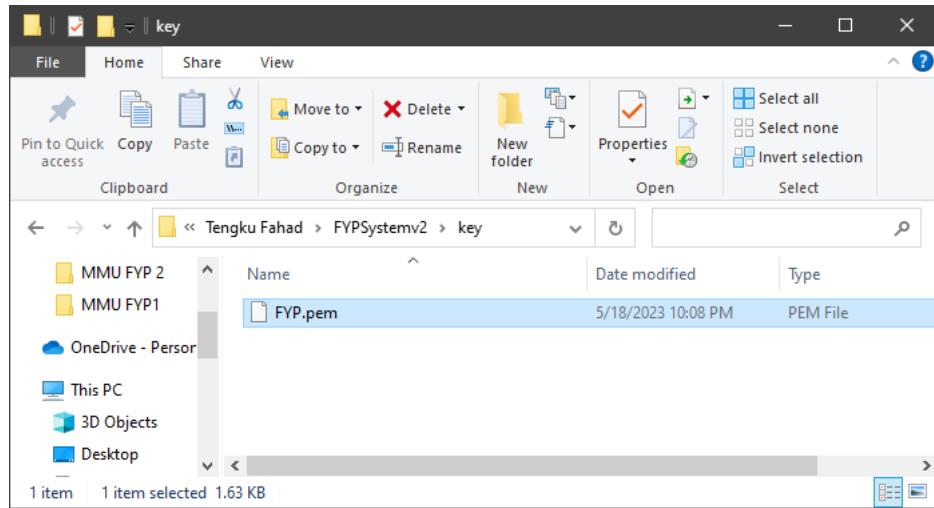


Figure 5.9 EC2 Key

The security group associated with the EC2 instance was configured to allow inbound rules, including SSH access and necessary ports for communication.

| Inbound rules (4)      |            |            |          |            |  |
|------------------------|------------|------------|----------|------------|--|
| Security group rule... | IP version | Type       | Protocol | Port range |  |
| sgr-02e8449e474782...  | IPv4       | SSH        | TCP      | 22         |  |
| sgr-0a1cdeafa145df009  | IPv4       | HTTP       | TCP      | 80         |  |
| sgr-05838489e0e786c... | IPv4       | HTTPS      | TCP      | 443        |  |
| sgr-04d20581b68eb4...  | IPv4       | Custom TCP | TCP      | 3000       |  |

Figure 5.10 EC2 Inbound Rules

To store and retrieve project files, an Amazon S3 bucket was configured by creating a bucket in the AWS Management Console.

| Bucket overview                                    |  |   |
|--|--|---|
| AWS Region<br>Asia Pacific (Sydney) ap-southeast-2 | Amazon Resource Name (ARN)<br>arn:aws:s3:::fyp-uploads | Creation date<br>June 15, 2023, 01:27:59<br>(UTC+08:00) |

Figure 5.11 AWS S3 Bucket Overview

AWS Identity and Access Management (IAM) was employed to establish secure interaction between EC2, React Native, and S3. This involved creating a user and role with appropriate permissions.

| Permissions policies (1)   |                             |             |                |           |
|--|-----------------------------|-------------|----------------|-----------|
| Permissions are defined by policies attached to the user directly or through groups. |                             |             |                |           |
|  |                             |             |                |           |
|  |                             |             | Filter by Type |           |
|  | <input type="text"/> Search |             | All types      |           |
|  |                             |             | < 1 >          |           |
|  | <input type="checkbox"/>    | Policy name | Type           | Attach... |
|  | <input type="checkbox"/>    |             | AWS managed    | Directly  |

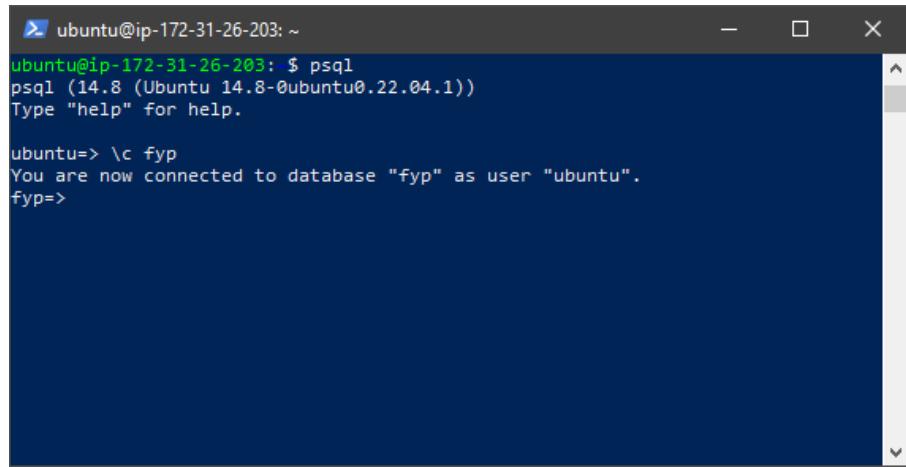
Figure 5.12 IAM Users Permissions

| Permissions policies (2) <small>Info</small> |  |             |                            |   |
|--|--|-------------|----------------------------|---|
| You can attach up to 10 managed policies.    |  |             |                            |   |
|  |  |             |                            |   |
|  |  |             | Add permissions            |   |
|  | <input type="text"/> Filter policies by property or policy name and press enter. |             |                            |   |
|  |  |             | < 1 >                      |   |
|  | <input type="checkbox"/>   | Policy name | Type                       | D |
|  | <input type="checkbox"/>   |             | AWS managed - job function | P |
|  | <input type="checkbox"/>   |             | AWS managed                | P |

Figure 5.13 IAM Role Permissions

### 5.1.2 Database

The solution system utilizes PostgreSQL as the database management system, which is hosted on the EC2 instances.



```
ubuntu@ip-172-31-26-203: ~
ubuntu@ip-172-31-26-203: $ psql
psql (14.8 (Ubuntu 14.8-0ubuntu0.22.04.1))
Type "help" for help.

ubuntu=> \c fyp
You are now connected to database "fyp" as user "ubuntu".
fyp=>
```

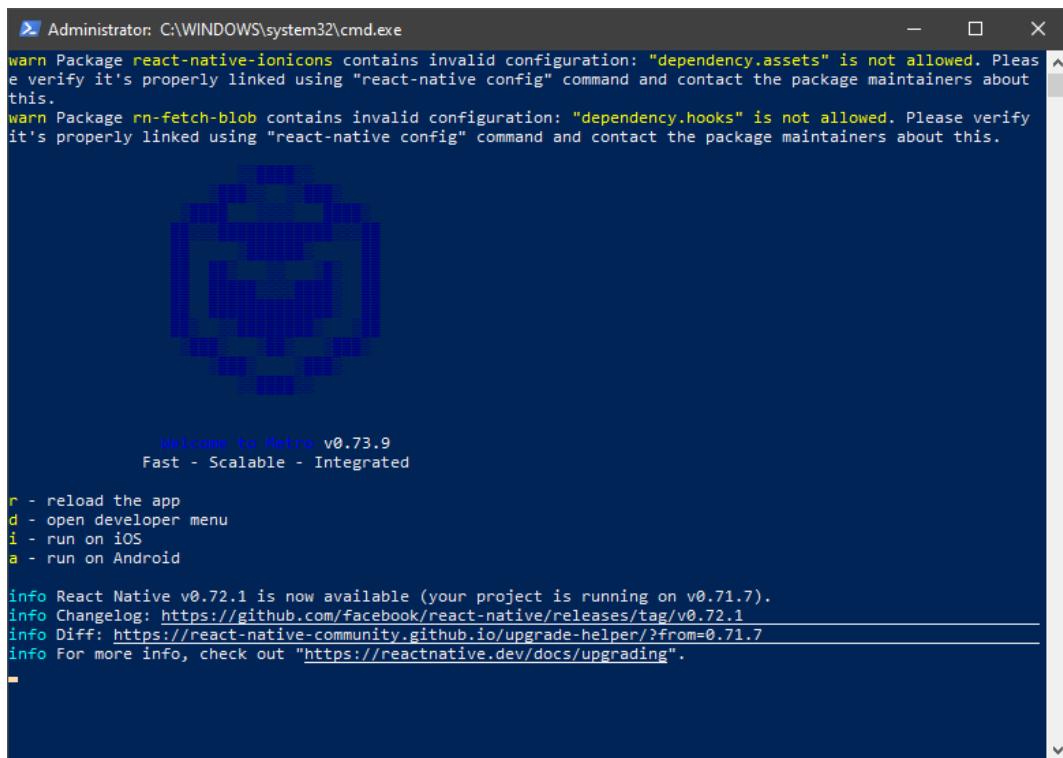
Figure 5.14 Connecting to psql through EC2 instance

PostgreSQL (psql) was installed and configured as the Database Management System (DBMS) on the EC2 instance for efficient data storage and retrieval. Deploying the database involves creating the necessary database schema and tables to store and manage the application data.

## 5.2 Development Environment

The development environment for this project has been configured during the deployment process. It encompasses various tools and technologies necessary for coding, and testing

React Native, a cross-platform mobile application framework, has been successfully set up. To run the React Native application, commands such as “npx react-native start” and “npx react-native start-android” are used in Windows PowerShell.



```
Administrator: C:\WINDOWS\system32\cmd.exe
warn Package react-native-ionicons contains invalid configuration: "dependency.assets" is not allowed. Please verify it's properly linked using "react-native config" command and contact the package maintainers about this.
warn Package rn-fetch-blob contains invalid configuration: "dependency.hooks" is not allowed. Please verify it's properly linked using "react-native config" command and contact the package maintainers about this.

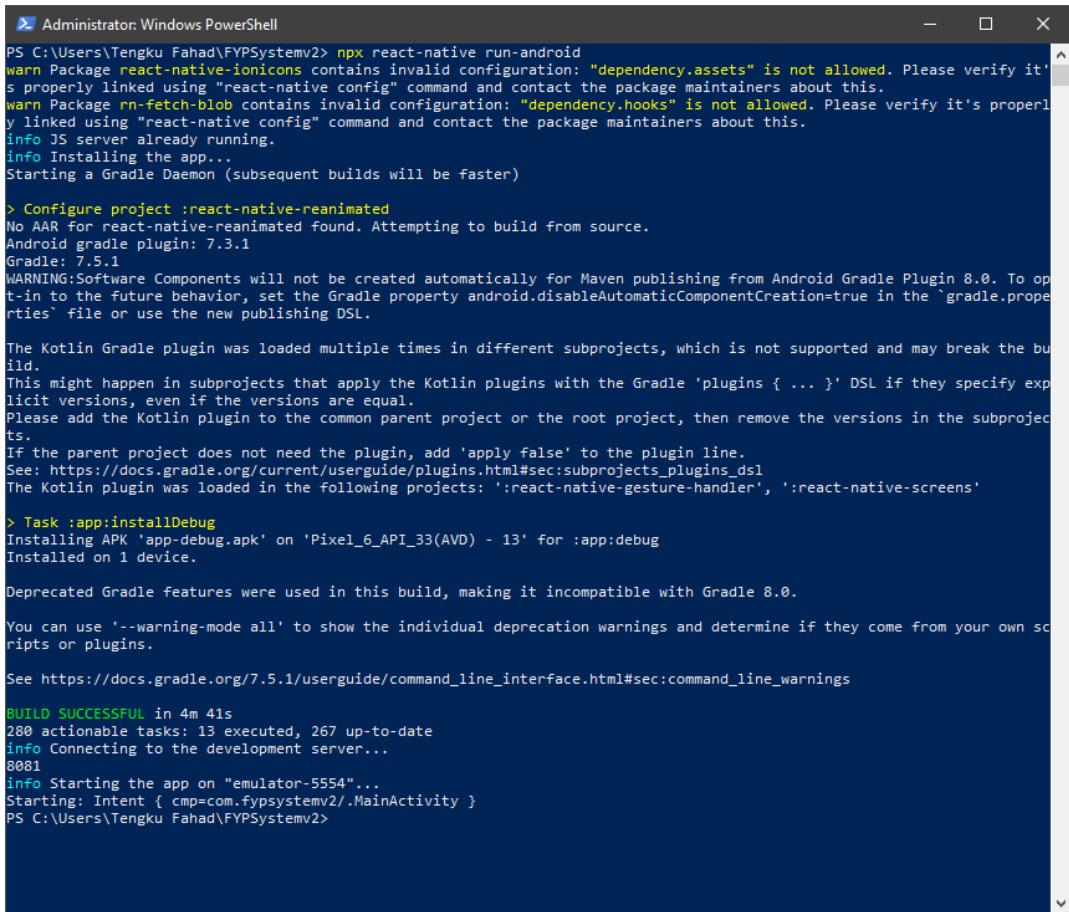
[Pixelated React Native logo]

Welcome to Metro v0.73.9
Fast - Scalable - Integrated

r - reload the app
d - open developer menu
i - run on iOS
a - run on Android

info React Native v0.72.1 is now available (your project is running on v0.71.7).
info Changelog: https://github.com/facebook/react-native/releases/tag/v0.72.1
info Diff: https://react-native-community.github.io/upgrade-helper/?from=0.71.7
info For more info, check out "https://reactnative.dev/docs/upgrading".
```

Figure 5.15 Running React Native



```
Administrator: Windows PowerShell
PS C:\Users\Tengku Fahad\FYPSystemv2> npx react-native run-android
warn Package react-native-ionicons contains invalid configuration: "dependency.assets" is not allowed. Please verify it's properly linked using "react-native config" command and contact the package maintainers about this.
warn Package rn-fetch-blob contains invalid configuration: "dependency.hooks" is not allowed. Please verify it's properly linked using "react-native config" command and contact the package maintainers about this.
info JS server already running.
info Installing the app...
Starting a Gradle Daemon (subsequent builds will be faster)

> Configure project :react-native-reanimated
No AAR for react-native-reanimated found. Attempting to build from source.
Android gradle plugin: 7.3.1
Gradle: 7.5.1
WARNING:Software Components will not be created automatically for Maven publishing from Android Gradle Plugin 8.0. To opt-in to the future behavior, set the Gradle property android.disableAutomaticComponentCreation=true in the `gradle.properties` file or use the new publishing DSL.

The Kotlin Gradle plugin was loaded multiple times in different subprojects, which is not supported and may break the build.
This might happen in subprojects that apply the Kotlin plugins with the Gradle 'plugins { ... }' DSL if they specify explicit versions, even if the versions are equal.
Please add the Kotlin plugin to the common parent project or the root project, then remove the versions in the subprojects.
If the parent project does not need the plugin, add 'apply false' to the plugin line.
See: https://docs.gradle.org/current/userguide/plugins.html#sec:subprojects\_plugins\_dsl
The Kotlin plugin was loaded in the following projects: ':react-native-gesture-handler', ':react-native-screens'

> Task :app:installDebug
Installing APK 'app-debug.apk' on 'Pixel_6_API_33(AVD) - 13' for :app:debug
Installed on 1 device.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

See https://docs.gradle.org/7.5.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings

BUILD SUCCESSFUL in 4m 41s
280 actionable tasks: 13 executed, 267 up-to-date
info Connecting to the development server...
8081
info Starting the app on "emulator-5554"...
Starting: Intent { cmp=com.fypsystemv2/.MainActivity }
PS C:\Users\Tengku Fahad\FYPSystemv2>
```

Figure 5.16 Starting Android Emulator

These commands initiate the development server and launch the application on the Android emulator.

The backend of the solution system is built using Node.js. It has been configured on the EC2 instance during the deployment process. Commands like “pm2 start index.js” in Windows PowerShell are utilized to manage and start the Node.js application using PM2, a process manager specifically designed for Node.js applications.

```

ubuntu@ip-172-31-26-203:~/backend
ubuntu@ip-172-31-26-203:~$ pm2 start index.js
[PM2] [ERROR] Script not found: /home/ubuntu/index.js
ubuntu@ip-172-31-26-203:~$ cd ~/backend
ubuntu@ip-172-31-26-203:~/backend$ pm2 start index.js
[PM2] Applying action restartProcessId on app [index](ids: [ 0 ])
[PM2] [index](0) @
[PM2] Process successfully started

```

| id | name  | mode | status | cpu    | memory |
|----|-------|------|--------|--------|--------|
| 0  | index | Fork | 63     | online | 0%     |
|    |       |      |        |        | 16.7mb |

```

ubuntu@ip-172-31-26-203:~/backend$ 

```

Figure 5.17 Starting pm2

PostgreSQL, a robust database management system, has been installed on the EC2 instance. The psql command-line interface is used to create the PostgreSQL database, set up tables, and perform necessary database operations.

```

ubuntu@ip-172-31-26-203: ~
ubuntu@ip-172-31-26-203:~$ psql
psql (14.8 (Ubuntu 14.8-0ubuntu0.22.04.1))
Type "help" for help.

ubuntu=> \c fyp
You are now connected to database "fyp" as user "ubuntu".
fyp=> \d
fyp=> \d
      List of relations
 Schema |        Name         |   Type   | Owner
-----+----------------+-----+-----
 public | accounts        | table  | ubuntu
 public | accounts_accid_seq | sequence | ubuntu
 public | accounts_address_id_seq | sequence | ubuntu
 public | addresses        | table  | ubuntu
 public | addresses_id_seq | sequence | ubuntu
 public | announcements    | table  | ubuntu
 public | announcements_annnid_seq | sequence | ubuntu
 public | meeting_details  | table  | ubuntu
 public | meeting_details_meetdetid_seq | sequence | ubuntu
 public | meetings          | table  | ubuntu
 public | meetings_meetid_seq | sequence | ubuntu
 public | presentations     | table  | ubuntu
 public | presentations_presid_seq | sequence | ubuntu
 public | projects          | table  | ubuntu
 public | projects_projid_seq | sequence | ubuntu
 public | students           | table  | ubuntu
 public | students_stuid_seq | sequence | ubuntu
(17 rows)
fyp=>

```

Figure 5.18 List of Database Tables

### 5.2.1 Development IDE

Visual Studio Code was the primary development IDE. It provided a comprehensive set of features and extensions that enhanced our coding experience. Its intuitive interface, built-in terminal, and extensive plugin ecosystem allowed us to write and debug code effectively.

```

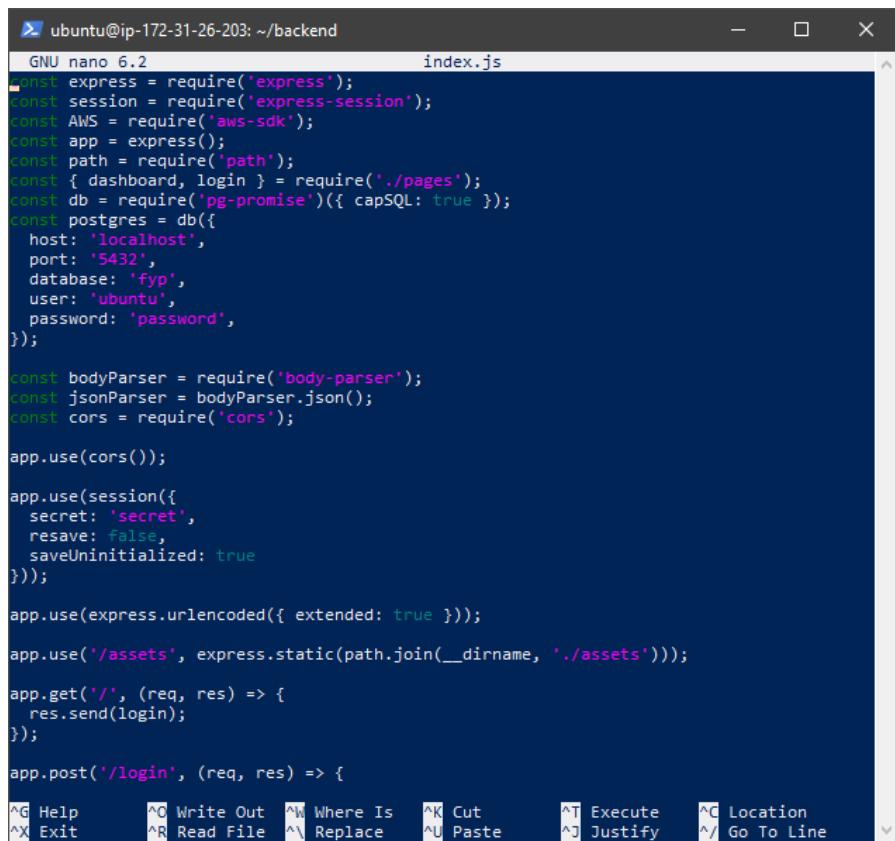
File Edit Selection View Go Run ...
App.tsx - FVPSystmv2 - Visual Studio Code [Administrator]
OPEN EDITORS
EXPLORER
App.tsx
  1 import React from 'react';
  2 import { createStackNavigator } from '@react-navigation/stack';
  3 import { NavigationContainer } from '@react-navigation/native';
  4
  5 import { AuthProvider } from './AuthContext';
  6
  7 import LoginScreen from './Login';
  8 import SUMainContainer from './navigation/SUmainContainer';
  9 import FMMainContainer from './navigation/FMmainContainer';
10 import COMainContainer from './navigation/COMainContainer';
11 import MOMainContainer from './navigation/MOmainContainer';
12
13 // Import screens
14 import SUProjPropScreen from './navigation/Screens/SUProjPropScreen';
15 import SUMeetCreateScreen from './navigation/Screens/SUMeetCreateScreen';
16 import FMCreateAnsScreen from './navigation/Screens/FMCreateAnsScreen';
17 import COCreateAnsScreen from './navigation/Screens/COCreateAnsScreen';
18 import FMPresCreateScreen from './navigation/Screens/FMPresCreateScreen';
19 import FMPresInfoScreen from './navigation/FMScreens/FMPresInfoScreen';
20 import SUProjInfoScreen from './navigation/Screens/SUProjInfoScreen';
21 import SUMeetInfoScreen from './navigation/Screens/SUMeetInfoScreen';
22 import FMMeetInfoScreen from './navigation/FMScreens/FMMeetInfoScreen';
23 import COModAssignScreen from './navigation/OScreens/COModAssignScreen';
24 import SUStuAssignScreen from './navigation/Screens/SUStuAssignScreen';
25 import SUStuAssig2Screen from './navigation/Screens/SUStuAssig2Screen';
26 import SUMeetWeekScreen from './navigation/Screens/SUMeetWeekScreen';
27 import SUMeetWeekInputScreen from './navigation/Screens/SUMeetWeekInputScreen';
28 import ListOfStudents from './navigation/LISTScreens/ListOfStudents';
29 import ListOfModerators from './navigation/LISTScreens/ListOfModerators';
30 import ListOfSupervisors from './navigation/LISTScreens/ListOfSupervisors';
31
32 const Stack = createStackNavigator();
33
34 function App(): JSX.Element {
35   return (
36     <NavigationContainer>
37       <AuthProvider>
38         <Stack.Navigator>
39           /* Set the login screen as the initial screen */
40           <Stack.Screen name="Login" component={LoginScreen} options={{ headerShown: false }} />
41
42           /* Main Containers */
43           <Stack.Screen name="SUMainScreen" component={SUMainContainer} options={{ headerShown: false }} />
44           <Stack.Screen name="FMMainScreen" component={FMMainContainer} options={{ headerShown: false }} />
45           <Stack.Screen name="COMainContainer" component={COMainContainer} options={{ headerShown: false }} />
46           <Stack.Screen name="MOMainScreen" component={MOMainContainer} options={{ headerShown: false }} />
47
48           /* Screens */
49           <Stack.Screen name="SUProjPropScreen" component={SUProjPropScreen} options={{ headerShown: false }} />
50           <Stack.Screen name="SUMeetCreateScreen" component={SUMeetCreateScreen} options={{ headerShown: false }} />
51           <Stack.Screen name="FMCreateAnsScreen" component={FMCreateAnsScreen} options={{ headerShown: false }} />
52           <Stack.Screen name="COCreateAnsScreen" component={COCreateAnsScreen} options={{ headerShown: false }} />
53           <Stack.Screen name="FMPresCreateScreen" component={FMPresCreateScreen} options={{ headerShown: false }} />
54
55           <Stack.Screen name="FMPresInfoScreen" component={FMPresInfoScreen} options={{ headerShown: true }} />
56           <Stack.Screen name="SUProjInfoScreen" component={SUProjInfoScreen} options={{ headerShown: true }} />
57           <Stack.Screen name="SUMeetInfoScreen" component={SUMeetInfoScreen} options={{ headerShown: true }} />
58           <Stack.Screen name="FMMeetInfoScreen" component={FMMeetInfoScreen} options={{ headerShown: true }} />
59           <Stack.Screen name="COModAssignScreen" component={COModAssignScreen} options={{ headerShown: true }} />
60           <Stack.Screen name="SUStuAssignScreen" component={SUStuAssignScreen} options={{ headerShown: true }} />
61           <Stack.Screen name="SUStuAssig2Screen" component={SUStuAssig2Screen} options={{ headerShown: true }} />
62           <Stack.Screen name="SUMeetWeekScreen" component={SUMeetWeekScreen} options={{ headerShown: true }} />
63           <Stack.Screen name="SUMeetWeekInputScreen" component={SUMeetWeekInputScreen} options={{ headerShown: true }} />
64
65           /* List screens */
66           <Stack.Screen name="ListOfStudents" component={ListOfStudents} options={{ headerShown: true }} />
67           <Stack.Screen name="ListofModerators" component={ListofModerators} options={{ headerShown: true }} />
68           <Stack.Screen name="ListofSupervisors" component={ListofSupervisors} options={{ headerShown: true }} />
69
70         </Stack.Navigator>
71       <AuthProvider>
72         <NavigationContainer>
73       </NavigationContainer>
74     </AuthProvider>
75   );
}

```

Figure 5.19 Visual Code Studio

### 5.2.2 Development Tools

In addition to the IDE, several development tools were leveraged to streamline the development process. Nano, a lightweight text editor, was utilized for editing code files related to the Node.js backend.



