

**UMM AL-QURA UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SYSTEMS
INFORMATION SYSTEMS DEPARTMENT**



مُعَد

Muad

Reem Saeed Al-Asmari

443008338

Lujain Fahd Al-Harbi

443000385

Raghad Saeed Al Zahrani

443008847

Asmaa Mohammed AL-Azman

443000658

Reham Abdulkarim Abuhusayn

442014839

Project Instructor:

Hanan M. Hayat

Abstract

The "Muad" app is the perfect companion for students to manage their academic life with intelligence and efficiency. Using artificial intelligence technologies, the app helps organize study schedules, track academic progress, and improve time management. It also offers essential features such as attendance tracking and GPA calculation. Additionally, students can receive support and guidance from academic ambassadors outstanding students in their fields who offer advice and assistance to their peers.

Table of contents

Table of Contents

Abstract.....	2
1 CHAPTER 1 INTRODUCTION	7
1.0 Introduction.....	7
1.1 Problem domain.....	7
1.2 Problem statement	8
1.3 Proposed system.....	8
1.4 Aims and Objectives	9
1.5 Proposed system features	9
1.5.1 Planning and Requirements Gathering:	9
1.5.2 System Design:	9
1.5.3 Development:.....	10
1.5.4 Testing:.....	10
1.5.5 Deployment:.....	10
1.5.6 Maintenance and Updates:.....	10
1.6 Project Methodology.....	11
1.7 Report Layout	13
1.8 Conclusion	14
2 CHAPTER 2 BACKGROUND/EXISTING WORK.....	15
2.1 Introduction.....	15
2.2 Overview of existing projects	16
2.3 Limitations of existing projects.....	17
2.4 Innovations of our project.....	18
2.5 Conclusion	19
3 CHAPTER 3 SOFTWARE REQUIREMENTS SPECIFICATION	20
3.1 Introduction.....	20
3.1.1 Purpose.....	20
3.2 Document Conventions.....	21
3.2.1 Document Format	21
3.2.2 Language.....	21
3.2.3 Section Headings	21
3.2.4 Numbering System.....	21
3.2.5 Version Control.....	21
3.2.6 Stakeholder Signatures.....	22

3.3	Intended Audience and Reading Suggestions	22
3.3.1	Project Stakeholders	22
3.3.2	Development Team	22
3.3.3	End Users	22
3.3.4	Academic Advisors and Educational Experts	22
3.4	Suggestions for the software requirements specification document include....	23
3.4.1	Provide an Overview.....	23
3.4.2	Use Clear Language.....	23
3.4.3	Organize the Document Effectively.....	23
3.4.4	Include Functional Requirements	23
3.4.5	Address Non-Functional Requirements.....	23
3.4.6	Provide a Glossary of Terms.....	24
3.4.7	Include Appendices.....	24
3.4.8	Seek Feedback and Clarification	24
3.5	Product Scope.....	25
3.6	Product Perspective	25
3.7	Project requirements	26
3.7.1	Functional Requirements	26
3.7.2	Non-Functional Requirements.....	28
3.8	Use Case Diagram and Scenarios	30
3.8.1	Use Case Log in:.....	31
3.8.2	Use Case Sing UP:.....	33
3.8.3	Use Case Home Page:	35
3.8.4	Use Case Academic schedule:	37
3.8.5	Use Case GPA:	39
3.8.6	Use Case Attendance Tracking:	41
3.8.7	Use Case contact Ambassador	43
3.9	User Classes and Characteristics.....	45
3.9.1	Leisure Students.....	45
3.9.2	Family