The screenshot shows a terminal window titled "ubuntu@ip-172-31-26-203: ~/backend". Inside the terminal, the command "GNU nano 6.2" is running on the file "index.js". The code in the file is a Node.js application. It starts by requiring 'express', 'express-session', 'aws-sdk', and 'pg-promise'. It then defines a database connection 'postgres' using 'pg-promise' with a configuration object containing host ('localhost'), port ('5432'), database ('fyp'), user ('ubuntu'), and password ('password'). The application uses 'body-parser' for parsing requests and 'cors' for handling cross-origin requests. It sets up session middleware with a secret ('secret') and saves uninitialized sessions. It also enables URL encoding and serves static assets from the current directory. Finally, it defines routes for the root ('/') which returns the 'login' page, and a POST route for '/login' which handles login requests. The bottom of the window shows the standard nano key bindings.

```
GNU nano 6.2                               index.js
const express = require('express');
const session = require('express-session');
const AWS = require('aws-sdk');
const app = express();
const path = require('path');
const { dashboard, login } = require('./pages');
const db = require('pg-promise')({ capsSQL: true });
const postgres = db({
  host: 'localhost',
  port: '5432',
  database: 'fyp',
  user: 'ubuntu',
  password: 'password',
});

const bodyParser = require('body-parser');
const jsonParser = bodyParser.json();
const cors = require('cors');

app.use(cors());

app.use(session({
  secret: 'secret',
  resave: false,
  saveUninitialized: true
}));

app.use(express.urlencoded({ extended: true }));

app.use('/assets', express.static(path.join(__dirname, './assets')));

app.get('/', (req, res) => {
  res.send(login);
});

app.post('/login', (req, res) => {
```

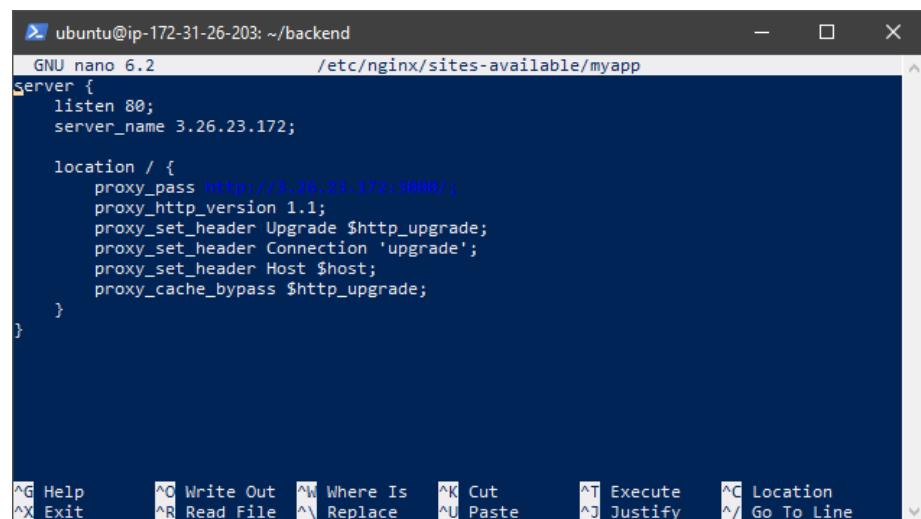
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^Y Replace ^U Paste ^J Justify ^/ Go To Line

Figure 5.20 Nano

Git, a widely-used version control system, enabled efficient management of source code changes, and revision tracking.

For managing the Node.js application on the EC2 instance, PM2 was employed. PM2 acted as a process manager, offering advanced features such as automatic application restart, load balancing, and monitoring.

To optimize the performance and scalability of the Node.js application, Nginx was implemented as a reverse proxy and load balancer. Nginx efficiently distributed incoming requests, resulting in improved response times and overall system performance enhancement.



The screenshot shows a terminal window titled "ubuntu@ip-172-31-26-203: ~/backend". The window contains the following nginx configuration code:

```
GNU nano 6.2          /etc/nginx/sites-available/myapp
server {
    listen 80;
    server_name 3.26.23.172;

    location / {
        proxy_pass http://3.26.23.172:3000/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for navigating and editing the file.

Figure 5.12 nginx Configuration

By carefully configuring the development environment and utilizing the appropriate tools, a seamless and productive development experience was ensured for the solution system.

## ***5.3 Software Modules***

The application includes several main modules for FYP (Final Year Project) management. The authentication module allows secure login. The announcement module enables the creation and viewing of important announcements. The project module handles project proposal creation, management, and approval. The report module allows uploading and downloading project reports. The meeting module supports scheduling and updating meetings with students. The presentation module facilitates the creation of project presentations.

### ***5.3.1 Authentication Algorithm***

The Authentication Algorithm is responsible for handling the authentication process within the app. It defines the steps and logic required to authenticate users and manage their login status. Algorithm Description and Sample code can be found in Appendix E (E - 5.3.1).

### ***5.3.2 Announcement Creation Algorithm***

The Announcement Creation Algorithm is responsible for creating announcements within the app. It outlines the steps and logic required to create an announcement with the provided data.

Algorithm Description:

1. Retrieve the necessary data for creating an announcement, such as the title, description, and any attachments.
2. Validate the input data to ensure it meets the required criteria.
3. If the input data is valid, proceed with the creation process.

4. Handle the attachment upload (if applicable):
  - a. Implement the handleUpload function to open the file picker and allow the user to select a file.
  - b. Use the DocumentPicker library to handle the file selection.
  - c. Update the upload state with the selected file(s).
  - d. Provide a UI to display the selected file and allow the user to clear it if needed.
5. Handle the announcement submission:
  - a. Implement the handleSubmit function to handle the announcement creation process.
  - b. Validate that the title and description are not empty.
  - c. If an attachment is selected, upload it to an AWS S3 bucket:
    - Configure the AWS SDK with the necessary credentials and region.
    - Convert the file to base64 and create an arrayBuffer.
    - Create an AWS S3 instance and set the upload parameters.
    - Use the s3.upload method to upload the file to the S3 bucket.
    - Obtain the signed URL for the uploaded file.
  - d. Make a POST request to the server's API endpoint for creating announcements.
    - Include the announcement data, uploaded file URL (if applicable), and user information in the request body.

e. Handle the server response:

- If the response is successful, display a success message to the user.
- If the response indicates an error, handle the error logic appropriately.

6. Display the UI elements for entering the announcement details, including the title and description fields.

7. Provide validation and feedback for the input fields:

- Validate that the title and description are not empty when submitting the announcement.
- Display error messages or visual indicators if the input is invalid.

8. Display the UI element for selecting an attachment file, including the "Choose File" button and the selected file name (if applicable).

9. Display the "Create Announcement" button to initiate the announcement creation process.

10. Implement a modal component to display success or error messages to the user.

11. Navigate to the appropriate screen after a successful announcement creation.

Sample Code:

The following code snippet demonstrates the implementation of the announcement creation algorithm:

JavaScript

```
import React, { useState, useRef, useContext } from 'react';
import { SafeAreaView, ScrollView, View, Text, TextInput, StyleSheet,
TouchableOpacity } from 'react-native';
```

```

import DocumentPicker from 'react-native-document-picker'; // Import
the document picker library
import { decode } from 'base64-arraybuffer';
import AWS from 'aws-sdk';

import { AuthContext } from '../../AuthContext';

function Ann_CreateScreen({ navigation }) {
  const { user } = useContext(AuthContext);

  // State and event handlers

  const handleSubmit = async () => {
    try {
      if (!annTitle || !annDescription) {
        setModalTitle('Error');
        setModalMessage('Please ensure that you fill in all the
required fields');
        setModalButtonText('Try again');
        setModalVisible(true);
        return;
      }
    }

    // If a file is selected, upload it to S3 bucket
    let uploadedFileUrl = '';
    let fileKey = '';
    if (upload.length > 0) {
      // AWS S3 configuration
      AWS.config.update({
        accessKeyId: 'ACCESS_KEY_ID',
        secretAccessKey: 'SECRET_ACCESS_KEY',
        region: 'AWS_REGION',
      });

      const base64 = await fs.readFile(upload[0].uri, 'base64');
      const arrayBuffer = decode(base64);
      const s3 = new AWS.S3();
      fileKey = `${user.accid}_${upload[0].name}`;

      // Configure the parameters for the S3 upload
      const params = {
        Bucket: 'fyp-uploads',
        Key: fileKey,
        Body: arrayBuffer,
        ContentType: upload[0].type,
        ContentDisposition: 'attachment',
      };
    }
  }
}

```

```

    // Upload the file to the S3 bucket
    const result = await s3.upload(params).promise();
    const expirationTime = 10 * 365 * 24 * 60 * 60; // 10 years
in seconds
        uploadedFileUrl = s3.getSignedUrl('getObject', { Bucket:
'fyp-uploads', Key: fileKey, Expires: expirationTime });
    }

    const response = await
fetch('http://3.26.23.172/announcements', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify({
        annTitle,
        annDescription,
        upload: upload.length > 0 ? [{ fileName: upload[0].name,
fileUrl: uploadedFileUrl, bucket: 'fyp-uploads', key: fileKey }] :
[],
        annPostedBy: user.name,
    }),
});
}

if (response.ok) {
    setModalTitle('Success!');
    setModalMessage('Announcement created');
    setModalButtonText('OK');
    setModalVisible(true);
} else {
    console.error('Error creating announcement:', response.statusText);
}
} catch (error) {
    console.error('Error creating announcement:', error);
}
};

// ... (rest of the components and UI elements)

return (
    // ... (JSX for the rest of the screen)
);
}

```

### **5.3.3 Project Proposal Creation Algorithm**

The Project Proposal Creation Algorithm is responsible for creating project proposals within the app. It outlines the steps and logic required to create a project proposal with the provided data.

Algorithm Description:

1. Retrieve the necessary data for creating a project proposal, such as projectTitle, projectType, projectSpecialization, projectFocus, projectDescription, projectObjectives, projectScope, numOfStudents, studentOneSub, studentOneWork, studentTwoSub, studentTwoWork, industryCollab, industryCompany, and industryContact.
2. Validate the input data to ensure all the required fields are filled in and selected.
3. If any of the required fields are missing, display an error message to the user and prompt them to fill in all the required fields.
4. If all the required fields are present, proceed with the creation process.
5. Make an HTTP POST request to the server's API endpoint for creating project proposals.
6. Set the appropriate headers for the request, including the "Content-Type" header with the value "application/json".
7. Create a JSON payload with the project proposal data, including all the necessary fields.
8. Send the JSON payload as the request body.

9. Handle the server response:

- a. If the response is successful (status code 200), display a success message to the user and take any necessary actions.
  - b. If the response indicates an error, display an error message to the user and handle any necessary error logic.
10. If the project proposal creation is successful, display a success message to the user and take any additional actions as needed.
11. If there are any errors during the creation process, handle the errors and display appropriate error messages to the user.

Sample Code:

The following code snippet demonstrates the implementation of the project proposal creation algorithm:

```
JavaScript
import React, { useState, useContext } from 'react';
import { View, Text, TextInput, TouchableOpacity } from
'react-native';
import { AuthContext } from '../../AuthContext';

function Proj_PropScreen({ navigation }) {
  const { user } = useContext(AuthContext);

  // State variables for project proposal data
  const [projectTitle, setProjectTitle] = useState('');
  const [projectType, setProjectType] = useState('');
  // ... other state variables for project proposal data

  // ... other state variables for error handling and UI

  const handleSubmit = async () => {
    try {
```

```

    // Validation check for required fields
    if (!projectTitle || !projectType /* ... other required fields
*/) {
    // Display error message for missing fields
    console.log('Please fill in all the required fields.');
    return;
}

// Create project proposal payload
const projectProposalData = {
    projectTitle,
    projectType,
    // ... other project proposal data
};

// Make HTTP POST request to create project proposal
const response = await fetch('http://3.26.23.172/projects', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify(projectProposalData),
});

if (response.ok) {
    console.log('Project proposal created successfully!');
    navigation.goBack(); // Go back to previous screen
} else {
    console.error('Error creating project proposal:',
    response.statusText);
}
} catch (error) {
    console.error('Error creating project proposal:', error);
}
};

return (
<View>
    {/* ... UI components for project proposal form */}
    <TextInput
        value={projectTitle}
        onChangeText={setProjectTitle}
        // ... other TextInput components for project proposal data
    />
    {/* ... other UI components for project proposal form */}
    <TouchableOpacity onPress={handleSubmit}>
        <Text>Create Project Proposal</Text>
    </TouchableOpacity>

```

```
        </View>
    );
}

export default Proj_PropScreen;
```

### **5.3.4 Project Approval and Rejection Algorithm**

The Project Approval and Rejection Algorithm handles the approval and rejection of project proposals within the app. It describes the steps and logic involved in accepting or declining a project proposal. Algorithm Description and Sample code can be found in Appendix E (E - 5.3.4).

### **5.3.5 Upload and Download Report Algorithm**

The Upload and Download Report Algorithm is responsible for uploading and downloading reports within the app. It outlines the steps and logic required to upload and download reports.

Algorithm Description:

1. Upload Report:

- Implement the handleProjectUpload function to handle the selection of the project report file using a file picker (such as DocumentPicker).
- Retrieve the selected file and set it in the component state using the setProjectUpload function.
- Configure the AWS S3 credentials (accessKeyId, secretAccessKey, and region) using the AWS.config.update method.

- Read the file content using the appropriate method (e.g., `readFile`) and convert it to the desired format (e.g., `base64` or array buffer).
- Create an instance of the AWS S3 service using `new AWS.S3()`.
- Set the necessary parameters for the S3 upload, including the bucket name, file key, body (file content), content type, and content disposition.
- Use the upload method of the S3 service to upload the file to the S3 bucket.
- Retrieve the signed URL for the uploaded file using the `getSignedUrl` method of the S3 service.
- Make an HTTP POST request to the server's API endpoint to update the project report information, including the project ID and the uploaded file details (e.g., file name, file URL, bucket, and key).
- Handle the server response:
- If the response is successful (status code 200), show a success message to the user using a modal or alert.
- If the response indicates an error, handle the error logic appropriately (e.g., display an error message).
- Invoke the `handleProjectUpload` function when the user clicks on the upload button within the `TouchableOpacity` component.

## 2. Download Report:

- Implement the `handleDownloadStudent` function to handle the download of the student report.

- Check if the project has a project report (project.projectupload) and a valid file URL (project.projectupload.fileUrl).
- If the report is available, use the Linking.openURL method to open the file URL and initiate the download.
- If the report is not available, show a modal or alert with a message indicating that the file is not available for download.
- Invoke the handleDownloadStudent function when the user clicks on the download button within the TouchableOpacity component.

Sample Code:

The following code snippet demonstrates the implementation of uploading and downloading reports algorithm:

```
JavaScript
// ...import statements and component setup...

const handleProjectUpload = async () => {
  try {
    const file = await DocumentPicker.pick({
      type: [DocumentPicker.types.allFiles],
    });
    setProjectUpload(file);
  } catch (error) {
    console.error('Error selecting file:', error);
    // Handle any error logic here
    return;
  }
};

const handleDownloadStudent = () => {
  if (project.projectupload && project.projectupload.fileUrl) {
    Linking.openURL(project.projectupload.fileUrl);
  } else {
    setModalTitle('File Not Available');
    setModalMessage('The student report is not available for
download. Check back later.');
  }
};
```

```

        setModalButtonText('OK');
        setModalVisible(true);
    }
};

// ...other component code...

/* Project Upload Button */
{project.projectstatus === 'Accepted' && project.studentassigned ===
'Assigned' && project.moderatorassigned === 'Assigned' &&
project.numofstudents === 1 && (
    <View style={{ alignItems: 'center', marginBottom: 20 }}>
        <TouchableOpacity
            style={[styles.button, styles.stuButton]}
            onPress={handleProjectUpload}
        >
            <Text style={styles.buttonText}>Upload Student Report</Text>
        </TouchableOpacity>
        {project.projectupload && (
            <Text style={{ textAlign: 'center', fontSize: 14, color:
                '#212529' }}>{project.projectupload.fileName}</Text>
        )}
    </View>
)}

/* Download Student Report */
{project.numofstudents === 1 && (
    <View style={{ alignItems: 'center', marginBottom: 20 }}>
        <TouchableOpacity
            style={[styles.button, styles.stuButton]}
            onPress={handleDownloadStudent}
        >
            <Text style={styles.buttonText}>Download Student Report</Text>
        </TouchableOpacity>
        {project.projectupload && (
            <Text style={{ textAlign: 'center', fontSize: 14, color:
                '#212529' }}>{project.projectupload.fileName}</Text>
        )}
    </View>
)}

// ...export statement...

```

### **5.3.6 Meeting Creation Algorithm**

The Meeting Creation Algorithm is responsible for creating meetings within the app.

It outlines the steps and logic required to create a meeting with the provided data.

Algorithm Description:

1. Initialize state variables for fetched projects, project ID, fetched students, student ID, meeting date, meeting time, date picker visibility, and time picker visibility.
2. Access the user's authentication state from the AuthContext.
3. Use the useEffect hook to fetch the projects and filtered students based on the selected project ID when the project ID changes.
4. Define handleSubmit function to handle the submission of the meeting creation.
5. Validate that all the required fields are selected; if any field is missing, show an error modal with the appropriate message.
6. Get the supervisor ID from the user's authentication state.
7. Send a POST request to the server to create the meeting with the provided data.
8. If the response is successful, show a success modal with the appropriate message.
9. Handle any errors that occur during the meeting creation.

Sample Code:

The following code snippet demonstrates the implementation of the meeting creation algorithm:

```
JavaScript
import React, { useEffect, useState, useContext } from 'react';
// Other imports...

function Meet_CreateScreen({ navigation }) {
    // State variables...

    // Functions...

    const handleSubmit = async () => {
        // Code for meeting creation...

        try {
            // Validation...

            const supervisorId = user?.accid; // Get the accId from the authState

            const response = await fetch('http://3.26.23.172/meetings', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({
                    projId,
                    stuId,
                    meetDate: formattedDate,
                    meetTime: formattedTime,
                    supervisorId,
                }),
            });

            if (response.ok) {
                setModalTitle('Success!');
                setModalMessage('Meeting created');
                setModalButtonText('OK');
                setModalVisible(true);
            } else {
                console.error('Error creating meeting:', response.statusText);
                // Handle any error logic here
            }
        } catch (error) {
            console.error('Error creating meeting:', error);
        }
    };
}
```

```

        // Handle any error logic here
    }
};

// Render UI components...
}

export default Meet_CreateScreen;

```

### **5.3.7 Meeting Update Algorithm**

The Meeting Update Algorithm is responsible for updating meetings within the app.

It outlines the steps and logic required to create a meeting with the provided data.

Algorithm Description:

1. Retrieve the initial meeting details for the selected week:
  - Initialize state variables for initialMeetingMode, initialWorkDone, initialWorkToBeDone, initialComments, and meetStatus.
  - Fetch the initial meeting details from the API using the meetId and week.
  - Set the state variables with the fetched data.
2. Render the screen with the initial meeting details:
  - Create a functional component named SUMeetWeekInputScreen.
  - Extract the route and navigation parameters using destructuring.
  - Define state variables for meetingMode, workDone, workToBeDone, comments, and isModalVisible.
  - Implement the handleMeetingModeChange function to update the meetingMode state.

- Use the `useEffect` hook to set the header options and fetch the initial meeting details when the component mounts.
- Use the `useState` hook to initialize the state variables with the initial meeting details.
- Render the screen components, including `TextInput` and `Picker` components for input fields, and a modal for error messages.
- Implement the `handleSaveInput` function to handle saving the updated meeting details:
- Validate that all required fields (`meetingMode`, `workDone`, `workToBeDone`, and `comments`) are filled in.
- Create an object with the input values.
- Make a POST request to the API endpoint to save the updated meeting details.
- Display a success message in a modal if the request is successful, or handle any errors.

Sample Code:

The following code snippet demonstrates the implementation of the meeting update algorithm:

```
JavaScript
export default function SUMeetWeekInputScreen({ route, navigation })
{
  const { week, meetId, studentName } = route.params;

  const [initialMeetingMode, setInitialMeetingMode] = useState('');
  const [initialWorkDone, setInitialWorkDone] = useState('');
  const [initialWorkToBeDone, setInitialWorkToBeDone] = useState('');
```

```

const [initialComments, setInitialComments] = useState('');
const [meetStatus, setMeetStatus] = useState('');

// ...

useEffect(() => {
  navigation.setOptions({
    // ...
  });

  fetch(`http://3.26.23.172/meeting_details/${meetId}/${week}`)
    .then(response => response.json())
    .then(data => {
      setInitialMeetingMode(data.meetingmode);
      setInitialWorkDone(data.workdone);
      setInitialWorkToBeDone(data.worktobedone);
      setInitialComments(data.comments);
      setMeetStatus(data.meetstatus);
    })
    .catch(error => {
      console.error('Error fetching meeting details:', error);
    });
}, []);

function handleSaveInput() {
  if (!meetingMode || !workDone || !workToBeDone || !comments) {
    setModalTitle('Error');
    setModalMessage('Please ensure that you fill in and select all the required fields');
    setModalButtonText('Try again');
    setModalVisible(true);
    return;
  }

  const meetingDetails = {
    week: week,
    meetId: meetId,
    meetingMode: meetingMode,
    workDone: workDone,
    workToBeDone: workToBeDone,
    comments: comments
  };

  fetch('http://3.26.23.172/meeting_details', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
  });
}

```

```

        body: JSON.stringify(meetingDetails)
    })
    .then(response => {
        if (response.ok) {
            setModalTitle('Success!');
            setModalMessage('Meeting details updated');
            setModalButtonText('OK');
            setModalVisible(true);
        } else {
            console.error('Error saving meeting details');
        }
    })
    .catch(error => {
        console.error('Error saving meeting details:', error);
    });
}

// ...

return (
    <SafeAreaView style={styles.container}>
        {/* ... */}
    </SafeAreaView>
);
}

```

### **5.3.8 Presentation Creation Algorithm**

The Presentation Creation Algorithm is responsible for creating presentations within the app. It outlines the steps and logic required to create a meeting with the provided data.

Algorithm Description:

1. Retrieve the necessary data for creating a presentation, such as the presentation type, specialization, venue, date, and time.
2. Validate the input data to ensure it meets the required criteria.
3. If the input data is valid, proceed with the creation process.

4. Make an HTTP POST request to the server's API endpoint for creating presentations.
5. Set the appropriate headers for the request, including the "Content-Type" header with the value "application/json".
6. Create a JSON payload with the presentation data, including the presentation type, specialization, venue, date, and time.
7. Send the JSON payload as the request body.
8. Handle the server response:
  - a. If the response is successful (status code 200), display a success message to the user and take any necessary actions.
  - b. If the response indicates an error, display an error message to the user and take any necessary actions.
9. The presentation creation process is complete.

#### Sample Code:

The following code snippet demonstrates the implementation of the presentation creation algorithm:

```
JavaScript
// ...import statements and component setup...

function Pres_CreateScreen({ navigation }) {
  // ...state variables and event handlers...

  const handleSubmit = async () => {
    // ...data formatting and validation...

    try {
```

```

    // Send a POST request to create the presentation
    const response = await
fetch('http://3.26.23.172/presentations', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify({
        presType,
        presSpecialization,
        presVenue,
        presDate: formattedDate,
        presTime: formattedTime,
    }),
});

if (response.ok) {
    // Show success modal
    setModalTitle('Success!');
    setModalMessage('Presentation created');
    setModalButtonText('OK');
    setModalVisible(true);
} else {
    console.error('Error creating presentation:',
    response.statusText);
}
} catch (error) {
    console.error('Error creating presentation:', error);
}
};

return (
    // ...JSX components and UI layout...
);
}

export default Pres_CreateScreen;

```

## ***5.4 Sample Outputs***

Sample Outputs showcase a collection of visual representations that demonstrate the key screens of the application. These samples provide a practical understanding of the application's user interface design, layout, and features. By exploring these examples, it will serve as a valuable reference to acknowledge the application's functionality and the user experience it offers.

### ***5.4.1 Login Screen***

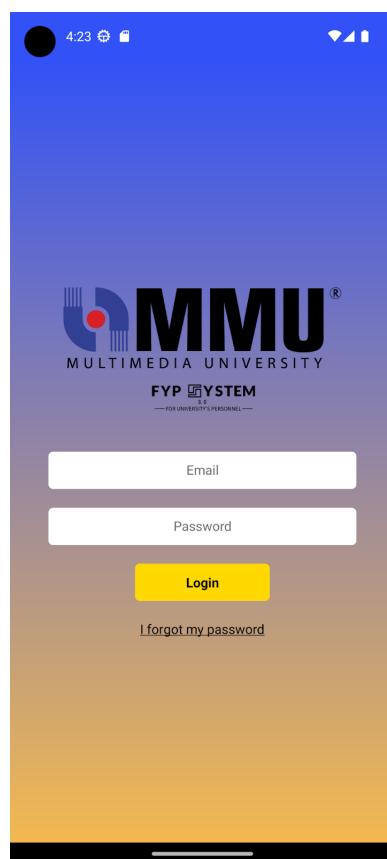


Figure 5.22 Login Screen

The Login Screen is the entry point for users to access the application. It features an intuitive interface with email and password input fields, and a login button. Users can

securely authenticate themselves, and upon successful login, they are directed to their respective main screens based on their account type. In case of an error or invalid login attempt, a modal is displayed with a corresponding message. The Login Screen ensures secure access to the application's features and functionalities.

#### **5.4.2 Main Screen**

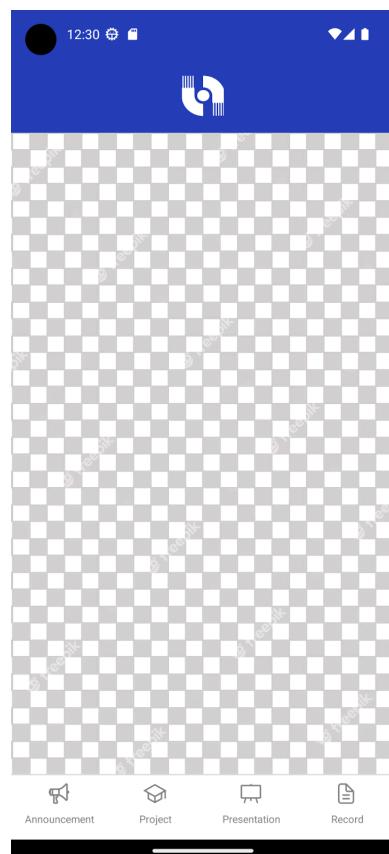


Figure 5.23 Faculty Manager,  
Coordinator, Moderator Main Container

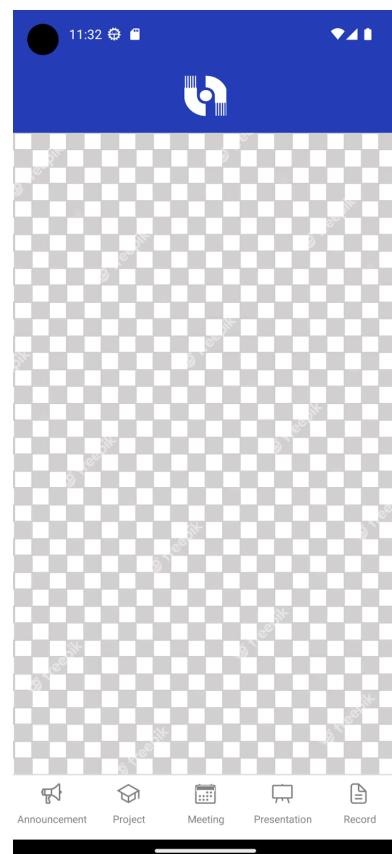


Figure 5.24 Supervisor Main Container

The main screen comprises five interconnected screens: the Announcement screen, Project screen, Meeting screen, Presentation screen, and Records screen. These screens are seamlessly integrated within the main container, allowing easy navigation through the bottom tab navigator. The tab navigator icon highlights in blue to

indicate the currently selected screen, providing a seamless and user-friendly navigation experience within the main container. The Announcement screen serves as the default home screen, and users can return to it by clicking the "MMU" logo located at the top center of the screen. It's important to note that the Meeting screen is exclusively accessible and navigable for users with a Supervisor account type, providing them with specialized functionalities and features.

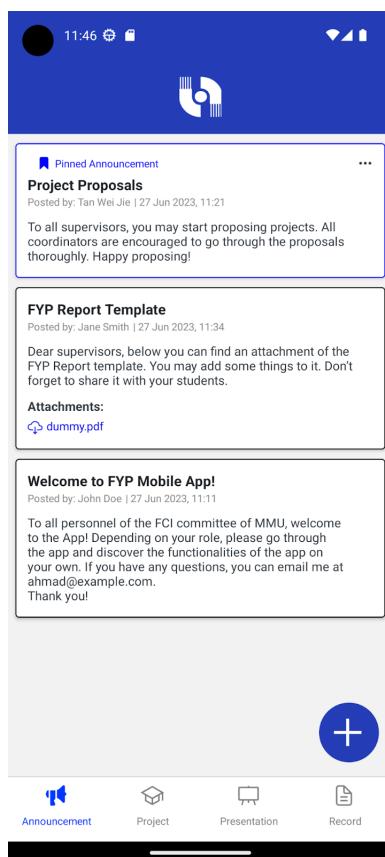


Figure 5.25 Announcement Screen

The announcement screen displays a list of announcements in card format, accessible to all users. Each announcement card encompasses essential details such as the announcement title, the name of the poster, the date and time it was posted, the announcement description, and the option to download any attached files, if

available. Announcements are arranged chronologically, with pinned announcements appearing at the top. The pinned announcement is indicated by a top text and a border is highlighted in blue. If the user has a faculty manager or coordinator account type, they will find a plus button at the bottom right corner. Clicking the plus button takes them to a separate screen where they can create a new announcement. The poster of the announcement can access a three-dot button on the top right of the card, providing options to pin, unpin, or delete the announcement.

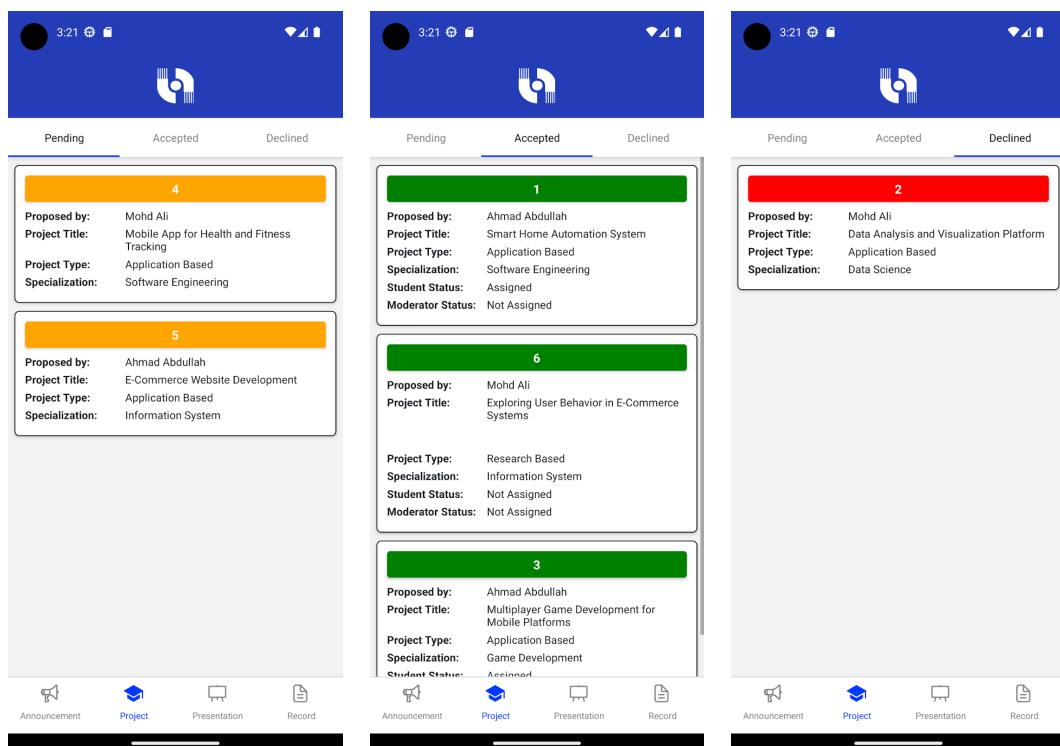


Figure 5.26 Project Screen

The project screen showcases projects through cards. Each project card displays important information including the project ID, the name of the proposer, the project type, and specialization. Additionally, if the project is approved, the card also indicates the student and moderator status. For supervisors, faculty managers, and coordinators, three tabs are available: "Pending", "Accepted", and "Declined". These

tabs organize projects based on their current status. Selecting a card leads to a detailed view of the chosen project. Faculty managers and coordinators have access to all projects, while supervisors can only view the projects they proposed. Moderators have a single tab labeled "Assigned Students", exclusively displaying projects assigned to them. Supervisors can create or propose new projects using the plus button located at the bottom right corner of the screen.

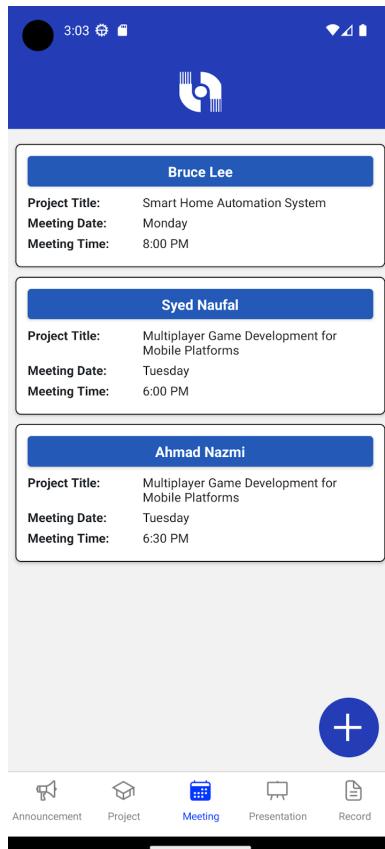


Figure 5.27 Meeting Screen

The meeting screen caters specifically to users with a Supervisor account type. It presents a list of meeting cards exclusively visible to the creator. Each meeting card presents essential details such as the student's name, the project title, the meeting

date, and the meeting time. A plus button is located at the bottom right corner, allowing users to create new meetings by navigating to a separate screen.

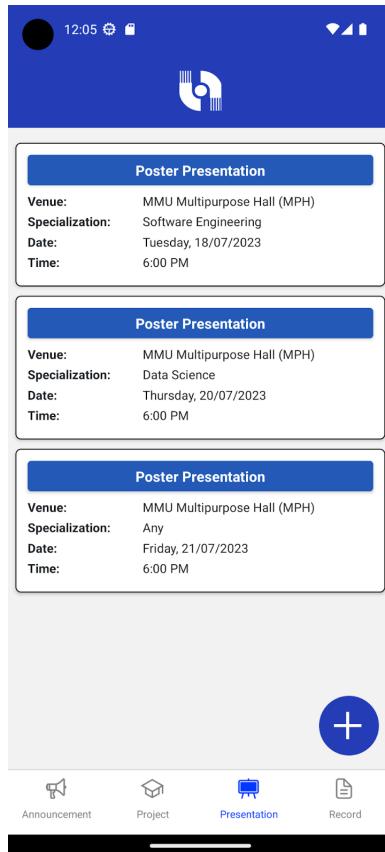


Figure 5.28 Presentation Screen

The presentation screen features a collection of presentation cards visible to all users. Users can browse and view these presentation schedules. Each presentation card displays key information including the presentation type, venue, specialization, date, and time. Faculty managers have access to a plus button at the bottom right corner, enabling them to create presentations. Clicking this button navigates them to a screen for creating presentations.

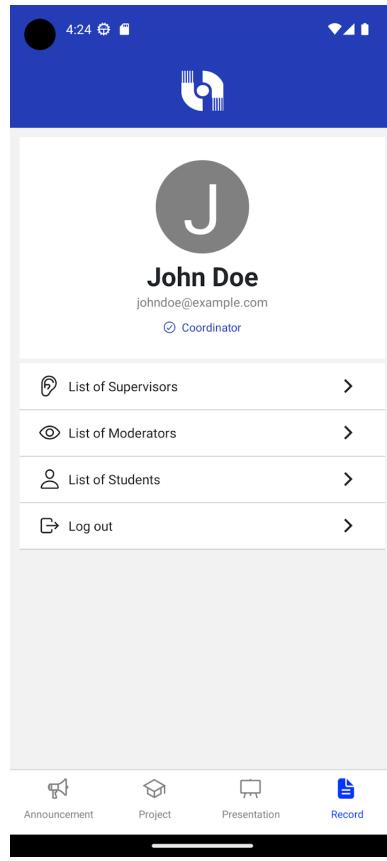


Figure 5.29 Records Screen

The records screen provides an overview of the user's information, including their name, email, and account type. Additionally, there is a clickable list featuring options such as "List of Supervisors", "List of Moderators", "List of Students", and "Log-out". By selecting "List of Supervisors," users are directed to a screen displaying a comprehensive list of registered supervisors. Similarly, choosing "List of Moderators" leads to a screen showcasing all registered moderators, while selecting "List of Students" presents a list of registered students. Finally, clicking on "Log-out" allows the user to securely log out of the system.

# Chapter 6: TESTING

## 6.1 Test Plan

The Test Plan includes a range of use cases, each designed to evaluate different functionalities of the system. The test criteria specified for each use case highlight the specific aspects that will be examined to ensure the system's reliability and accuracy. Additionally, the planned test dates indicate when these evaluations are scheduled to take place. The subsequent Test Plan table provides a comprehensive breakdown of each use case, its associated use case ID, use case name, test criteria, and the planned date for conducting the respective tests. For a more detailed perspective, refer to the Test Plan table 6.1.

Table 6.1 Test Plan

| Use Case ID | Use Case Name  | Test Criteria  | Test Date  |
|-------------|--|--|------------|
| <b>ALUC</b> |  |  |            |
| ALUC-01     | Login/Logout Module                                  | 1) Valid login if user is registered<br>2) Successful logout   | 16/06/2023 |
| ALUC-02     | View Announcement Module                             | 1) Announcements are displayed correctly<br>2) Sorting of announcements should be from newest to oldest<br>3) Pinned announcement should be at the top, ignoring the sorting order | 16/06/2023 |
| ALUC-03     | View Presentation Schedule Module                    | 1) Presentation schedules are displayed correctly  | 16/06/2023 |
| ALUC-04     | View Profile Info Module                             | 1) Profile information is displayed correctly.   | 16/06/2023 |
| ALUC-05     | View List of Students/Supervisors /Moderators Module | 1) List of students are displayed correctly<br>2) List of supervisors are displayed correctly<br>3) List of moderators are displayed correctly                                     | 16/06/2023 |

| <b>SUUC</b> |                                   |   |            |
|-------------|-----------------------------------|---|------------|
| SUUC-01     | Add New Project Module            | 1) Only accessible by role “Supervisor”<br>2) Successful submission only if the required inputs are met.<br>3) Project information is sent to the database.   | 16/06/2023 |
| SUUC-02     | View Projects (own) Module        | 1) Projects are displayed correctly based on their status<br>2) Project information is displayed correctly for the selected project<br>3) Project displayed based on the creator  | 16/06/2023 |
| SUUC-03     | Assign Students to Project Module | 1) “Assign Student” button displaying based on the number of student<br>2) Successful submission only if the required inputs are met.<br>3) Project and Student are updated in the database   | 16/06/2023 |
| SUUC-04     | Upload Project Report Module      | 1) “Upload Report” button displaying based on moderator assignment<br>2) Uploaded report is uploaded to the bucket and the uploaded report information is sent to the database  | 16/06/2023 |
| SUUC-05     | Create Meeting Module             | 1) Only accessible by role “Supervisor”<br>2) Successful submission only if the required inputs are met.<br>3) Project selection only available when project is approved<br>4) Student selection only available when student is assigned to a selected project<br>5) Meeting information is sent to the database. | 16/06/2023 |
| SUUC-06     | View Meetings Module              | 1) Only accessible by role “Supervisor”<br>2) Meeting displayed based on the creator  | 16/06/2023 |
| SUUC-07     | Update Meeting Module             | 1) Only accessible by role “Supervisor”<br>2) Meeting details is displayed correctly for the selected meeting<br>2) Successful submission only if the required inputs are met.<br>3) Meeting details information update based on selected meeting and week and is sent to the database.                           | 16/06/2023 |
| <b>FCUC</b> |                                   |   |            |
| FCUC-01     | Create Announcements Module       | 1) Only accessible by role “Faculty Manager” and “Coordinator”<br>2) Successful submission only if the required inputs are met.<br>3) Announcement information is sent to the database.   | 16/06/2023 |

|             |  |  |            |
|-------------|--|--|------------|
| FCUC-02     | Update Announcements Module            | 1) “Update Announcement” button displaying on the top right of an announcement card if user is the creator of the announcement<br>2) Able to pin or unpin if user is the creator of the announcement<br>3) Able to delete if user is the creator of the announcement<br>4) Update pinStatus in database<br>5) Delete the announcement entirely from the database | 16/06/2023 |
| FCUC-03     | View Projects Module                   | 1) Only accessible by role “Faculty Manager” and “Coordinator”<br>2) Projects are displayed correctly based on their status<br>2) Project information is displayed correctly for the selected project  | 16/06/2023 |
| <b>FMUC</b> |  |  |            |
| FMUC-01     | Setup Presentation Parameters Module   | 1) Only accessible by role “Faculty Manager”<br>2) Successful submission only if the required inputs are met.<br>3) Presentation information is sent to the database.  | 16/06/2023 |
| <b>COUC</b> |  |  |            |
| COUC-01     | Approve/Reject Project Module          | 1) “Approve” and “Reject” button displaying based on project status<br>2) Approved project will set the status of project to “Approved”<br>3) Rejected project will set the status of project to “Rejected”  | 16/06/2023 |
| COUC-02     | Assign Moderator to Project Module     | 1) “Assign Moderator” button displaying based on the student assignment<br>2) Successful submission only if the required inputs are met.<br>3) Project and Moderator status are updated in the database  | 16/06/2023 |
| <b>MOUC</b> |  |  |            |
| MOUC-01     | View Assigned Students’ Project Module | 1) Projects are displayed correctly based on their moderator/user assigned<br>2) Project information is displayed correctly for the selected project   | 16/06/2023 |
| MOUC-02     | Download Project Report Module         | 1) “Download Report” button displaying based on moderator assignment<br>2) Downloaded report is downloaded from the bucket   | 16/06/2023 |
| <b>ADUC</b> |  |  |            |
| ADUC-01     | Create Accounts Module                 | 1) Successful submission only if the required inputs are met.<br>2) Account information is sent to the database  | 16/06/2023 |

### **6.1.1 Test Data**

Table 6.2 is a table showcasing the test data and their corresponding details within the system. This test data provides an overview of the accounts, addresses, announcements, projects, meetings, meeting details, presentations, and students data involved in the system. Note that the information in this table is purely for testing purposes. Announcements (1-3), Projects (1-6), and Meeting Details (1-3) test data can be found in Appendix E (E - 6.1.1).

Table 6.2 Test Data

| <b>Accounts (1 - 4)</b>  |                     |                       |                   |                     |
|--------------------------|---------------------|-----------------------|-------------------|---------------------|
| accId                    | 1                   | 2                     | 3                 | 4                   |
| addId                    | 1                   | 2                     | 3                 | 4                   |
| name                     | John Doe            | Jane Smith            | Ahmad Abdullah    | Lim Wei Ling        |
| phone                    | +6012-3456789       | +6019-8765432         | +6016-1234567     | +6017-2345678       |
| email                    | johndoe@example.com | janesmith@example.com | ahmad@example.com | linglim@example.com |
| password                 | Password123         | P@ssw0rd456           | Test@123          | 12345678            |
| account_type             | coordinator         | faculty_manager       | supervisor        | moderator           |
| <b>Addresses (1 - 4)</b> |                     |                       |                   |                     |
| addId                    | 1                   | 2                     | 3                 | 4                   |
| street                   | 123 Jalan ABC       | 456 Jalan XYZ         | 789 Jalan DEF     | 567 Jalan MNO       |
| city                     | Kuala Lumpur        | Petaling Jaya         | Penang            | George Town         |
| state                    | Wilayah Persekutuan | Selangor              | Penang            | Penang              |
| postal_code              | 50000               | 47800                 | 10000             | 10400               |
| country                  | Malaysia            | Malaysia              | Malaysia          | Malaysia            |
| <b>Accounts (5 - 8)</b>  |                     |                       |                   |                     |

|                              |                          |                          |                          |                      |
|------------------------------|--------------------------|--------------------------|--------------------------|----------------------|
| accId                        | 5                        | 6                        | 7                        | 8                    |
| addId                        | 5                        | 6                        | 7                        | 8                    |
| name                         | Tan Wei Jie              | Nurul Huda               | Mohd Ali                 | Sarah Tan            |
| phone                        | +6012-9876543            | +6016-3456789            | +6019-8765432            | +6018-9876543        |
| email                        | wjtan@example.com        | huda@example.com         | mohdali@example.com      | sarahtan@example.com |
| password                     | Passw0rd!                | Testing#123              | 98765432                 | SecurePass!12        |
| account_type                 | coordinator              | faculty_manager          | supervisor               | moderator            |
| <b>Addresses (5 - 8)</b>     |                          |                          |                          |                      |
| addId                        | 5                        | 6                        | 7                        | 8                    |
| street                       | 890 Jalan PQR            | 1234 Jalan STU           | 5678 Jalan VWX           | 234 Jalan GHI        |
| city                         | Malacca City             | Kuching                  | Kota Kinabalu            | Johor Bahru          |
| state                        | Malacca                  | Sarawak                  | Sabah                    | Johor                |
| postal_code                  | 75000                    | 93000                    | 88000                    | 80000                |
| country                      | Malaysia                 | Malaysia                 | Malaysia                 | Malaysia             |
| <b>Meetings (1 - 3)</b>      |                          |                          |                          |                      |
| meetId                       | 1                        | 2                        | 3                        |                      |
| supervisor_id                | 3                        | 3                        | 3                        |                      |
| projId                       | 1                        | 3                        | 3                        |                      |
| stuId                        | 1                        | 2                        | 3                        |                      |
| meetDate                     | 2023-04-03T00:00:00.000Z | 2023-04-04T00:00:00.000Z | 2023-04-04T00:00:00.000Z |                      |
| meetTime                     | 12:00                    | 10:00                    | 10:30                    |                      |
| <b>Presentations (1 - 3)</b> |                          |                          |                          |                      |
| presId                       | 1                        | 2                        | 3                        |                      |
| presType                     | Poster Presentation      | Poster Presentation      | Poster Presentation      |                      |

|                         |                                |                                |                                |
|-------------------------|--------------------------------|--------------------------------|--------------------------------|
| presVenue               | MMU Multipurpose Hall<br>(MPH) | MMU Multipurpose Hall<br>(MPH) | MMU Multipurpose Hall<br>(MPH) |
| presDate                | 2023-07-18T00:00:00.000<br>Z   | 2023-07-20T00:00:00.000<br>Z   | 2023-07-21T00:00:00.000<br>Z   |
| presTime                | 10:00                          | 10:00                          | 10:00                          |
| presSpecialization      | Software Engineering           | Data Science                   | Any                            |
| <b>Students (1 - 3)</b> |                                |                                |                                |
| stuId                   | 1                              | 2                              | 3                              |
| name                    | Bruce Lee                      | Syed Naufal                    | Ahmad Nazmi                    |
| id                      | 101                            | 110                            | 115                            |
| email                   | blee101@student.com            | naufal110@student.com          | nazmi115@student.com           |
| projId                  | 1                              | 3                              | 3                              |

## **6.2 Test Results**

The system underwent comprehensive testing, covering all use cases. To streamline the testing process, use cases were grouped based on their IDs (ALUC, SUUC, FCUC, COUC, MOUC, and ADUC). Different types of tests, such as functionality, database, and acceptance tests, were performed based on the specific requirements of each use case. Test inputs were derived from the test data, and the results, sample outputs, and additional remarks were documented. The test case table contains all the relevant information. Note that some test case tables can be found in Appendix E (E -6.2.1 to E - 6.2.6).

### **6.2.1 Test Case - ALUC**

Table 6.3 Test Case - ALUC-01

| Use Case ID           | ALUC-01                                   | Use Case Name                          | Login/Logout Module |
|-----------------------|---|--|---------------------|
| Test Case Description | Valid login of registered users           |  |                     |
| Test Type             | Functionality and Database Test Case      |  |                     |
| Test Input            | Valid Input                               | Invalid Input                          |                     |
| Email                 | johndoe@example.com                       | johndoe@example.com    No input (NULL) |                     |
| Password              | Password123                               | password    No input (NULL)            |                     |
| Result                |   |  |                     |
| Expected Test Result  | Login successful. Navigate to home screen | Login unsuccessful                     |                     |
| Actual Result         | Login successful. Navigate to home screen | Login unsuccessful                     |                     |
| Status                | Pass                                      | Pass                                   |                     |

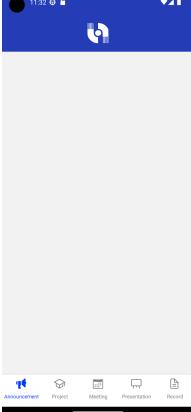
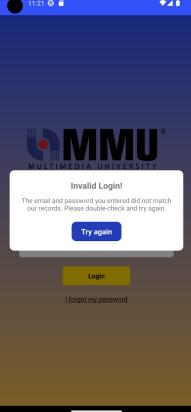
|               |   |   |
|---------------|---|---|
| Sample Output |                        |  |
| Remarks       | Successful login will set the session as active. Depending on the user role, functionalities may differ | Wrong email and/or wrong password and/or no input                                   |

Table 6.5 Test Case - ALUC-02-03-04-05

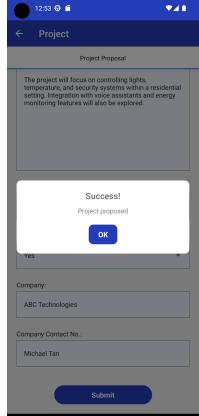
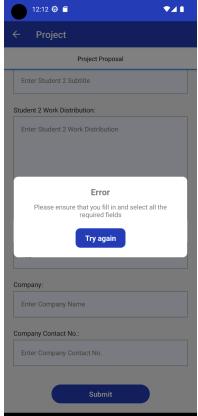
|                       |   |                                 |  |                                  |  |  |
|-----------------------|---|---------------------------------|--|----------------------------------|--|--|
| Use Case ID           | ALUC-02<br>ALUC-03<br>ALUC-04<br>ALUC-05  | Use Case Name                   | View Announcement Module<br>View Presentation Schedule Module<br>View Profile Info Module<br>View List of Students/Supervisors/Moderators Module |                                  |  |  |
| Test Case Description | Clicking on “Announcement” button on the bottom navigation bar will navigate to the “Announcement screen”<br>Clicking on “Presentation” button on the bottom navigation bar will navigate to the “Presentation screen”<br>Clicking on “Records” button on the bottom navigation bar will navigate to the “Records screen”<br>Clicking on “List of Supervisors” button at the “Records screen” will navigate to the “List of Supervisors screen”<br>Clicking on “List of Moderators” button at the “Records screen” will navigate to the “List of Moderators screen”<br>Clicking on “List of Students” button at the “Records screen” will navigate to the “List of Students screen” |                                 |  |                                  |  |  |
| Test Type             | Acceptance and Database Test Case   |                                 |  |                                  |  |  |
| Test Condition        | Click on “Announcement” button  | Click on “Presentation” button  | Click on “Records” button  | Click on “List” button           |  |  |
| Result                |   |                                 |  |                                  |  |  |
| Expected Test Result  | Navigate to announcement screen   | Navigate to presentation screen | Navigate to records screen   | Navigate to specific list screen |  |  |
| Actual Result         | Navigate to announcement screen   | Navigate to presentation screen | Navigate to records screen   | Navigate to specific list screen |  |  |
| Status                | Pass  | Pass                            | Pass   | Pass                             |  |  |

|               |  |  |   |  |
|---------------|--|--|---|--|
| Sample Output |  |  |   |  |
| Remarks       | Announcements are correctly displayed. Pinned announcement is at the top and the others are sorted from newest to oldest | Presentation schedules are displayed correctly | Profile information is displayed correctly based on the user. | <ul style="list-style-type: none"> <li>1) List of students are displayed correctly</li> <li>2) List of supervisors are displayed correctly</li> <li>3) List of moderators are displayed correctly</li> </ul> |

### 6.2.2 Test Case - SUUC

Table 6.7 Test Case - SUUC-01-2

| Use Case ID            | SUUC-01   | Use Case Name | Add New Project Module |
|------------------------|---|---------------|------------------------|
| Test Case Description  | Successful submission only if all required inputs are met. Project information is sent to the database if successful submission.  |               |                        |
| Test Type              | Functionality and Database Test Case  |               |                        |
| Test Input             | Valid Input   |               | Invalid Input          |
| Project Title          | Smart Home Automation System  |               | No input (NULL)        |
| Project Type           | Application Based   |               | No input (NULL)        |
| Project Specialization | Software Engineering  |               | No input (NULL)        |
| Project Focus          | Home Automation   |               | No input (NULL)        |
| Project Description    | The project aims to develop a smart home automation system using IoT technologies. It will enable users to control various home devices remotely and automate tasks for enhanced convenience and energy efficiency. |               | No input (NULL)        |
| Project Objectives     | 1. Design and implement a centralized   |               | No input (NULL)        |

|                        |   |   |
|------------------------|---|---|
|                        | control system, 2. Develop IoT device integration and communication protocols, 3. Create a user-friendly mobile application for remote control  |   |
| Project Scope          | The project will focus on controlling lights, temperature, and security systems within a residential setting. Integration with voice assistants and energy monitoring features will also be explored. | No input (NULL)   |
| Number of Students     | 1   | No input (NULL)   |
| Student 1 Subtitle     | NULL  | No input (NULL)   |
| Student 1 Work         | NULL  | No input (NULL)   |
| Student 2 Subtitle     | NULL  | No input (NULL)   |
| Student 2 Work         | NULL  | No input (NULL)   |
| Industry Collaboration | Yes   | No input (NULL)   |
| Company                | ABC Technologies  | No input (NULL)   |
| Company Contact No.    | Michael Tan (michael@example.com)   | No input (NULL)   |
| Result                 |   |   |
| Expected Test Result   | Project proposal submission successful. Send data to the server. Navigate to “Project screen”   | Project proposal submission unsuccessful. Data not sent to the server.                |
| Actual Result          | Project proposal submission successful. Send data to the server. Navigate to “Project screen”   | Project proposal submission unsuccessful. Data not sent to the server.                |
| Status                 | Pass  | Pass  |
| Sample Output          |    |  |

|         |                                     |                                      |
|---------|-------------------------------------|--------------------------------------|
| Remarks | Data is sent to the database server | Data not sent to the database server |
|---------|-------------------------------------|--------------------------------------|

Table 6.8 Test Case - SUUC-02

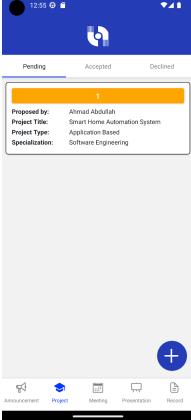
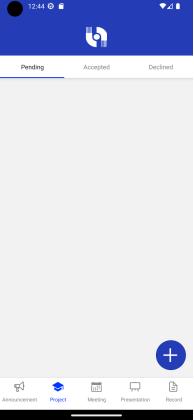
| Use Case ID           | SUUC-02   | Use Case Name   | View Project (own) Module                              |
|-----------------------|---|---|--|
| Test Case Description | Projects are displayed correctly based on their status. Project display based on the creator. Project information is displayed correctly for the selected project |   |  |
| Test Type             | Functionality and Database Test Case  |   |  |
| Test User             | Valid User  |   | Invalid User   |
| User Acc ID           | 3   |   | 7  |
| Result                |   |   |  |
| Expected Test Result  | Project displayed   |   | No project displayed                                   |
| Actual Result         | Project displayed   |   | No project displayed                                   |
| Status                | Pass  |   | Pass   |
| Sample Output         |    |  |  |
| Remarks               | Clicking on the project card navigates to the “Project Information screen” and displays the correct project information.  |   | No project found as the user never proposed a project. |

Table 6.9 Test Case - SUUC-03

| Use Case ID           | SUUC-03   | Use Case Name | Assign Student to Project Module |
|-----------------------|---|---------------|----------------------------------|
| Test Case Description | “Assign Student” button available (based on the number of students) in the “Project Info screen”. Navigate to the “Student Assignment screen” when the button is clicked. |               |                                  |
| Test Type             | Acceptance and Database Test Case   |               |                                  |
| Test Condition        | Click on a project card.  |               |                                  |

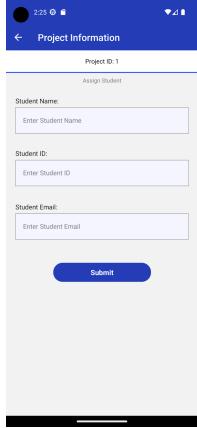
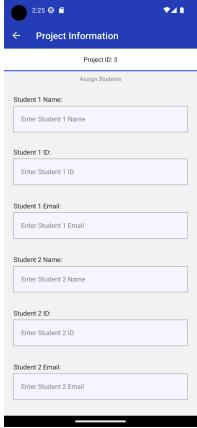
| Result               |   |  |
|----------------------|---|--|
| Expected Test Result | Navigate to “Student Assignment screen” when project’s numOfStudents = 1  | Navigate to “Two Students Assignment screen” when project’s numOfStudents = 2                                    |
| Actual Result        | Navigate to “Student Assignment screen” when project’s numOfStudents = 1  | Navigate to “Two Students Assignment screen” when project’s numOfStudents = 2                                    |
| Status               | Pass  | Pass   |
| Sample Output        |                               |                              |
| Remarks              | The “Assign Student” button only appears in the “Project Info screen” if the projectStatus is set to “Accepted” | The “Assign Students” button only appears in the “Project Info screen” if the projectStatus is set to “Accepted” |

Table 6.11 Test Case - SUUC-04

| Use Case ID           | SUUC-04  | Use Case Name | Upload Project Report Module |
|-----------------------|--|---------------|------------------------------|
| Test Case Description | “Upload Report” button available (based on moderator assignment) in the “Project Info screen”. Navigate to the file picker opens when the button is clicked. |               |                              |
| Test Type             | Acceptance and Database Test Case  |               |                              |
| Test Condition        | Click on “Upload Report” button  |               |                              |
| Result                |  |               |                              |
| Expected Test Result  | Opens file picker. On file select, file uploaded   |               |                              |
| Actual Result         | Opens file picker. On file select, file uploaded   |               |                              |
| Status                | Pass   |               |                              |

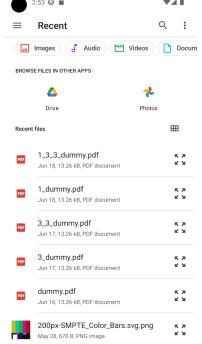
|               |  |
|---------------|--|
| Sample Output |  <p>The screenshot shows a file explorer interface with a sidebar for 'Recent' files. The main area lists several PDF documents and one PNG image. The PDFs are named '1.3.3_dummy.pdf', '1_dummy.pdf', '3.3_dummy.pdf', '3_dummy.pdf', and 'dummy.pdf', all created on June 18, 2023. The PNG image is '200px-SMPTE_Colour_Bars.svg.png' from May 20, 2023.</p> |
| Remarks       | Uploaded report is uploaded to the bucket and the uploaded report information is sent to the database.   |

Table 6.13 Test Case - SUUC-05

| Use Case ID           | SUUC-05  | Use Case Name | Create Meeting Module   |
|-----------------------|--|---------------|---|
| Test Case Description | Successful submission only if all required inputs are met. Project selection is only available when the project is approved. Student selection is only available when a student is assigned to the selected project. Meeting information is sent to the database if successful submission. |               |   |
| Test Type             | Functionality and Database Test Case   |               |   |
| Test Input            | Valid Input  |               | Invalid Input   |
| Project               | Smart Home Automation System   |               | No input (NULL)   |
| Student               | Bruce Lee  |               | No input (NULL)   |
| Meeting Date          | Mon Apr 03 2023  |               | No input (NULL)   |
| Meeting Time          | 12:00PM  |               | No input (NULL)   |
| <b>Result</b>         |  |               |   |
| Expected Test Result  | Meeting submission successful. Send data to the server. Navigate to "Meeting screen"   |               | Meeting submission unsuccessful. Data not sent to the server. |
| Actual Result         | Meeting submission successful. Send data to the server. Navigate to "Meeting screen"   |               | Meeting submission unsuccessful. Data not sent to the server. |
| Status                | Pass   |               | Pass  |

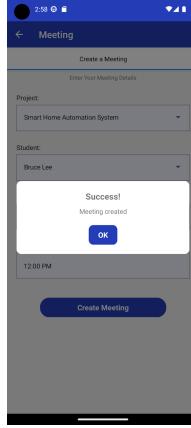
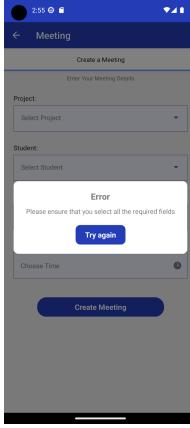
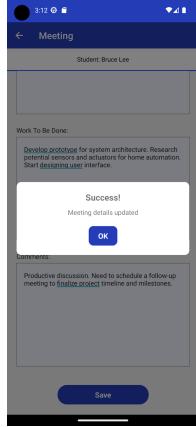
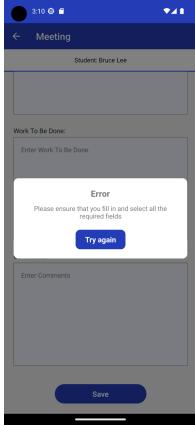
|               |  |   |
|---------------|--|---|
| Sample Output |                 |  |
| Remarks       | Project selection and student selection is working as intended. Data sent to the database server | Data not sent to the database server  |

Table 6.15 Test Case - SUUC-07

| Use Case ID           | SUUC-07  | Use Case Name | Update Meeting Module  |
|-----------------------|--|---------------|--|
| Test Case Description | Successful save only if all required inputs are met. Meeting details information is sent to the database if successful save.             |               |  |
| Test Type             | Functionality and Database Test Case   |               |  |
| Test Input            | Valid Input  |               | Invalid Input  |
| Meeting Mode          | Face to Face   |               |  |
| Work Done             | Discussed project requirements and goals. Assigned tasks. Conducted initial research on available automation technologies.               |               |  |
| Work To Be Done       | Develop prototype for system architecture. Research potential sensors and actuators for home automation. Start designing user interface. |               |  |
| Comments              | Productive discussion. Need to schedule a follow-up meeting to finalize project timeline and milestones                                  |               |  |
| Result                |  |               |  |
| Expected Test Result  | Meeting details saved successfully. Send data to the server. Navigate to “Meeting screen”  |               | Meeting details unsuccessfully saved. Data not sent to the server. |
| Actual Result         | Meeting details saved successfully. Send data to the server. Navigate to “Meeting screen”  |               | Meeting details unsuccessfully saved. Data not sent to the server. |

|               |   |   |
|---------------|---|---|
| Status        | Pass  | Pass  |
| Sample Output |      |  |
| Remarks       | Data sent to the database server. Meeting details are saved based on meetId and week. | Data not sent to the database server  |

### 6.2.3 Test Case - FCUC

Table 6.16 Test Case - FCUC-01

| Use Case ID           | FCUC-01   | Use Case Name   | Create Announcement Module   |  |  |  |
|-----------------------|---|---|--|--|--|--|
| Test Case Description | Only accessible by role “Faculty Manager” and “Coordinator”, thus only faculty managers and coordinators are allowed to create an announcement. |   |  |  |  |  |
| Test Type             | Acceptance Test Case  |   |  |  |  |  |
| Test Condition        | Logged in as the role “Faculty Manager”   | Logged in as the role “Coordinator”   | Logged in as the role “Supervisor”   | Logged in as the role “Moderator”  |  |  |
| Result                |   |   |  |  |  |  |
| Expected Test Result  | Navigate to “Create Announcement screen”  | Navigate to “Create Announcement screen”  | No navigation to “Create Announcement screen”  | No navigation to “Create Announcement screen”  |  |  |
| Actual Result         | Navigate to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen”   | Navigate to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” |  |  |
| Status                | Pass  | Pass  | Pass   | Pass   |  |  |

|               |  |  |   |   |
|---------------|--|--|---|---|
| Sample Output |  |  |   |   |
| Remarks       | “+” button appears in the “Announcement screen” to allow access. | “+” button appears in the “Announcement screen” to allow access. | No “+” button found in the “Announcement screen”. Access denied | No “+” button found in the “Announcement screen”. Access denied |

Table 6.19 Test Case - FCUC-02-2

| Use Case ID           | FCUC-02   | Use Case Name            | Update Announcement Module |
|-----------------------|---|--------------------------|----------------------------|
| Test Case Description | Able to pin or unpin if the user is the creator of the announcement. Able to delete if the user is the creator of the announcement. Update pinStatus in database. Delete the announcement entirely from the database. |                          |                            |
| Test Type             | Acceptance and Database Test Case   |                          |                            |
| Test Condition        | Click on “Pin” button   | Click on “Unpin” button  | Click on “Delete” button   |
| Result                |   |                          |                            |
| Expected Test Result  | Announcement is pinned  | Announcement is unpinned | Announcement is deleted    |
| Actual Result         | Announcement is pinned  | Announcement is unpinned | Announcement is deleted    |
| Status                | Pass  | Pass                     | Pass                       |

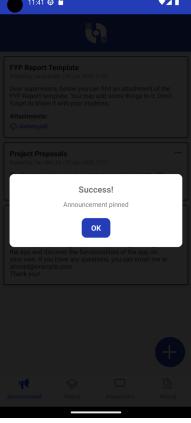
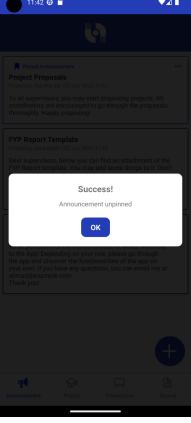
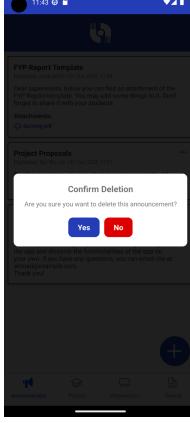
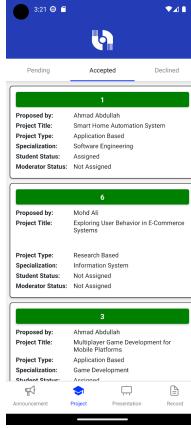
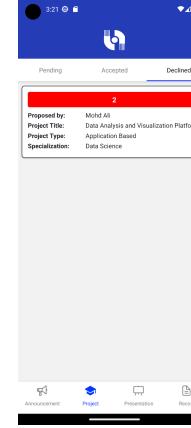
|               |   |  |   |
|---------------|---|--|---|
| Sample Output |  |  |  |
| Remarks       | Selected announcement pinStatus is updated to “Yes” through the database server   | Selected announcement pinStatus is updated to “No” through the database server     | Selected announcement is deleted from the database server                           |

Table 6.20 Test Case - FCUC-03

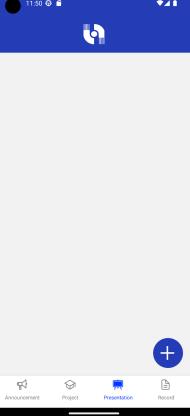
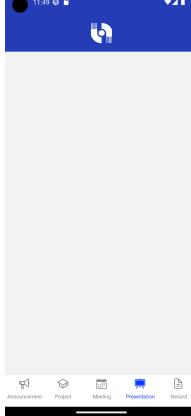
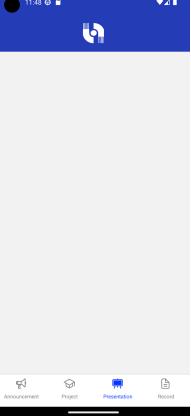
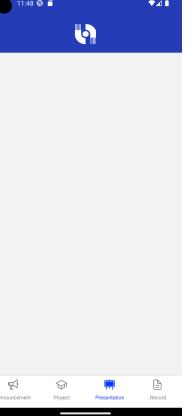
| Use Case ID           | FCUC-03  | Use Case Name                  | View Projects Module           |
|-----------------------|--|--------------------------------|--------------------------------|
| Test Case Description | Only accessible by role “Faculty Manager” and “Coordinator”. Projects are displayed correctly based on their status. Project information is displayed correctly for the selected project |                                |                                |
| Test Type             | Functionality and Database Test Case   |                                |                                |
| Test Condition        | Project Status: Pending  | Project Status: Approved       | Project Status: Rejected       |
| Result                |  |                                |                                |
| Expected Test Result  | All pending project displayed  | All approved project displayed | All rejected project displayed |
| Actual Result         | All pending project displayed  | All approved project displayed | All rejected project displayed |
| Status                | Pass   | Pass                           | Pass                           |

|               |   |  |   |
|---------------|---|--|---|
| Sample Output |    |    |    |
| Remarks       | <p>Clicking on the project card navigates to the “Project Information screen” and displays the correct project information. For the role “Coordinator” the buttons “Approve” and “Reject” will be available</p> | <p>Clicking on the project card navigates to the “Project Information screen” and displays the correct project information. For the role “Coordinator” a button “Assign Moderator” will be available</p> | <p>Clicking on the project card navigates to the “Project Information screen” and displays the correct project information.</p> |

#### 6.2.4 Test Case - FMUC

Table 6.21 Test Case - FMUC-01

| Use Case ID           | FMUC-01  | Use Case Name                                 | Setup Presentation Parameters Module          |   |   |  |  |
|-----------------------|--|---|---|---|---|--|--|
| Test Case Description | Only accessible by role “Faculty Manager”, thus only faculty managers are allowed to setup presentation parameters |   |   |   |   |  |  |
| Test Type             | Acceptance Test Case   |   |   |   |   |  |  |
| Test Condition        | Logged in as the role “Faculty Manager”  | Logged in as the role “Supervisor”            | Logged in as the role “Coordinator”           | Logged in as the role “Moderator”             |   |  |  |
| Result                |  |   |   |   |   |  |  |
| Expected Test Result  | Navigate to “Create Presentation screen”   | No navigation to “Create Presentation screen” | No navigation to “Create Presentation screen” | No navigation to “Create Presentation screen” | No navigation to “Create Presentation screen” |  |  |
| Actual Result         | Navigate to “Create Presentation screen”   | No navigation to “Create Presentation screen” | No navigation to “Create Presentation screen” | No navigation to “Create Presentation screen” | No navigation to “Create Presentation screen” |  |  |
| Status                | Pass   | Pass  | Pass  | Pass  | Pass  |  |  |

|               |   |   |  |   |
|---------------|---|---|--|---|
| Sample Output |  |  |  |  |
| Remarks       | "+" button appears in the "Presentation screen" to allow access.                  | No "+" button found in the "Presentation screen". Access denied                   | No "+" button found in the "Presentation screen". Access denied                    | No "+" button found in the "Presentation screen". Access denied                     |

### 6.2.5 Test Case - COUC

Table 6.23 Test Case - COUC-01

| Use Case ID           | COUC-01  | Use Case Name | Approve/Reject Project Module                  |
|-----------------------|--|---------------|--|
| Test Case Description | "Approve" and "Reject" button available (based on project status) in the "Project Info" screen. Clicking on either one will update the project status. |               |  |
| Test Type             | Acceptance and Database Test Case  |               |  |
| Test Condition        | Click on "Approve" button  |               | Click on "Reject" button                       |
| Result                |  |               |  |
| Expected Test Result  | Project accepted. Navigate to "Project screen"   |               | Project rejected. Navigate to "Project screen" |
| Actual Result         | Project accepted. Navigate to "Project screen"   |               | Project rejected. Navigate to "Project screen" |
| Status                | Pass   |               | Pass   |

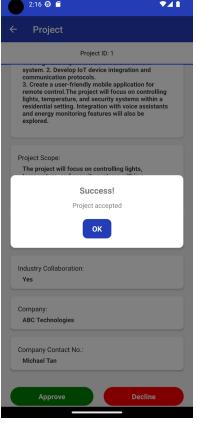
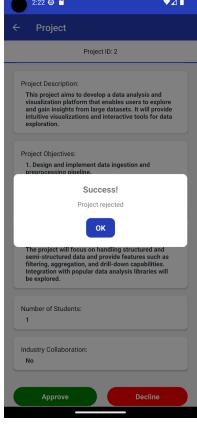
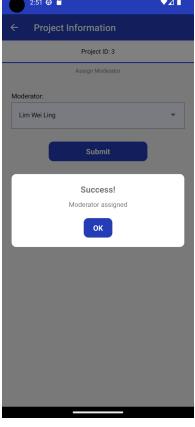
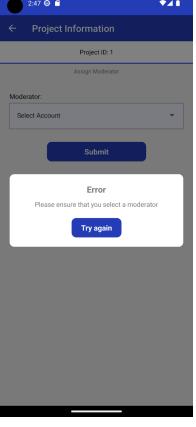
|               |   |   |
|---------------|---|---|
| Sample Output |    |    |
| Remarks       | Project status updated to “Approved” through database server. Buttons only available if project status is pending and disappears once the project status is not pending | Project status updated to “Rejected” through database server. Buttons only available if project status is pending and disappears once the project status is not pending |

Table 6.24 Test Case - COUC-02

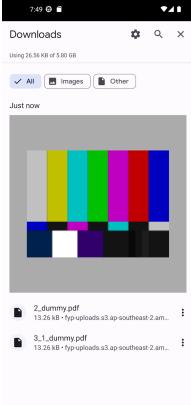
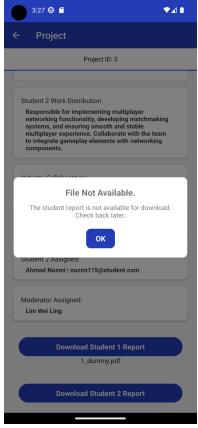
| Use Case ID           | COUC-02  | Use Case Name | Assign Moderator to Project Module  |
|-----------------------|--|---------------|---|
| Test Case Description | Successful submission only if the required inputs are met. Project and Moderator are updated in the database |               |   |
| Test Type             | Functionality and Database Test Case   |               |   |
| Test Input            | Valid Input  |               | Invalid Input   |
| Moderator             | Lim Wei Ling   |               | NULL    any uploads   |
| Result                |  |               |   |
| Expected Test Result  | Moderator assignment submission successful. Update data on the server. Navigate to “Project Info screen”     |               | Moderator assignment submission unsuccessful. Data not updated on the server. |
| Actual Result         | Moderator assignment submission successful. Update data on the server. Navigate to “Project Info screen”     |               | Moderator assignment submission unsuccessful. Data not updated on the server. |
| Status                | Pass   |               | Pass  |

|               |   |   |
|---------------|---|---|
| Sample Output |    |    |
| Remarks       | The “Assign Moderator” button only appears in the “Project Info screen” if the studentStatus is set to “Assigned”. On success, project database updated | The “Assign Moderator” button only appears in the “Project Info screen” if the studentStatus is set to “Assigned”. On failure, project database not updated |

### 6.2.6 Test Case - MOUC

Table 6.26 Test Case - MOUC-02

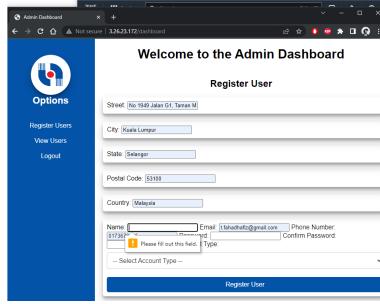
| Use Case ID           | MOUC-02  | Use Case Name    | Download Project Report Module |
|-----------------------|--|------------------|--------------------------------|
| Test Case Description | “Download Report” button available (based on moderator assignment) in the “Project Info screen”. Download file |                  |                                |
| Test Type             | Acceptance and Database Test Case  |                  |                                |
| Test Condition        | Click on “Download Report” button  |                  |                                |
| Result                |  |                  |                                |
| Test Input            | File uploaded  | No file upload   |                                |
| Expected Test Result  | File download  | No file download |                                |
| Actual Result         | File download  | No file download |                                |
| Status                | Pass   | Pass             |                                |

|               |   |   |
|---------------|---|---|
| Sample Output |  |  |
| Remarks       | File downloaded through link  | File not downloaded   |

### 6.2.7 Test Case - ADUC

Table 6.27 Test Case - ADUC-01

| Use Case ID           | SUUC-01  | Use Case Name | Add New Project Module                                  |
|-----------------------|--|---------------|---|
| Test Case Description | Successful submission only if all required inputs are met. Project information is sent to the database if successful submission. |               |   |
| Test Type             | Functionality and Database Test Case   |               |   |
| Test Input            | Valid Input  |               | Invalid Input   |
| Name                  | John Doe   |               | No input (NULL)   |
| Email                 | johndoe@example.com  |               | No input (NULL)   |
| Phone Number          | +6012-3456789  |               | No input (NULL)   |
| Account Type          | coordinator  |               | No input (NULL)   |
| Street                | 123 Jalan ABC  |               | No input (NULL)   |
| City                  | Kuala Lumpur   |               | No input (NULL)   |
| State                 | Wilayah Persekutuan  |               | No input (NULL)   |
| Postal Code           | 50000  |               | No input (NULL)   |
| Country               | Malaysia   |               | No input (NULL)   |
| Result                |  |               |   |
| Expected Test Result  | Account created successfully. Send data to the server.   |               | Fail to create an account. Data not sent to the server. |

|               |   |  |
|---------------|---|--|
| Actual Result | Account created successfully. Send data to the server.  | Fail to create an account. Data not sent to the server.  |
| Status        | Pass  | Pass   |
| Sample Output |  A screenshot of a web browser window titled "3.26.23.172/register_user". The page displays a "success" message. On the left, there is a sidebar with options: "Register Users", "View Users", and "Logout". |  A screenshot of the "Admin Dashboard" showing the "Register User" form. The form fields include: Street (No 1989 Jalan 51, Taman 10), City (Kuala Lumpur), State (Selangor), Postal Code (53100), Country (Malaysia). Below these are input fields for Name, Email (maznahata@gmail.com), Phone Number, and Confirm Password. A validation error message "Please fill out this field" is shown above the "Email" field. A dropdown menu for "Select Account Type" is visible. At the bottom is a blue "Register User" button. |
| Remarks       | Data is sent to the database server   | Data not sent to the database server   |

## **Chapter 7: CONCLUSION**

The "Mobile app for FYP" project has successfully revolutionized the management and monitoring of FYP projects, bringing significant improvements to the usability of the FYP System. Throughout the project, comprehensive research was conducted to understand the needs and requirements of FYP supervisors, faculty managers, moderators, and coordinators. Despite resource constraints and being a one-person team, remarkable progress was made in developing a functional prototype of the app.

An important achievement of the project was the identification and implementation of the cross-platform React Native framework, which greatly enhanced the efficiency and effectiveness of the FYP System. The focus on real-time communication between the backend and mobile app, utilizing WebSocket (JavaScript) hosting, ensured seamless and instantaneous interaction.

While the project encountered challenges, such as limited resources, significant strides were made in the prototype development. Moving forward, attention will be given to completing the prototype and enhancing the user interface and user experience. By making the app more user-friendly and intuitive for all stakeholders, the project aims to further elevate the management and monitoring of FYP projects.

In conclusion, the "Mobile app for FYP" project has successfully addressed the needs and requirements of FYP stakeholders, making remarkable progress despite resource constraints. The project has the potential to transform FYP management and monitoring practices, and its future focus on prototype completion and user experience improvement ensures a promising continuation of its positive impact.

## REFERENCES

- M.A. Mohamed (2017). An Implementation of Final Year Project Management System:World Applied Sciences Journal.
- Syila Izawana Ismail (2017). Online Project Evaluation and Supervision System. 2017 IEEE 15th Student Conference on Research and Development (SCoReD).
- Abdulkareem Ademola (2013). Design and Development of a University Portal for the Management of Final. International Journal Of Engineering And Computer Science.
- Marini Abu Bakar (2011). Final Year Supervision Management System as a Tool for. Elsevier.
- Wahab (2012). FINAL YEAR PROJECT ONLINE MANAGEMENT SYSTEM (FYPOS):: Perak: Universiti Teknologi Petronas.
- Affandi, R., & Omar, Z. (2018). Development of Mobile App (P\_Lab) for Enhancing the Student Understanding In Practical Work. International Research Journal of Electronics and Computer Engineering.
- Sunga, Y.-T., Changb, K.-E., & Liu, (2015). The Effects of Integrating Mobile Devices with Teaching and Learning on Students' Learning Performance: A Meta-Analysis and Research Synthesis. Computers & Education.
- Gaurav R. Sawarkar (2019). effectiveness of Mobile App as a Teaching and Learning Tool. Journal of Indian System of Medicine.
- Buddhini, Gaya R. & Chamali, Malinda C.B (2019). Development of mobile application through design-based research.
- Mobile Operating System Market Share Worldwide. (n.d.), from <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- Mobile Operating System Market Share United States of America. (2021, March). from <https://gs.statcounter.com/os-market-share/mobile/united-states-of-america>

Rakestraw, T. L., Eunni, R. V., & Kasuganti, R. R. (2013). The mobile apps industry: A case study. *Journal of Business Cases and Applications*.

Joorabchi, M. E., Mesbah, A., & Kruchten, P. (2013). Real Challenges in Mobile App Development. 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. Published. <https://doi.org/10.1109/esem.2013.9>

Jones, M. (2020). Native app development tools: Key features, pros, and cons. Retrieved from <https://buildfire.com/native-app-development-tools/>

Apple. (2022). Xcode. Retrieved from <https://developer.apple.com/xcode/>  
Google. (n.d.). Android Studio. Retrieved from <https://developer.android.com/studio>

Jones, C., & Smith, J. (2019). The Pros and Cons of Cross-Platform Mobile Development. *Journal of Mobile App Development & Management*.

React Native. (n.d.). Retrieved April 6, 2023, from <https://reactnative.dev/>

Pereira, L. S. (2018). Mobile Application Development: Challenges and Best Practices. In *Software Engineering for Mobile Devices* (pp. 17-31). Springer, Cham.

Smith, J. (2018). Node.js: An open-source JavaScript runtime environment for scalable app development. Retrieved from <https://www.example.com/nodejs-scalability>

Kaufmann, K. (2021). Fly.io: A Global Platform for Building Modern Cloud Applications. Retrieved from <https://thenewstack.io/fly-io-a-global-platform-for-building-modern-cloud-applications/>

Backblaze. (2021). B2 Cloud Storage. <https://www.backblaze.com/b2/cloud-storage.html>

AWS, Azure, and Google Cloud: A Comparison. (n.d.). Retrieved from  
<https://www.ibm.com/cloud/learn/aws-vs-azure-vs-google-cloud>

Software Testing - Test Case (n.d.). Retrieved from  
<https://www.geeksforgeeks.org/software-testing-test-case/>

## **APPENDIX A: COMMERCIALISATION PROPOSAL**

### **Final Year Project Management Application (For Committee Members) Commercialisation Proposal**

**Prepared by:**

TENGKU FAHAD HAFIZ BIN T. ABD NAWAWI  
Multimedia University  
Undergraduate  
+60 17 3678385  
[t.fahadhafize@gmail.com](mailto:t.fahadhafize@gmail.com)

## **Executive Summary**

The "Final Year Project Management Application (For Committee Members)" is a mobile application designed for Android and iOS platforms. The application aims to provide a comprehensive and efficient solution for University Personnel to manage the final year projects effectively. With its user-friendly interface and a range of essential functionalities, the application facilitates seamless communication, coordination, and monitoring of projects within the university environment.

The application caters to different user roles, including Supervisors, Faculty Managers, Moderators, Coordinators, and Administrators, each having specific access and privileges tailored to their responsibilities.

Overall, the "Final Year Project Management Application (For Committee Members)" streamlines and centralizes project management processes within the university, enhancing collaboration, efficiency, and organization. By leveraging technology, this application simplifies administrative tasks, fosters effective communication, and promotes transparency among committee members involved in the final year project management.

The objective of this document is to propose a commercialisation plan for this project which includes market analysis, problem statement, value proposition, mission, objectives, scope of work, business model, milestones, project outcome, and SWOT analysis.

## **Market Analysis**

The market analysis for the "Final Year Project Management Application (For Committee Members)" indicates a significant opportunity within the higher education sector. The management and coordination of final year projects in universities often involve complex processes and the need for effective communication among various stakeholders. The application aims to address these challenges and streamline project management tasks, making it an attractive solution for universities and educational institutions.

## **Target Customers**

The target customers for the "Final Year Project Management Application (For Committee Members)" are University Personnel involved in the management and oversight of final year projects. This includes Supervisors, Faculty Managers, Moderators, Coordinators, and Administrators. These individuals play crucial roles in ensuring the smooth execution of final year projects and require an efficient and streamlined solution to manage project-related tasks and communications. The application aims to cater to universities and educational institutions that have a structured final year project system in place.

## **Competitors**

In the market for project management tools and software, there are existing solutions that universities may adapt for their project management needs. However, these solutions often lack the specific features and functionalities required for the unique aspects of final year project management.

## **Problem Statement**

The current process of managing final year projects in universities is often manual, time-consuming, and prone to miscommunication. University Personnel, including Supervisors, Faculty Managers, Moderators, Coordinators, and Administrators, face challenges in effectively coordinating project-related tasks, overseeing progress, and facilitating seamless collaboration among committee members. There is a need for a comprehensive and user-friendly solution that addresses these issues and streamlines the management of final year projects.

## **Unique Value Proposition**

The "Final Year Project Management Application (For Committee Members)" is a unique and tailored solution that specifically caters to the needs of University Personnel involved in managing final year projects. It offers a centralized platform that enables efficient collaboration, communication, and task management, saving time and enhancing productivity. The application's user roles, customizable features, and mobile accessibility make it a comprehensive and user-friendly tool for optimizing the entire project management process.

The key advantages of the "Final Year Project Management Application (For Committee Members)" over potential competitors are:

1. Customization: The application is specifically designed to meet the requirements of final year project management in educational institutions, providing tailored features and workflows.
2. User Roles and Permissions: The application offers different user roles with specific access levels and privileges, ensuring efficient collaboration and control among committee members.
3. Project-centric Approach: The application focuses on the specific needs of final year projects, offering features such as project assignment, document uploads, meeting scheduling, and presentation management.
4. Centralized Communication: The application provides a centralized platform for communication, allowing committee members to share announcements, updates, and project-related information easily.
5. Mobile Accessibility: With support for both Android and iOS platforms, the application offers convenient access on mobile devices, enabling users to manage projects on-the-go.

## **Mission**

Our mission is to revolutionize the management of final year projects in universities by providing a specialized application that empowers University Personnel with the necessary tools to streamline project coordination, enhance communication, and ensure successful project outcomes.

## **Objectives**

- Simplify Project Management: Provide a user-friendly interface that simplifies the management of final year projects, allowing University Personnel to easily handle tasks, assignments, and communications within a centralized platform.
- Enhance Collaboration: Foster seamless collaboration among committee members by facilitating effective communication, document sharing, and progress tracking.
- Improve Efficiency: Streamline project-related processes, such as meeting scheduling, document uploads, and presentation management, to increase overall efficiency and reduce manual workloads.
- Promote Transparency: Enhance transparency and visibility into project status, progress, and updates, ensuring all committee members are well-informed and aligned.
- Ensure Project Success: Support University Personnel in achieving successful project outcomes by providing a comprehensive solution that optimizes project management and fosters effective teamwork.

## **Methods and Scope of Work**

To develop the "Final Year Project Management Application (For Committee Members)," we will employ a combination of modern technologies and frameworks. The frontend of the application will be built using React Native, a popular and efficient framework for developing cross-platform mobile applications. This will ensure that the application can run seamlessly on both Android and iOS devices, providing a consistent user experience.

The backend of the application will be developed using Node.js, a powerful and scalable runtime environment. Node.js will allow us to build a robust and efficient server-side application that can handle various functionalities, such as user authentication, data storage, and real-time communication.

For hosting and infrastructure, we will utilize Amazon Web Services (AWS) resources. The application will be deployed on an EC2 instance to ensure reliable and scalable performance. User files, such as project reports and attachments, will be stored securely in Amazon S3 (Simple Storage Service). Additionally, we will utilize PostgreSQL (psql) as the database management system for storing and retrieving project-related data.

The scope of work for the project includes:

- UI/UX Design: Creating an intuitive and visually appealing user interface design that ensures a seamless user experience.
- Frontend Development: Implementing the application's frontend using React Native, incorporating screens, navigation, and user interactions.
- Backend Development: Developing the backend logic using Node.js, including user authentication, data management, and real-time communication.
- Database Design and Management: Designing the database schema using PostgreSQL and implementing data storage and retrieval functionalities.
- Integration and Testing: Ensuring smooth integration of frontend and backend components and conducting rigorous testing to identify and resolve any issues.
- Deployment and Infrastructure Setup: Deploying the application on an EC2 instance, configuring S3 for file storage, and ensuring proper server setup and security measures.

## **Business Model**

The business model for the "Final Year Project Management Application (For Committee Members)" will be based on a subscription fee model. The application will offer different subscription tiers with varying features and capabilities, catering to the needs of different universities and their respective committee members.

Key aspects of the business model include:

- Subscription Tiers: Offering multiple subscription tiers, such as Basic, Pro, and Enterprise, with varying levels of functionality, storage capacity, and support options. This allows universities to choose the subscription plan that best fits their requirements.
- Monthly Subscription Fee: Charging a monthly fee for each subscription tier, providing universities with continuous access to the application's features and support services.
- Value-added Services: Offering additional value-added services, such as training, customization options, and priority support, for an additional fee. This allows universities to further tailor the application to their specific needs and receive personalized assistance.
- Scalability and Flexibility: Designing the business model to accommodate the growth and scalability of universities, allowing them to easily upgrade or downgrade their subscription plans based on their evolving requirements.
- Customer Support: Providing responsive and efficient customer support to address any technical issues or inquiries, ensuring a positive user experience and customer satisfaction.

By adopting a subscription fee model, we can establish a recurring revenue stream while providing ongoing support and continuous enhancements to meet the evolving needs of universities and their committee members.

## Milestones and Key Metrics

The milestone table outlines the key milestones and their corresponding durations for the "Final Year Project Management Application (For Committee Members)" project.

| Milestone                              | Duration          |
|--|-------------------|
| Project Plan                           | 1 week            |
| Background Study                       | 2 weeks           |
| Requirements                           | 3 weeks           |
| Design & Process Models                | 6 weeks           |
| Prototype                              | 4 weeks           |
| <b>Phase 1 Complete</b>                | <b>3.6 months</b> |
| Update Project Plan                    | 1 week            |
| System Design                          | 4 weeks           |
| Development                            | 12 weeks          |
| Testing                                | 2 weeks           |
| <b>Phase 2 Complete</b>                | <b>4.3 months</b> |
| Maintenance and Enhancement Activities | 1 week            |
| <b>Project End</b>                     | <b>8.3 Months</b> |

## Key Metrics

1. Timely completion of milestones within the specified durations.
2. Adherence to the project plan and schedule.
3. Successful development and implementation of project components such as the project plan, background study, requirements, design and process models, prototype, system updates, and testing.
4. Efficient collaboration and progress tracking among team members (supervisor, faculty manager, moderator, coordinator, and admin).
5. Effective testing and evaluation of the application, ensuring functionality, reliability, and user satisfaction.
6. Achievement of key project objectives and deliverables.

These key metrics will serve as performance indicators throughout the project's lifecycle, enabling the project team to monitor progress, identify areas for improvement, and ensure the successful execution of the "Final Year Project Management Application (For Committee Members)" project.

## Project Outcome

The ultimate outcome of the "Final Year Project Management Application (For Committee Members)" is to transform the way final year projects are managed in universities. By implementing this application, universities can expect improved project coordination, streamlined communication, and enhanced efficiency throughout the project lifecycle. The application aims to ensure successful project outcomes, reduce administrative burdens, and provide a seamless experience for all committee members involved.

## Technical Benefits

- Enhanced Project Management Efficiency: The application improves project management for university staff by enabling real-time communication and data sharing, leading to streamlined workflows, quicker decision-making, and better project oversight..
- Mobile Accessibility and Cross-Platform Compatibility: Leveraging React Native, the application ensures mobile accessibility for Android and iOS platforms, providing flexibility and convenience for committee members. It also reduces development efforts and maintenance costs while delivering a consistent user experience across devices.
- Secure Data Management: The implementation of Amazon S3 and PostgreSQL ensures secure and reliable data storage. It safeguards sensitive information from unauthorized access and ensures data integrity and efficient retrieval.

## Other Benefits

- Improved Collaboration and Decision-Making: The application enhances collaboration among committee members by facilitating effective idea sharing, feedback, and discussions. This leads to better-informed decisions and improved project outcomes.
- Transparent Project Tracking: Committee members have access to project status and progress, enabling transparent monitoring of project development stages. This transparency facilitates better project tracking and proactive measures to ensure project success.
- Time and Resource Savings: The application digitizes and automates project management processes, reducing manual efforts and paperwork. This results in

significant time and resource savings for committee members, allowing for more strategic allocation of efforts.

- Strengthened Accountability: Functionalities such as project assignment and report uploads promote accountability among committee members. Responsibilities and contributions are recorded, fostering a culture of responsibility and commitment to project excellence.

## SWOT Analysis

### Strengths:

- Tailored solution specifically designed for final year project management in universities.
- User-friendly interface and customizable features catering to the unique needs of University Personnel.
- Mobile accessibility, enabling on-the-go project management.
- Centralized platform for communication and collaboration among committee members.

### Weaknesses:

- Reliance on user adoption and acceptance within university settings.
- Potential learning curve for new users transitioning to the application.

### Opportunities:

- Growing demand for streamlined project management solutions in universities.
- Collaboration and partnership opportunities with universities and educational institutions.
- Scope for continuous improvement based on user feedback and emerging industry trends.

### Threats:

- Competition from generic project management tools and collaboration platforms.
- Resistance to change and adoption of new technologies within universities.
- Security and data privacy concerns in handling sensitive project information.

## APPENDIX B: MEETING LOGS (FYP1)



### TPT3101 Final Year Project (FYP1) Meeting Log Trimester 1, 2022/23 (Trimester ID:2210)

|  |   |
|--|---|
| Meeting Date:<br>30/09/2022  | Meeting No.:<br>1                                     |
| Meeting Mode:<br>Online  |   |
| Project ID:<br>2355  | Project Type:<br>Application-based                    |
| Project Title :<br>Mobile App for FYP Supervisors  |   |
| Student ID :<br>1151102898   | Student Name:<br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| Student Programme and Specialisation:<br>Bachelor of Computer Science (Software Engineering) |   |
| Supervisor Name:<br>Dr. Yeoh Eng Thiam   | Co-Supervisor Name:<br>-                              |
| Collaborating Company:<br>-  | Company Supervisor Name:<br>-                         |

**1. WORK DONE**

First meeting on guidelines on what to do for the next week. Getting ready for the first task of the Final Year Project 1.

**Tasks:** Problem Formulation and Project Planning /Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Listen and ready up.
- Clean up on possible misunderstandings.

**2. WORK TO BE DONE**

*Finish up on project planning (Gantt Chart).*

**Tasks:** Problem Formulation and Project Planning /Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Project planning (Milestones and Gantt Chart)
- Project Overview
- Project Objectives
- Goals of project

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

No problems encountered.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**  
No comments.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
**(if applicable)**

.....  
Company Supervisor's Signature  
**(if applicable)**

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>06/10/2022  | <b>Meeting No.:</b><br>2                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

Most part of the project planning is done including the introduction, milestones and gantt chart.

**Tasks:** Problem Formulation and Project Planning /~~Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion~~

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Project Overview
- Project Objectives
- Project Plan (Milestones and Gantt Chart)

**2. WORK TO BE DONE**

*Few touch ups on the Gantt Chart and start on the first part of Chapter 2:Background Study.*

**Tasks:** Problem Formulation and Project Planning /Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Update Gantt Chart
- Research/Background study on Existing System(s)
- Finish up Survey Questions

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

No problems encountered.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**  
Update Gantt Chart and start on background study.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>13/10/2022  | <b>Meeting No.:</b><br>3                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

50% of background study is done. This includes, the study of existing system, and literature review.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Research/Background study of existing systems
  - MMU FCI FYP System
  - UiTM FEE FYP System
  - UTP FYP System
  - UM FYP System
  - UNIMAS FYP System
- Literature review based on the studied existing system
- Survey Questionnaire for FCI FYP Supervisors

**2. WORK TO BE DONE**

Finish up on the remaining of the background study, including sending out the survey to FCI FYP supervisors and gaining the data.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Email finished survey to FCI FYP Supervisors
- Data entry of the survey answers
- Research/Background study on technological backgrounds
- Propose a solution on technological backgrounds

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

No problems encountered.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Continue working on existing systems and survey. Start on technology background.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>20/10/2022  | <b>Meeting No.:</b><br>4                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

90% of background study is done. This includes, the study of technological background, and proposed solution. Update of survey questions and gantt chart upon advised.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Update survey questions
- Improve readability of the Gantt Chart
- Research/Background study of technological background
  - Native Mobile App Development Tools
    - Xcode
    - Android Studio
  - Cross-Platform Mobile Development Tools
    - React Native
  - Backend
    - Node.js
    - Ruby on Rails
  - Host
    - Fly.io
    - Backblaze B2
    - WebSocket
- Proposed Solution

**2. WORK TO BE DONE**

Email survey questions to FCI FYP Supervisors and studying the data gained from the survey answers. Continue study on existing systems and review all of Chapter 2: Background Study. Start on the first part of Chapter 3: Requirements.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Email finished survey to FCI FYP Supervisors
- Data entry of the survey answers
- Review and update on existing systems
- Review and update on technological backgrounds
- Start on Chapter 3: Requirements
  - System Overview
  - System minimum requirements
  - Use Case Diagram

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

No problems encountered.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
**(if applicable)**

.....  
Company Supervisor's Signature  
**(if applicable)**

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  

This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>27/10/2022  | <b>Meeting No.:</b><br>5                                     |
| <b>Meeting Mode:</b><br>Online  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

98% of background study is done. This includes the study of problems to be solved, and updated study of existing systems. Update of survey questions and study of existing systems upon advised. Early start on Chapter 3 where system overview, system minimum requirements, and use case diagram is almost complete.

**Tasks:** ~~Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion~~

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Update survey questions
- Update Research/Background study of existing systems
  - BINUS Mobile for Lecturer
  - Sister for Lecturer UNEJ
- Update Research/Background study of problems to be solved
- Research/Study of Requirements
  - System Overview
  - System minimum requirements
  - Use Case Diagram

**2. WORK TO BE DONE**

Email survey questions to FCI FYP Supervisors and studying the data gained from the survey answers. Continue Chapter 3: Requirements.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework /Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Email finished survey to FCI FYP Supervisors
- Data entry of the survey answers
- Continue on Chapter 3: Requirements
  - Update System Overview description
  - Update Use Case Diagram
  - Use Case Diagram Modules
  - Class Diagram

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

No problems encountered.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Background Study - mostly completed, to do data collection for survey  
Requirements - continue working on System Overview

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>10/11/2022  | <b>Meeting No.:</b><br>6                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

100% of background study is done after finishing up survey analysis and touching up problems to be solved. Survey was sent on Early start on 28th October 2022 and ended in 4th November 2022. New data discovered on the survey analysis, hence big changes on Chapter 3. Use case diagram updated and explanation of each modules/use cases was done.

**Tasks:** ~~Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion~~

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Survey questions sent out
- Updated survey data/analysis
- Update Research/Background study of problems to be solved
- Research/Study of Requirements
  - Use Case Diagram describing the FYP Mobile Application
  - Use Case Diagram for actor Lecturer/Supervisor
  - Use Case Diagram for actor Faculty Manager
  - Use Case Diagram for actor Admin
  - Explanation of each module/use cases in table form

**2. WORK TO BE DONE**

If use case diagrams are all good, start on class diagram and a bit on Chapter 4. If not, update of use cases required.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Start on Class Diagram
- Start on Chapter 4
- Update Use Case Diagram (if required)

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

Had a hard time figuring out what modules and features to choose and do. Survey analysis helped in deciding out. Decided to scrap out student functionalities entirely and focus on lecturer/supervisor and faculty managers functionalities.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Continue working Chap 3 Requirements. Start on ER diagram. Survey - write brief summary if possible

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>14/12/2022  | <b>Meeting No.:</b><br>7                                     |
| <b>Meeting Mode:</b><br>Online  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

Use case diagram updated but the table explanation is still incomplete. Screen design is now in early development.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Updated Use Case Diagram
  - Use Case Diagram describing the FYP Mobile Application
  - Use Case Diagram for actor Lecturer/Supervisor
  - Use Case Diagram for actor Faculty Manager
  - Use Case Diagram for actor Moderator
  - Use Case Diagram for actor Coordinator
  - Explanation of each module/use cases in table form
- Screen Design
  - Login Screen
  - Announcement/Home screen for Lecturer/Supervisor
  - Announcement/Home screen for Faculty Manager

**2. WORK TO BE DONE**

Finish up on use case diagrams and screen design.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework /~~Design or Research Methodology~~ / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Finish Use Case Diagram
- Finish Screen Design

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

Cannot imagine how the app and its functionalites/modules of each actor, so I cannot do the data tables and ER diagram completely. I think I will need to do screen designs first to get a good look of how the flow of the functionalities will go.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

-

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
**(if applicable)**

.....  
Company Supervisor's Signature  
**(if applicable)**

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP1) Meeting Log**  
**Trimester 1, 2022/23 (Trimester ID:2210)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>01/12/2022  | <b>Meeting No.:</b><br>8                                     |
| <b>Meeting Mode:</b><br>Online  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

Use case diagram updated but the table explanation is still incomplete. Class Diagram and explanation completed. Software architecture and explanation is completed. First draft of report is ready.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Updated Use Case Diagram
  - Use Case Diagram describing the FYP Mobile Application
  - Use Case Diagram for actor Supervisor
  - Use Case Diagram for actor Faculty Manager
  - Use Case Diagram for actor Moderator
  - Use Case Diagram for actor Coordinator
  - Use Case Diagram for actor Administrator
  - Explanation of each module/use cases in table form
- Class Diagram
  - Diagram
  - Explanation
- Software Architecture
  - Diagram
  - Explanation
- First Draft of report is completed

**2. WORK TO BE DONE**

Finish up on use case diagrams tables and ERD.

**Tasks:** Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Finish Use Case Diagram table explanations
- Finish ERD

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Update chapter 3 and 4. Start working on prototype

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

## APPENDIX C: MEETING LOGS (FYP2)



### TPT3101 Final Year Project (FYP2) Meeting Log Trimester 2, 2023 (Trimester ID:2220)

|  |   |
|--|---|
| Meeting Date:<br>31/03/2023  | Meeting No.:<br>1                                     |
| Meeting Mode:<br>Face to Face  |   |
| Project ID:<br>2355  | Project Type:<br>Application-based                    |
| Project Title :<br>Mobile App for FYP Supervisors  |   |
| Student ID :<br>1151102898   | Student Name:<br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| Student Programme and Specialisation:<br>Bachelor of Computer Science (Software Engineering) |   |
| Supervisor Name:<br>Dr. Yeoh Eng Thiam   | Co-Supervisor Name:<br>-                              |
| Collaborating Company:<br>-  | Company Supervisor Name:<br>-                         |

**1. WORK DONE**

Use case diagram table explanation completed. Sequence diagram completed. Data dictionary completed. ER Diagram completed. Appendix A/meeting logs updated in draft.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Project Plan (Milestones and Gantt Chart)

**2. WORK TO BE DONE**

Review and update documentation based on presentation feedback

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Update documentation

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**  
Submit Gantt chart by next week. Review Chap 2 & 3 by next week.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP2) Meeting Log**  
**Trimester 2, 2023 (Trimester ID:2220)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>07/04/2023  | <b>Meeting No.:</b><br>2                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

Milestones and Gantt Chart for Trimester 2 completed. Use Case Diagram and Use Case Diagrams Tables explanation updated. References fixed. Added two new subsections.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Project Plan
  - Updated Milestones
  - Updated Gantt Chart
- Use Case Diagram Tables Updated
  - Supervisor Module
  - Faculty Manager Module
  - Moderator Module
  - Coordinator Module
- Use Case Diagram Updated
- References Updated
- Added New Subsection
  - Survey Conclusion
  - AWS vs Azure vs Google Cloud

**2. WORK TO BE DONE**

Finish up on Use Case Diagram Table. Update Class and ER Diagram. Start on review and update of Chapter 4.

**Tasks:** Implementation / Testing (Application based projects) or Evaluation of Findings and Research Contribution (Research based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Finish Use Case Diagram Table
  - Supervisor
  - Coordinator
  - Moderator
  - Admin
- Update Class Diagram
- Update ER Diagram
- Review and Update Chapter 4:Design

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Gantt chart ok  
Chap 2 ok  
Finalise Chap 3 by next week

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  

This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP2) Meeting Log**  
**Trimester 2, 2023 (Trimester ID:2220)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>14/04/2023  | <b>Meeting No.:</b><br>3                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

System Overview updated to explain better. Use Case Diagram Tables completed. Class Diagram updated and summarized the explanation.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Updated System Overview
- Use Case Diagram Tables Updated
  - Supervisor Module
  - Moderator Module
  - Coordinator Module
  - Admin Module
- Class Diagram Updated
- Summarized Class Diagram Explanation

**2. WORK TO BE DONE**

Update on Chapter 4 which focuses on databases. Update Software Architecture, ER Diagram and Data Dictionary

**Tasks:** Implementation / Testing (Application based projects) or Evaluation of Findings and Research Contribution (Research based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Update Software Architecture
- Update ER Diagram
- Update Data Dictionary

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP2) Meeting Log**  
**Trimester 2, 2023 (Trimester ID:2220)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>06/05/2023  | <b>Meeting No.:</b><br>4                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

**1. WORK DONE**

ER Diagram and Data Dictionary updated and explained better. Software Architecture updated and explained better. Setup database (RDS postgres). Started development work.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Updated ERD
- Updated Data Dictionary
- Updated Software Architecture
- Summarized Software Architecture Explanation
- Setup database using Amazon RDS with PostgreSQL as the main engine.
- Started Development work
  - UI/UX for the role Supervisor 80% done
  - UI/UX for the role Faculty Manager 80% done
  - UI/UX for the role Coordinator 80% done
  - UI/UX for the role Moderator 80% done
  - UI/UX for the role Admin 70% done

**2. WORK TO BE DONE**

Continue on development work. Mainly focusing on the role Supervisor's frontend and backend.

**Tasks:** Implementation / Testing (Application based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Development work
  - Frontend for the role Supervisor
  - Backend for the role Supervisor

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**  
Continue development work as planned.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  

This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP2) Meeting Log**  
**Trimester 2, 2023 (Trimester ID:2220)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>02/06/2023  | <b>Meeting No.:</b><br>5                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

### **1. WORK DONE**

Continue developing the app. Frontend and backend development for all users module, announcement module, project module, meeting module partially done. AWS re-setup.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Frontend and Backend development
  - Users module
    - Login by role
  - Announcement module
    - View announcements (all users)
    - Create announcements (faculty manager and coordinator)
  - Project module
    - Propose projects (supervisor)
    - Show all projects (faculty manager and coordinator)
    - Show projects based on user (supervisor)
    - Project Info screen based on selected project
  - Meeting module (supervisor)
    - Create meeting
    - View meeting
- Re-setup database

## **2. WORK TO BE DONE**

Continue on development work for all modules.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Development work
  - Users module
  - Announcement module
  - Project module
  - Meeting module
  - Presentation module
  - Records module
  - Admin module

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

AWS account got suspended for violating terms of use. AWS support team managed to unsuspend me but the account was replaced with a new one.

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Continue development work as planned.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  
  
This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP2) Meeting Log**  
**Trimester 2, 2023 (Trimester ID:2220)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>16/06/2023  | <b>Meeting No.:</b><br>6                                     |
| <b>Meeting Mode:</b><br>Face to Face  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

### **1. WORK DONE**

Continue developing the app. Frontend and backend development for announcement module, project module partially done. Frontend and backend development for all users module, meeting module, presentation module, and records module done. AWS S3 bucket setup.

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Frontend and Backend development
  - Users module
    - Authentication and logout.
  - Announcement module
    - View announcements (all users)
    - Create announcements (faculty manager and coordinator)
  - Project module
    - Change status of project to accepted or rejected (coordinator)
    - Assignment of student or students (supervisor)
    - Assignment of moderator (coordinator)
    - Project shown based on moderator assignment (moderator)
  - Meeting module (supervisor)
    - List of meeting week
    - Update meeting status, work done, work to be done
  - Presentation module
    - Presentation screen
    - Create/Setup presentation parameters (faculty manager)
    - Display presentation parameters to all users.
  - Records module
    - User info screen
    - List of supervisors
    - List of moderators
    - List of students
- Setup AWS S3 for file uploads

## **2. WORK TO BE DONE**

Continue on development work for announcement module, project module, and admin module. Update documentation and start on Chapter 5 and 6.

**Tasks:** Implementation / Testing (Application based projects) or Evaluation of Findings and Research Contribution (Research based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Development work
  - Announcement module
    - Attachment upload
    - Attachment download
    - Announcement sorting
  - Project module
    - Project report upload
    - Project report download
  - Admin module
    - View all users
    - Edit users
- Documentation
  - Update Data Dictionary
  - Update ERD
  - Start on Chapter 5
  - Start on Chapter 6

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Continue development work, finalise soon.  
Draft report - update Chap 5 & 6.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  

This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.



**TPT3101 Final Year Project (FYP2) Meeting Log**  
**Trimester 2, 2023 (Trimester ID:2220)**

|   |  |
|---|--|
| <b>Meeting Date:</b><br>28/06/2023  | <b>Meeting No.:</b><br>7                                     |
| <b>Meeting Mode:</b><br>Online  |  |
| <b>Project ID:</b><br>2355  | <b>Project Type:</b><br>Application-based                    |
| <b>Project Title :</b><br>Mobile App for FYP Supervisors  |  |
| <b>Student ID :</b><br>1151102898   | <b>Student Name:</b><br>Tengku Fahad Hafiz Bin T. Abd Nawawi |
| <b>Student Programme and Specialisation:</b><br>Bachelor of Computer Science (Software Engineering) |  |
| <b>Supervisor Name:</b><br>Dr. Yeoh Eng Thiam   | <b>Co-Supervisor Name:</b><br>-                              |
| <b>Collaborating Company:</b><br>-  | <b>Company Supervisor Name:</b><br>-                         |

### **1. WORK DONE**

Finished developing the app. Frontend and backend development for announcement module, project module done. Final report draft is ready. Chapter 5 is 80% done with only 5.3: Software Modules and 5.4: Sample outputs left. Testing is done so Chapter 6 is 100% completed

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research-based projects) / ~~Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects)~~ / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Development work
  - Announcement module
    - Attachment upload
    - Attachment download
    - Announcement sorting
  - Project module
    - Project report upload
    - Project report download
- Documentation
  - Updated Data Dictionary
  - Updated ERD
  - Chapter 5
    - Deployment
    - Development Environment
  - Chapter 6
    - Test Plan
    - Test Data
    - Test Results

## **2. WORK TO BE DONE**

Finish documentation on Chapter 5.3, 5.4, and conclusion

**Tasks:** Implementation / Testing (Application-based projects) or Evaluation of Findings and Research Contribution (Research based projects) / Commercialisation Proposal (Application-based projects) or Research Paper (Research-based Projects) / Draft Final Report Completion

*(Please strike out the tasks, which are not applicable)*

**Details (in point form):**

- Documentation
  - Start on Chapter 5.3
  - Start on Chapter 5.4
  - Update Conclusion

**3. PROBLEMS ENCOUNTERED AND SOLUTIONS**

-

**4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)**

Finalise the prototype.  
Update Chap 5.

.....  
Supervisor's Signature

.....  
Student's Signature

.....  
Co-Supervisor's Signature  
(if applicable)

.....  
Company Supervisor's Signature  
(if applicable)

**IMPORTANT NOTES TO STUDENTS:**

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.
2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.  
Minimum requirement is SIX Meeting Logs (Period: Week 4 to Week 14). Students can have fortnightly meetings with the supervisor.
3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.  

This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.
4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

## APPENDIX D: PLAGIARISM CHECKING REPORT

### MOBILE APP FOR FYP SUPERVISORS - TENGKU FAHAD

#### ORIGINALITY REPORT

|                  |                  |              |                |
|------------------|------------------|--------------|----------------|
| 2%               | 1%               | 0%           | 1%             |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

#### PRIMARY SOURCES

|   |   |     |
|---|---|-----|
| 1 | Submitted to Multimedia University<br>Student Paper               | 1%  |
| 2 | fikribalerong.blogspot.com<br>Internet Source                     | <1% |
| 3 | www.citepa.org<br>Internet Source                                 | <1% |
| 4 | www.kuchingsarawak.com<br>Internet Source                         | <1% |
| 5 | Submitted to Laureate Higher Education<br>Group<br>Student Paper  | <1% |
| 6 | Submitted to University of Technology,<br>Sydney<br>Student Paper | <1% |
| 7 | Submitted to Liverpool John Moores<br>University<br>Student Paper | <1% |
| 8 | studentsrepo.um.edu.my<br>Internet Source                         | <1% |

|    |  |      |
|----|--|------|
| 9  | Submitted to American University in Cairo<br>Student Paper | <1 % |
| 10 | Submitted to University Of Tasmania<br>Student Paper       | <1 % |
| 11 | nccur.lib.nccu.edu.tw<br>Internet Source                   | <1 % |
| 12 | www.topuniversities.com<br>Internet Source                 | <1 % |
| 13 | www.coursehero.com<br>Internet Source                      | <1 % |
| 14 | www.easychair.org<br>Internet Source                       | <1 % |

---

Exclude quotes      On  
Exclude bibliography      On

Exclude matches      Off

## APPENDIX E: EXTRA CONTENTS

### ***E - 3.3.2 Faculty Manager Modules***

**Table 3.13 Faculty Manager - Login/Logout Module**

| Use Case ID    | ALUC-01   | Use Case Name | Login/Logout Module |
|----------------|---|---------------|---------------------|
| Pre-Condition  | User must be registered by the Administrator  |               |                     |
| Post-Condition | Valid login or invalid login   Logout   |               |                     |
| Scenario:      | 1) User inputs email and password.<br>2) If successful, the user is redirected to the home screen, the “Announcement” screen.<br>3) If unsuccessful, “Invalid email and/or password” alert will appear.<br>4) To logout, the user must navigate to the “Records” screen.<br>5) If the user is logged out, the user is set to null, thus ending the session. |               |                     |
| Extends List:  | None  |               |                     |

**Table 3.14 Faculty Manager - View Announcement Module**

| Use Case ID    | ALUC-02   | Use Case Name | View Announcement Module |
|----------------|---|---------------|--------------------------|
| Pre-Condition  | User must be logged in  |               |                          |
| Post-Condition | Display announcement  |               |                          |
| Scenario:      | 1) Upon successful login, the user is redirected to the “Announcement” screen, where the announcement posted is displayed.<br>2) The user can also click the “Announcement” button at the bottom navigation bar or the “MMU” logo on top of the screen to open this page. |               |                          |
| Extends List:  | None  |               |                          |

**Table 3.17 Faculty Manager - View Projects Module**

| Use Case ID    | FCUC-03  | Use Case Name | View Project Module |
|----------------|--|---------------|---------------------|
| Pre-Condition  | User must be logged in as Faculty Manager or Coordinator   |               |                     |
| Post-Condition | Able to view projects (proposed by Supervisor)   |               |                     |
| Scenario:      | 1) Upon successful login as Faculty Manager, the user must click the “Project” button on the bottom navigation bar.<br>2) This will display the Projects screen where the user can see the projects and their status:<br>a) Under Review<br>b) Accepted<br>c) Rejected<br>3) Page will be blank if there is no project proposed/created. |               |                     |

|               |   |
|---------------|---|
|               | 4) The user can click/select the project to redirect to the “Project Info” screen, where the selected project’s information will be displayed |
| Extends List: | None  |

Table 3.18 Faculty Manager - View Presentation Schedule Module

| Use Case ID    | ALUC-03  | Use Case Name | View Presentation Schedule Module |
|----------------|--|---------------|-----------------------------------|
| Pre-Condition  | User must be logged in.  |               |                                   |
| Post-Condition | Able to view presentation slots  |               |                                   |
| Scenario:      | 1) Upon successful login, the user must click the “Presentation” button on the bottom navigation bar.<br>2) This will display the “Presentation” screen where the user can see the presentations created (page will be blank if presentations are not yet set by the Faculty Manager). |               |                                   |
| Extends List:  | None   |               |                                   |

Table 3.20 Faculty Manager - View Profile Info Module

| Use Case ID    | ALUC-04  | Use Case Name | View Profile Info Module |
|----------------|--|---------------|--------------------------|
| Pre-Condition  | User must be logged in   |               |                          |
| Post-Condition | Able to view profile information   |               |                          |
| Scenario:      | 1) Upon successful login, the user must click the “Records” button on the bottom navigation bar.<br>2) This will display the Records screen where profile information is displayed:<br>a) Avatar<br>b) Name<br>c) Email<br>d) Role |               |                          |
| Extends List:  | None   |               |                          |

Table 3.21 Faculty Manager - View List of Students/Supervisors/Moderators Module

| Use Case ID    | ALUC-05  | Use Case Name | View List of Students/Supervisors/Moderators Module |
|----------------|--|---------------|---|
| Pre-Condition  | User must be logged in   |               |   |
| Post-Condition | Able to view list of supervisors, moderators, and students                                       |               |   |
| Scenario:      | 1) Upon successful login, the user must click the “Records” button on the bottom navigation bar. |               |   |

|               |  |
|---------------|--|
|               | <p>2) There will be a selectable list of buttons:</p> <ul style="list-style-type: none"> <li>a) List of Supervisors</li> <li>b) List of Moderators</li> <li>c) List of Students</li> </ul> <p>3) Depending on what is chosen, the user will be redirected to a list of Supervisors, Moderators, or Students that are registered.</p> |
| Extends List: | None   |

### **E - 3.3.3 Moderator Modules**

Table 3.22 Moderator - Login/Logout Module

| Use Case ID    | ALUC-01   | Use Case Name | Login/Logout Module |
|----------------|---|---------------|---------------------|
| Pre-Condition  | User must be registered by the Administrator  |               |                     |
| Post-Condition | Valid login or invalid login   Logout   |               |                     |
| Scenario:      | 1) User inputs email and password.<br>2) If successful, the user is redirected to the home screen, the “Announcement” screen.<br>3) If unsuccessful, “Invalid email and/or password” alert will appear.<br>4) To logout, the user must navigate to the “Records” screen.<br>5) If the user is logged out, the user is set to null, thus ending the session. |               |                     |
| Extends List:  | None  |               |                     |

Table 3.23 Moderator - View Announcement Module

| Use Case ID    | ALUC-02   | Use Case Name | View Announcement Module |
|----------------|---|---------------|--------------------------|
| Pre-Condition  | User must be logged in  |               |                          |
| Post-Condition | Display announcement  |               |                          |
| Scenario:      | 1) Upon successful login, the user is redirected to the “Announcement” screen, where the announcement posted is displayed.<br>2) The user can also click the “Announcement” button at the bottom navigation bar or the “MMU” logo on top of the screen to open this page. |               |                          |
| Extends List:  | None  |               |                          |

**Table 3.26 Moderator - View Presentation Schedule Module**

|                |  |               |                                   |
|----------------|--|---------------|-----------------------------------|
| Use Case ID    | ALUC-03  | Use Case Name | View Presentation Schedule Module |
| Pre-Condition  | User must be logged in.  |               |                                   |
| Post-Condition | Able to view presentation slots  |               |                                   |
| Scenario:      | 1) Upon successful login, the user must click the “Presentation” button on the bottom navigation bar.<br>2) This will display the “Presentation” screen where the user can see the presentations created (page will be blank if presentations are not yet set by the Faculty Manager). |               |                                   |
| Extends List:  | None   |               |                                   |

**Table 3.27 Moderator - View Profile Info Module**

|                |  |               |                          |
|----------------|--|---------------|--------------------------|
| Use Case ID    | ALUC-04  | Use Case Name | View Profile Info Module |
| Pre-Condition  | User must be logged in   |               |                          |
| Post-Condition | Able to view profile information   |               |                          |
| Scenario:      | 1) Upon successful login, the user must click the “Records” button on the bottom navigation bar.<br>2) This will display the Records screen where profile information is displayed:<br>a) Avatar<br>b) Name<br>c) Email<br>d) Role |               |                          |
| Extends List:  | None   |               |                          |

**Table 3.28 Moderator - View List of Students/Supervisors/Moderators Module**

|                |  |               |   |
|----------------|--|---------------|---|
| Use Case ID    | ALUC-05  | Use Case Name | View List of Students/Supervisors/Moderators Module |
| Pre-Condition  | User must be logged in   |               |   |
| Post-Condition | Able to view list of supervisors, moderators, and students   |               |   |
| Scenario:      | 4) Upon successful login, the user must click the “Records” button on the bottom navigation bar.<br>5) There will be a selectable list of buttons:<br>a) List of Supervisors<br>b) List of Moderators<br>c) List of Students<br>6) Depending on what is chosen, the user will be redirected to a list of Supervisors, Moderators, or Students that are registered. |               |   |
| Extends List:  | None   |               |   |

### **E - 3.3.4 Coordinator Modules**

**Table 3.29 Coordinator - Login/Logout Module**

| Use Case ID    | ALUC-01   | Use Case Name | Login/Logout Module |
|----------------|---|---------------|---------------------|
| Pre-Condition  | User must be registered by the Administrator  |               |                     |
| Post-Condition | Valid login or invalid login   Logout   |               |                     |
| Scenario:      | 1) User inputs email and password.<br>2) If successful, the user is redirected to the home screen, the “Announcement” screen.<br>3) If unsuccessful, “Invalid email and/or password” alert will appear.<br>4) To logout, the user must navigate to the “Records” screen.<br>5) If the user is logged out, the user is set to null, thus ending the session. |               |                     |
| Extends List:  | None  |               |                     |

**Table 3.30 Coordinator - View Announcement Module**

| Use Case ID    | ALUC-02   | Use Case Name | View Announcement Module |
|----------------|---|---------------|--------------------------|
| Pre-Condition  | User must be logged in  |               |                          |
| Post-Condition | Display announcement  |               |                          |
| Scenario:      | 1) Upon successful login, the user is redirected to the “Announcement” screen, where the announcement posted is displayed.<br>2) The user can also click the “Announcement” button at the bottom navigation bar or the “MMU” logo on top of the screen to open this page. |               |                          |
| Extends List:  | None  |               |                          |

**Table 3.31 Coordinator - Create Announcements Module**

| Use Case ID    | FCUC-01   | Use Case Name | Create Announcements Module |
|----------------|---|---------------|-----------------------------|
| Pre-Condition  | User must be logged in as Faculty Manager or Coordinator  |               |                             |
| Post-Condition | Announcement successfully created   |               |                             |
| Scenario:      | 1) Upon successful login as Faculty Manager, on the “Announcement” screen, the user can click the “+” button located at the bottom right of the Announcement screen.<br>2) After clicking the “+” button, this will redirect to the “Create Announcement” screen where the user must input:<br>a) Title<br>b) Description<br>c) Attachment (if any)<br>3) After inputting, the user must click submit to submit the announcement. If there’s missing input, an alert will appear.<br>4) If successful, a successful alert will appear and the announcement will be created.<br>5) Announcements created will be displayed on the “Announcement” Screen. |               |                             |

|               |                                      |
|---------------|--------------------------------------|
|               | (latest post will be on the topmost) |
| Extends List: | None                                 |

Table 3.32 Coordinator - Update Announcements Module

|                |  |               |                             |
|----------------|--|---------------|-----------------------------|
| Use Case ID    | FCUC-02  | Use Case Name | Update Announcements Module |
| Pre-Condition  | Announcements must be created by the user  |               |                             |
| Post-Condition | Announcement successfully updated  |               |                             |
| Scenario:      | 1) Upon successful login as Faculty Manager, on the “Announcement” screen, the user can click the “...” button located at the top right of any of the Announcements created.<br>2) The User can then pin the announcement, or delete it. |               |                             |
| Extends List:  | None   |               |                             |

Table 3.33 Coordinator - View Projects Module

|                |   |               |                     |
|----------------|---|---------------|---------------------|
| Use Case ID    | FCUC-03   | Use Case Name | View Project Module |
| Pre-Condition  | User must be logged in as Faculty Manager or Coordinator  |               |                     |
| Post-Condition | Able to view projects (proposed by Supervisor)  |               |                     |
| Scenario:      | 1) Upon successful login as Faculty Manager, the user must click the “Project” button on the bottom navigation bar.<br>2) This will display the Projects screen where the user can see the projects and their status:<br>a) Under Review<br>b) Accepted<br>c) Rejected<br>3) Page will be blank if there is no project proposed/created.<br>4) The user can click/select the project to redirect to the “Project Info” screen, where the selected project’s information will be displayed |               |                     |
| Extends List:  | None  |               |                     |

Table 3.36 Coordinator - View Presentation Schedule Module

|                |  |               |                                   |
|----------------|--|---------------|-----------------------------------|
| Use Case ID    | ALUC-03  | Use Case Name | View Presentation Schedule Module |
| Pre-Condition  | User must be logged in.  |               |                                   |
| Post-Condition | Able to view presentation slots  |               |                                   |
| Scenario:      | 1) Upon successful login, the user must click the “Presentation” button on the bottom navigation bar.<br>2) This will display the “Presentation” screen where the user can see the presentations created (page will be blank if presentations are not yet set by the Faculty Manager). |               |                                   |

|               |      |
|---------------|------|
| Extends List: | None |
|---------------|------|

Table 3.37 Coordinator - View Profile Info Module

| Use Case ID    | ALUC-04  | Use Case Name | View Profile Info Module |
|----------------|--|---------------|--------------------------|
| Pre-Condition  | User must be logged in   |               |                          |
| Post-Condition | Able to view profile information   |               |                          |
| Scenario:      | 1) Upon successful login, the user must click the “Records” button on the bottom navigation bar.<br>2) This will display the Records screen where profile information is displayed:<br>a) Avatar<br>b) Name<br>c) Email<br>d) Role |               |                          |
| Extends List:  | None   |               |                          |

Table 3.38 Coordinator - View List of Students/Supervisors/Moderators Module

| Use Case ID    | ALUC-05  | Use Case Name | View List of Students/Supervisors/Moderators Module |
|----------------|--|---------------|---|
| Pre-Condition  | User must be logged in   |               |   |
| Post-Condition | Able to view list of supervisors, moderators, and students   |               |   |
| Scenario:      | 1) Upon successful login, the user must click the “Records” button on the bottom navigation bar.<br>2) There will be a selectable list of buttons:<br>a) List of Supervisors<br>b) List of Moderators<br>c) List of Students<br>3) Depending on what is chosen, the user will be redirected to a list of Supervisors, Moderators, or Students that are registered. |               |   |
| Extends List:  | None   |               |   |

### **E - 5.3.1 Authentication Algorithm**

Algorithm Description:

1. Start: The user launches the app and is presented with the login screen.

2. User Input: The user enters their email and password in the respective input fields.
3. Login Request: When the user taps the "Login" button, a request is sent to the server to authenticate the user's credentials.
4. Server Verification: The server receives the login request and verifies the provided email and password against the stored user data.
5. Authentication Success: If the server verifies the credentials, it returns a successful authentication response along with the user data.
6. Update Authentication State: The app updates the authentication state by storing the user data in the authentication context.
7. Account Type Check: The app checks the user's account type to determine the appropriate screen to navigate to based on their role (e.g., supervisor, faculty manager, coordinator, moderator).
8. Navigate to Relevant Screen: The app navigates the user to the corresponding screen based on their account type.
9. Authentication Failure: If the server rejects the login request or the authentication process fails for any reason, an error message is displayed to the user.
10. Display Error Modal: The app shows a modal with an error title and message, allowing the user to try logging in again or take appropriate action based on the error.
11. End: The authentication process concludes, and the user is either successfully authenticated and redirected to the appropriate screen or notified of the authentication failure.

### Sample Code:

The following code snippet illustrates the implementation of the authentication algorithm in the Login.jsx component:

```
JavaScript
// ... Code before authentication process ...

const handleLogin = () => {
  const requestBody = {
    email: staffID,
    password: password
  };

  fetch('http://3.26.23.172/signin', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(requestBody)
  })
  .then(response => response.json())
  .then(data => {
    const { status, data: userData } = data;
    if (status) {
      login(userData); // Store user data in the context
      const account_type = userData.account_type;
      // Navigate based on the account type
      if (account_type === 'supervisor') {
        navigation.navigate('SUAnnScreen');
      } else if (account_type === 'faculty_manager') {
        navigation.navigate('FMAAnnScreen');
      } else if (account_type === 'coordinator') {
        navigation.navigate('COAnnScreen');
      } else if (account_type === 'moderator') {
        navigation.navigate('MOAnnScreen');
      } else {
        Alert.alert('Invalid account type');
      }
    } else {
      setModalTitle('Invalid Login!');
      setModalMessage('The email and password you entered did not
match our records. Please double-check and try again.');
      setModalButtonText('Try again');
      setModalVisible(true);
    }
  })
}

export default handleLogin;
```

```

})
.catch(error => {
  setModalTitle('Error');
  setModalMessage('Login failed. Something went wrong on our
end.');
  setModalButtonText('Try again');
  setModalVisible(true);
  console.error(error);
});
;

// ... Code after authentication process ...

```

#### **E - 5.3.4 Project Approval and Rejection Algorithm**

Algorithm Description:

1. When the component is rendered, the project information is displayed, including the project ID, title, proposed by, type, specialization, focus, description, objectives, scope, number of students, and industry collaboration details.
2. If the project's status is "Pending," the "Approve" and "Decline" buttons are displayed. Otherwise, if the status is "Accepted" and a student is assigned, the "Assign Moderator" or "Reassign Moderator" button is displayed.
3. When the "Approve" button is pressed, an HTTP POST request is sent to the specified URL ([http://3.26.23.172/update\\_project\\_status](http://3.26.23.172/update_project_status)) to update the project status to "Accepted."
4. If the response to the request is successful (status code 200), a success modal is displayed with the title "Success!" and the message "Project accepted."

5. If the response is not successful, an error message is logged to the console, and an alert is shown to the user with the message "Failed to accept project."
6. When the "Decline" button is pressed, a similar HTTP POST request is sent to update the project status to "Declined."
7. If the response is successful, a success modal is displayed with the title "Success!" and the message "Project rejected."
8. If the response is not successful, an error message is logged to the console, and an alert is shown to the user with the message "Failed to reject project."
9. The user can close the success modal by clicking the "OK" button.
10. If an error occurs during the API request, an error message is logged to the console, and an alert is shown to the user with the message "An error occurred while accepting/rejecting the project."

Sample Code:

The following code snippet demonstrates the implementation of the project approval and rejection algorithm:

```
JavaScript
import React, { useState } from 'react';
import { View, Text, TouchableOpacity, Modal } from 'react-native';

const ProjectApprovalScreen = ({ project }) => {
  const [isModalVisible, setModalVisible] = useState(false);
  const [modalTitle, setModalTitle] = useState('');
  const [modalMessage, setModalMessage] = useState('');
  const [modalButtonText, setModalButtonText] = useState('');

  const handleAccept = async () => {
    try {

```

```

        const response = await
fetch('http://3.26.23.172/update_project_status', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify({
        projectId: project.projid,
        status: 'Accepted',
    }),
});

if (response.ok) {
    setModalTitle('Success!');
    setModalMessage('Project accepted');
    setModalButtonText('OK');
    setModalVisible(true);
} else {
    console.error('Failed to accept project');
    alert('Failed to accept project');
}
} catch (error) {
    console.error('Error accepting project:', error);
    alert('An error occurred while accepting project');
}
};

const handleReject = async () => {
    try {
        const response = await
fetch('http://3.26.23.172/update_project_status', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
    },
    body: JSON.stringify({
        projectId: project.projid,
        status: 'Declined',
    }),
});

if (response.ok) {
    setModalTitle('Success!');
    setModalMessage('Project rejected');
    setModalButtonText('OK');
    setModalVisible(true);
} else {
    console.error('Failed to reject project');
}
}

```

```

        alert('Failed to reject project');
    }
} catch (error) {
    console.error('Error rejecting project:', error);
    alert('An error occurred while rejecting project');
}
};

const handleModalButtonPress = () => {
    if (modalButtonText === 'Try again') {
        setModalVisible(false);
    } else if (modalButtonText === 'OK') {
        setModalVisible(false);
        navigation.goBack();
    }
    setModalVisible(false);
};

return (
    <View>
        {/* Display project information */}
        <Text>Project Title: {project.projecttitle}</Text>
        {/* Add other project details here */}

        {/* Accept and Reject Buttons */}
        {project.projectstatus === 'Pending' && (
            <View>
                <TouchableOpacity onPress={handleAccept}>
                    <Text>Approve</Text>
                </TouchableOpacity>
                <TouchableOpacity onPress={handleReject}>
                    <Text>Decline</Text>
                </TouchableOpacity>
            </View>
        )}
    <Modal visible={isModalVisible} transparent>
        <View>
            <Text>{modalTitle}</Text>
            <Text>{modalMessage}</Text>
            <TouchableOpacity onPress={handleModalButtonPress}>
                <Text>{modalButtonText}</Text>
            </TouchableOpacity>
        </View>
    </Modal>
</View>
);
};

```

## E - 6.1.1 Test Data

Table 6.2 Test Data

| Announcements (1 - 3) |   |  |  |
|-----------------------|---|--|--|
| annId                 | 1   | 2  | 3  |
| annTitle              | Welcome to FYP Mobile App!  | Project Proposals  | FYP Report Template  |
| annDescription        | To all personnel of the FCI committee of MMU, welcome to the App! Depending on your role, please go through the app and discover the functionalities of the app on your own. If you have any questions, you can email me at ahmad@example.com. Thank you! | To all supervisors, you may start proposing projects. All coordinators are encouraged to go through the proposals thoroughly. Happy proposing! | Dear supervisors, below you can find an attachment of the FYP Report template. You may add some things to it. Don't forget to share it with your students. |
| upload                | NULL  | NULL   | dummy.pdf  |
| annDatePosted         | 2023-06-27T03:11:08.016Z  | 2023-06-27T03:21:54.556Z   |  |
| annPostedBy           | John Doe  | Tan Wei Jie  | Jane Smith   |
| isPinned              | No  | Yes  | NULL   |
| Projects (1 - 3)      |   |  |  |
| projId                | 1   | 2  | 3  |
| supervisor_id         | 3   | 7  | 3  |
| projectPostedBy       | Ahmad Abdullah  | Mohd Ali   | Ahmad Abdullah   |
| projectTitle          | Smart Home Automation System  | Data Analysis and Visualization Platform   | Multiplayer Game Development for Mobile Platforms  |
| projectType           | Application Based   | Application Based  | Application Based  |
| projectSpecialization | Software Engineering  | Data Science   | Game Development   |
| projectFocus          | Home Automation   | Data Visualization   | Multiplayer Gaming   |
| projectDescription    | The project aims to develop a smart home automation system using IoT technologies. It will  | This project aims to develop a data analysis and visualization platform that enables users to  | The project aims to develop a multiplayer game for mobile platforms, providing an immersive  |

|                   |  |   |   |
|-------------------|--|---|---|
|                   | enable users to control various home devices remotely and automate tasks for enhanced convenience and energy efficiency.   | explore and gain insights from large datasets. It will provide intuitive visualizations and interactive tools for data exploration.   | and engaging gaming experience for users. The game will incorporate real-time multiplayer functionality, interactive gameplay elements, and captivating visual effects.   |
| projectObjectives | <ul style="list-style-type: none"> <li>1. Design and implement a centralized control system,</li> <li>2. Develop IoT device integration and communication protocols,</li> <li>3. Create a user-friendly mobile application for remote control</li> </ul> | <ul style="list-style-type: none"> <li>1. Design and implement data ingestion and preprocessing pipeline,</li> <li>2. Develop visual analytics components and interactive dashboards,</li> <li>3. Incorporate advanced analytics algorithms for data exploration</li> </ul> | <ul style="list-style-type: none"> <li>1. Design and implement game mechanics, controls, and user interface,</li> <li>2. Develop multiplayer networking features and matchmaking system,</li> <li>3. Create captivating game levels and challenging gameplay scenarios</li> </ul> |
| projectScope      | The project will focus on controlling lights, temperature, and security systems within a residential setting. Integration with voice assistants and energy monitoring features will also be explored.  | The project will focus on handling structured and semi-structured data and provide features such as filtering, aggregation, and drill-down capabilities. Integration with popular data analysis libraries will be explored.   | The project will focus on developing a cross-platform game using popular game development frameworks and libraries. It will include features such as real-time multiplayer, in-app purchases, and leaderboard integration.  |
| numOfStudents     | 1  | 1   | 2   |
| studentOneSub     | NULL   | NULL  | Game Design and Level Development   |
| studentOneWork    | NULL   | NULL  | Responsible for designing game mechanics, creating game levels, and implementing gameplay elements. Collaborate with the team to ensure seamless integration of design and functionality.   |
| studentTwoSub     | NULL   | NULL  | Networking and Multiplayer Integration  |
| studentTwoWork    | NULL   | NULL  | Responsible for implementing multiplayer networking functionality, developing matchmaking systems, and ensuring smooth and stable multiplayer experience. Collaborate with the team to integrate gameplay elements with networking components.                                    |
| industryCollab    | Yes  | No  | No  |

|                         |  |   |  |
|-------------------------|--|---|--|
| industryCompany         | ABC Technologies   | NULL  | NULL   |
| industryContact         | Michael Tan<br>(michael@example.com)   | NULL  | NULL   |
| projectUpload           | NULL   | NULL  | NULL   |
| projectUpload1          | NULL   | [1_dummy.pdf]   | NULL   |
| projectUpload2          | NULL   | NULL  | NULL   |
| projectStatus           | Accepted   | Declined  | Accepted   |
| studentAssigned         | Assigned   | Assigned  | Not Assigned   |
| student                 | [Bruce, Lee]   | Syed Naufal, Ahmad Nazmi  | NULL   |
| moderatorAssigned       | Not Assigned   | Assigned  | Not Assigned   |
| moderator_id            | NULL   | 4   | NULL   |
| moderator_name          | NULL   | Lim Wei Ling  | NULL   |
| <b>Projects (4 - 6)</b> |  |   |  |
| projId                  | 4  | 5   | 6  |
| supervisor_id           | 7  | 3   | 7  |
| projectPostedBy         | Mohd Ali   | Ahmad Abdullah  | Mohd Ali   |
| projectTitle            | Mobile App for Health and Fitness Tracking   | E-Commerce Website Development  | Exploring User Behavior in E-Commerce Systems  |
| projectType             | Application Based  | Application Based   | Research Based   |
| projectSpecialization   | Software Engineering   | Information System  | Information System   |
| projectFocus            | Fitness Tracking   | Online Retail   | User Behavior Analysis   |
| projectDescription      | The project involves developing a mobile application for health and fitness tracking. The app will allow users to monitor their exercise routines, set goals, track nutrition, and receive personalized recommendations for a healthy lifestyle. | This project involves developing an e-commerce website for an online retail store. The website will provide a user-friendly interface for customers to browse products, add items to cart, and make secure online payments. | The project aims to conduct a comprehensive study on user behavior in e-commerce systems. It will involve collecting and analyzing user data to gain insights into user preferences, purchasing patterns, and factors influencing user decisions in online shopping. |

|                   |   |  |   |
|-------------------|---|--|---|
| projectObjectives | <ol style="list-style-type: none"> <li>1. Design and develop a user-friendly mobile app interface,</li> <li>2. Implement fitness tracking features and integrate with wearable devices,</li> <li>3. Provide personalized insights and recommendations based on user data</li> </ol> | <ol style="list-style-type: none"> <li>1. Design and develop a responsive website layout,</li> <li>2. Implement product catalog and shopping cart functionality,</li> <li>3. Integrate secure payment gateways"</li> </ol> | <ol style="list-style-type: none"> <li>1. Design and conduct surveys and interviews to collect user data,</li> <li>2. Analyze and interpret collected data using statistical analysis techniques,</li> <li>3. Identify and document key findings and implications for e-commerce system design and optimization.</li> </ol> |
| projectScope      | <p>The project will cover features such as step counting, calorie tracking, workout logging, and integration with popular fitness APIs. Data privacy and security measures will be implemented.</p>   | <p>The project will cover essential e-commerce features such as user registration, product search, order tracking, and inventory management. Scalability and performance optimization will be considered.</p>              | <p>The project will focus on studying user behavior across different e-commerce platforms and product categories. It will explore factors such as user demographics, product recommendations, user reviews, and website usability.</p>  |
| numOfStudents     | 1   | 1  | 2   |
| studentOneSub     | NULL  | NULL   | Data Collection and Analysis  |
| studentOneWork    | NULL  | NULL   | <p>Responsible for designing and administering surveys, conducting interviews, and collecting user data. Analyze collected data using statistical software and assist in interpreting the findings.</p>   |
| studentTwoSub     | NULL  | NULL   | User Behavior Insights and Documentation  |
| studentTwoWork    | NULL  | NULL   | <p>Responsible for analyzing and interpreting collected data, identifying user behavior patterns, and documenting key findings. Contribute to the preparation of research reports and presentations.</p>  |
| industryCollab    | Yes   | No   | No  |
| industryCompany   | FitTech Solutions   | NULL   | NULL  |
| industryContact   | Vincent Gogh<br>(vgogh@example.com)   | NULL   | NULL  |
| projectUpload     | NULL  | NULL   | NULL  |

|                                |  |   |  |
|--------------------------------|--|---|--|
| projectUpload1                 | NULL   | NULL  | NULL   |
| projectUpload2                 | NULL   | NULL  | NULL   |
| projectStatus                  | Pending  | Pending   | Accepted   |
| studentAssigned                | Not assigned   | Not assigned  | Not assigned   |
| student                        | NULL   | NULL  | NULL   |
| moderatorAssigned              | Not assigned   | Not assigned  | Not assigned   |
| moderator_id                   | NULL   | NULL  | NULL   |
| moderator_name                 | NULL   | NULL  | NULL   |
| <b>Meeting Details (1 - 3)</b> |  |   |  |
| meetDetId                      | 1  | 2   | 3  |
| meetId                         | 1  | 1   | 2  |
| week                           | 1  | 2   | 1  |
| meetingMode                    | Face To Face   | Online  | Face to Face   |
| workDone                       | Discussed project requirements and goals. Assigned tasks. Conducted initial research on available automation technologies.               | Reviewed progress on prototype development. Discussed challenges and possible solutions. Shared research findings on sensor integration and UI design.      | Discussed project goals, scope, and technical requirements. Established team roles and responsibilities. Set up project development environment.                                     |
| workToBeDone                   | Develop prototype for system architecture. Research potential sensors and actuators for home automation. Start designing user interface. | Integrate sensors with the central control system. Develop user interface screens for controlling different devices. Conduct initial testing and debugging. | Design game mechanics and user interface. Develop game levels and implement basic gameplay features. Research multiplayer networking solutions.                                      |
| comments                       | Productive discussion. Need to schedule a follow-up meeting to finalize project timeline and milestones                                  | Good progress made on the prototype. Need to allocate additional resources for sensor integration. Consideration of user feedback for UI enhancements.      | Positive start to the project. Team members are enthusiastic and motivated. Need to conduct further research on multiplayer networking options and finalize the game design concept. |
| meetStatus                     | Done   | Done  | Done   |

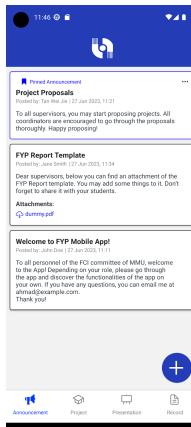
### **E - 6.2.1 Test Case - ALUC**

Table 6.4 Test Case - ALUC-01-2

| Use Case ID           | ALUC-01  | Use Case Name | Login/Logout Module |
|-----------------------|--|---------------|---------------------|
| Test Case Description | Valid logout and session will end  |               |                     |
| Test Type             | Acceptance Test Case   |               |                     |
| Test Condition        | Click on logout button   |               |                     |
| Result                |  |               |                     |
| Expected Test Result  | Logout successful. Navigate to login screen  |               |                     |
| Actual Result         | Logout successful. Navigate to login screen  |               |                     |
| Status                | Pass   |               |                     |
| Sample Output         |  |               |                     |
| Remarks               |  |               |                     |

Table 6.6 Test Case - ALUC-02

| Use Case ID           | ALUC-02  | Use Case Name | View Announcement Module |
|-----------------------|--|---------------|--------------------------|
| Test Case Description | Announcements are displayed correctly for every user and role. Sorting should be from newest to oldest. Pinned announcements should be at the topmost, ignoring the sorting order. |               |                          |
| Test Type             | Acceptance and Database Test Case  |               |                          |
| Test Condition        | Navigate to “Announcement screen” for every users  |               |                          |
| Result                |  |               |                          |

|                      |   |
|----------------------|---|
| Expected Test Result | Announcement cards are displayed with the correct information stored in the database. Announcement cards are displayed to every user. Sorting of announcements is from newest to oldest. Pinned announcement is at the top, ignoring the sorting order. |
| Actual Result        | Announcement cards are displayed with the correct information stored in the database. Announcement cards are displayed to every user. Sorting of announcements is from newest to oldest. Pinned announcement is at the top, ignoring the sorting order. |
| Status               | Pass  |
| Sample Output        |    |
| Remarks              | Test is done to every user with different roles   |

### E - 6.2.2 Test Case - SUUC

Table 6.7 Test Case - SUUC-01

| Use Case ID           | SUUC-01  | Use Case Name                              | Add New Project Module                     |  |  |  |  |  |
|-----------------------|--|--|--|--|--|--|--|--|
| Test Case Description | Only accessible by role “Supervisor”, thus only supervisor are allowed to add projects |  |  |  |  |  |  |  |
| Test Type             | Acceptance Test Case   |  |  |  |  |  |  |  |
| Test Condition        | Logged in as the role “Supervisor”   | Logged in as the role “Faculty Manager”    | Logged in as the role “Coordinator”        | Logged in as the role “Moderator”          |  |  |  |  |
| Result                |  |  |  |  |  |  |  |  |
| Expected Test Result  | Navigate to “Project proposal screen”  | No navigation to “Project proposal screen” | No navigation to “Project proposal screen” | No navigation to “Project proposal screen” |  |  |  |  |
| Actual Result         | Navigate to “Project proposal screen”  | No navigation to “Project proposal screen” | No navigation to “Project proposal screen” | No navigation to “Project proposal screen” |  |  |  |  |
| Status                | Pass   | Pass                                       | Pass                                       | Pass                                       |  |  |  |  |

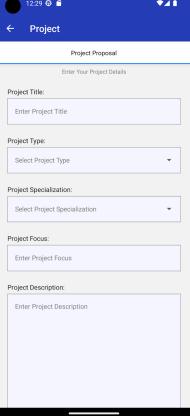
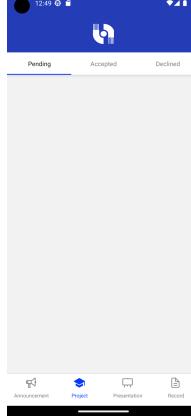
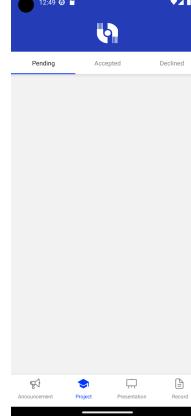
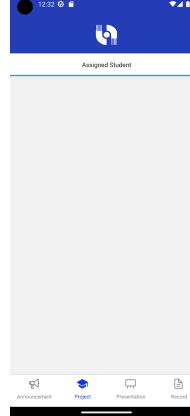
|               |   |   |  |   |
|---------------|---|---|--|---|
| Sample Output |  |  |  |  |
| Remarks       | “+” button appears in the “Project screen” to allow access.                       | No “+” button found in the “Project screen”. Access denied                        | No “+” button found in the “Project screen”. Access denied                         | No “+” button found in the “Project screen”. Access denied                          |

Table 6.10 Test Case - SUUC-03-2

| Use Case ID           | SUUC-03  | Use Case Name | Assign Student to Project Module   |
|-----------------------|--|---------------|--|
| Test Case Description | Successful submission only if the required inputs are met. Project and Student are updated in the database |               |  |
| Test Type             | Functionality and Database Test Case   |               |  |
| Test Input            | Valid Input  |               | Invalid Input  |
| Student Name          | Bruce Lee  |               | No input (NULL)  |
| Student Email         | blee101@student.com  |               | No input (NULL)  |
| Student ID            | 101  |               | No input (NULL)  |
| if numOfStudents = 2  |  |               |  |
| Student 1 Name        | Syed Naufal  |               | No input (NULL)  |
| Student 1 Email       | naufal110@student.com  |               | No input (NULL)  |
| Student 1 ID          | 110  |               | No input (NULL)  |
| Student 2 Name        | Ahmad Nazmi  |               | No input (NULL)  |
| Student 2 Email       | nazmi115@student.com   |               | No input (NULL)  |
| Student 2 ID          | 115  |               | No input (NULL)  |
| Result                |  |               |  |
| Expected Test Result  | Student/s assignment submission successful. Send data to the server.                                       |               | Student/s assignment submission unsuccessful. Data not sent to the server. |

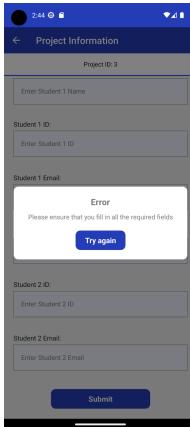
|               |  |   |
|---------------|--|---|
|               | Navigate to “Project Info screen”  |   |
| Actual Result | Student assignment submission successful. Send data to the server. Navigate to “Project Info screen” | Student/s assignment submission unsuccessful. Data not sent to the server.          |
| Status        | Partially Pass   | Partially Pass  |
| Sample Output |                     |  |
| Remarks       | The student id is submitted as null in the database  | The student id is submitted as null in the database                                 |

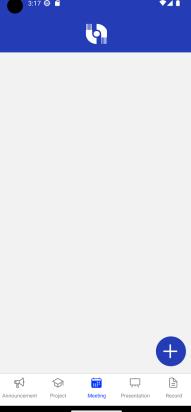
Table 6.12 Test Case - SUUC-05-06-07

|                       |  |  |  |  |
|-----------------------|--|--|--|--|
| Use Case ID           | SUUC-05<br>SUUC-06<br>SUUC-07  | Use Case Name  | Create Meeting Module<br>View Meeting Module<br>Update Meeting Module                    |  |
| Test Case Description | Only accessible by role “Supervisor”, thus only supervisor are allowed to view meeting, create a meeting, and update meeting |  |  |  |
| Test Type             | Acceptance Test Case   |  |  |  |
| Test Condition        | Logged in as the role “Supervisor”   | Logged in as the role “Faculty Manager”  | Logged in as the role “Coordinator”  | Logged in as the role “Moderator”  |
| Result                |  |  |  |  |
| Expected Test Result  | Navigate to “Meeting screen”, or “Create Meeting screen” or “Meeting Details screen”   | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” |
| Actual Result         | Navigate to “Meeting screen”, or “Create Meeting screen” or “Meeting Details screen”   | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” | No navigation to “Meeting screen” or “Create Meeting screen” or “Meeting Details screen” |

| Status        | Pass  | Pass  | Pass  | Pass  |
|---------------|---|---|---|---|
| Sample Output |   |   |   |   |
| Remarks       | “Meeting” button found in the bottom navigation bar. “+” button found in the “Meeting screen” | No “Meeting” button found in the bottom navigation bar. Access denied | No “Meeting” button found in the bottom navigation bar. Access denied | No “Meeting” button found in the bottom navigation bar. Access denied |

Table 6.14 Test Case - SUUC-06-07

|                       |   |               |  |
|-----------------------|---|---------------|--|
| Use Case ID           | SUUC-06<br>SUUC-07  | Use Case Name | View Meeting Module<br>Update Meeting Module |
| Test Case Description | Meeting display based on the creator. Meeting details is displayed correctly for the selected meeting |               |  |
| Test Type             | Functionality and Database Test Case  |               |  |
| Test User             | Valid User  |               | Invalid User                                 |
| User Acc ID           | 1   |               |  |
| Result                |   |               |  |
| Expected Test Result  | Meeting displayed   |               | No meeting displayed                         |
| Actual Result         | Meeting displayed   |               | No meeting displayed                         |
| Status                | Pass  |               |  |

|               |  |   |
|---------------|--|---|
| Sample Output |   |  |
| Remarks       | <p>Clicking on the project card navigates to the “Meeting Details screen” and displays the correct meeting details information based on selected meeting and week.</p> | <p>No meeting found as the user never created a meeting.</p>                        |

### E - 6.2.3 Test Case - FCUC

Table 6.17 Test Case - FCUC-01-2

| Use Case ID           | FCUC-01   | Use Case Name | Create Announcement Module   |
|-----------------------|---|---------------|--|
| Test Case Description | Successful submission only if all required inputs are met. Attachment is uploaded to the bucket if there is any. Announcement information is sent to the database if successful submission.   |               |  |
| Test Type             | Functionality and Database Test Case  |               |  |
| Test Input            | Valid Input   |               | Invalid  |
| Title                 | Welcome to FYP Mobile App!  |               | No input (NULL)  |
| Description           | To all personnel of the FCI committee of MMU, welcome to the App! Depending on your role, please go through the app and discover the functionalities of the app on your own. If you have any questions, you can email me at ahmad@example.com. Thank you! |               | No input (NULL)  |
| Attachment (if any)   | NULL    any uploads   |               | NULL    any uploads  |
| Result                |   |               |  |
| Expected Test Result  | Announcement submission successful. Send data to the server. Navigate to “Announcement screen”  |               | Announcement submission unsuccessful. Data not sent to the server. |
| Actual Result         | Announcement submission successful.   |               | Announcement submission unsuccessful.                              |

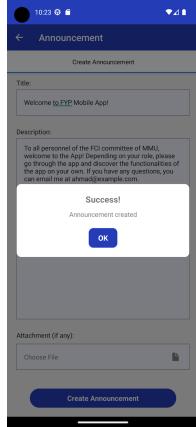
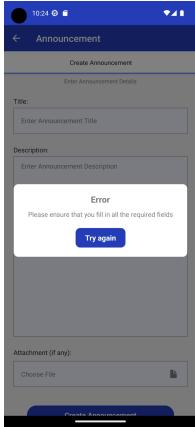
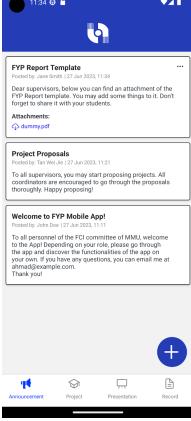
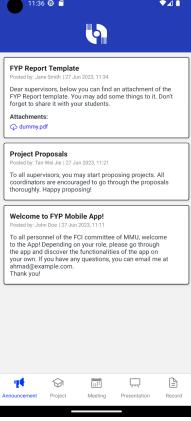
|               |   |   |
|---------------|---|---|
|               | Send data to the server. Navigate to “Announcement screen”                        | Data not sent to the server.  |
| Status        | Pass  | Pass  |
| Sample Output |  |  |
| Remarks       | Data is sent to the database server.<br>Attachment uploaded to the bucket         | Data not sent to the database server  |

Table 6.18 Test Case - FCUC-02

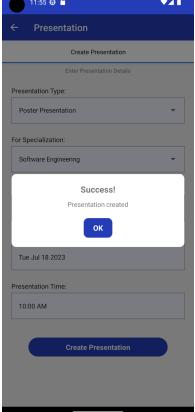
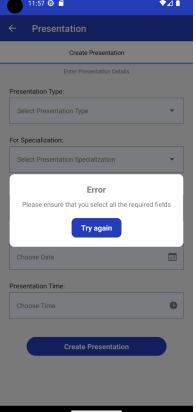
| Use Case ID           | FCUC-02   | Use Case Name                             | Update Announcement Module |
|-----------------------|---|---|----------------------------|
| Test Case Description | “Update Announcement” button displaying on the top right of an announcement card if user is the creator of the announcement |   |                            |
| Test Type             | Functionality and Database Test Case  |   |                            |
| Test User             | Valid User  | Invalid User                              |                            |
| User Acc ID           | 1   | 2   |                            |
| Result                |   |   |                            |
| Expected Test Result  | “Update Announcement” button appears on the top right of an announcement container if the user is the announcement creator. | No “Update Announcement” button displayed |                            |
| Actual Result         | “Update Announcement” button appears on the top right of an announcement container if the user is the announcement creator. | No “Update Announcement” button displayed |                            |
| Status                | Pass  | Pass                                      |                            |

|               |  |   |
|---------------|--|---|
| Sample Output |  <p>The screenshot shows the FYP Report Template mobile application interface. At the top, there is a header with the title 'FYP Report Template' and a timestamp '11:34'. Below the header, there is a pinned announcement from 'Dr. Suresh' dated '27 Jun 2023, 11:08'. The announcement text reads: 'Dear supervisors, below you can find an attachment of the FYP Report template. You may add some things to it. Don't forget to share it with your students.' An attachment named 'Summary.pdf' is listed. Below the announcement, there is a section titled 'Project Proposals' with a timestamp 'Posted by: Dr. Suresh on 13 Jul 2023, 11:11'. The text in this section encourages users to start proposing projects. At the bottom of the screen, there is a navigation bar with icons for 'Announcement', 'Project', 'Meeting', 'Presentation', and 'Record'.</p> |  <p>The screenshot shows the FYP Report Template mobile application interface. It displays a confirmation dialog box asking if the user wants to unpin the announcement. The dialog box contains the text: 'Are you sure you want to unpin this?'. The background of the app shows the same pinned announcement and project proposals sections as the previous screenshot.</p> |
| Remarks       | Clicking on the “Update Announcement” pops up a selection to pin, unpin, or delete announcement  |   |

#### E - 6.2.4 Test Case - FMUC

Table 6.22 Test Case - FMUC-01-2

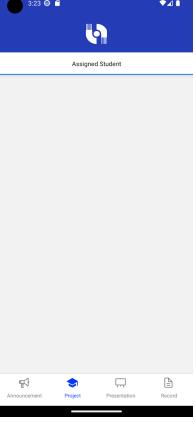
| Use Case ID           | FMUC-01   | Use Case Name | Setup Presentation Parameters Module  |
|-----------------------|---|---------------|---|
| Test Case Description | Successful submission only if all required inputs are met. Presentation information is sent to the database if successful submission. |               |   |
| Test Type             | Functionality and Database Test Case  |               |   |
| Test Input            | Valid Input   |               | Invalid Input   |
| Presentation Type     | Poster Presentation   |               | No input (NULL)   |
| For Specialization    | Software Engineering  |               | No input (NULL)   |
| Presentation Venue    | MMU Multipurpose Hall (MPH)   |               | No input (NULL)   |
| Presentation Date     | Tue Jul 18 2023   |               | No input (NULL)   |
| Presentation Time     | 10:00AM   |               | No input (NULL)   |
| Result                |   |               |   |
| Expected Test Result  | Presentation parameters submission successful. Send data to the server. Navigate to “Presentation screen”                             |               | Presentation parameters submission unsuccessful. Data not sent to the server. |
| Actual Result         | Presentation parameters submission successful. Send data to the server. Navigate to “Presentation screen”                             |               | Presentation parameters submission unsuccessful. Data not sent to the server. |
| Status                | Pass  |               | Pass  |

|               |   |   |
|---------------|---|---|
| Sample Output |  |  |
| Remarks       | Data is sent to the database server   | Data not sent to the database server  |

### E - 6.2.6 Test Case - MOUC

Table 6.25 Test Case - MOUC-01

| Use Case ID           | MOUC-01  | Use Case Name | View Assigned Students' Project Module |
|-----------------------|--|---------------|--|
| Test Case Description | Projects are displayed correctly based on their moderator/user assigned. Project information is displayed correctly for the selected project |               |  |
| Test Type             | Functionality and Database Test Case   |               |  |
| Test User             | Valid User   |               | Invalid User                           |
| User Acc ID           | 4  |               | 8                                      |
| <b>Result</b>         |  |               |  |
| Expected Test Result  | Project displayed  |               | No project displayed                   |
| Actual Result         | Project displayed  |               | No project displayed                   |
| Status                | Pass   |               | Pass                                   |

|               |  |   |
|---------------|--|---|
| Sample Output |   |  |
| Remarks       | Clicking on the project card navigates to the “Project Information screen” and displays the correct project information. | No project found as the user is not assigned to project                             |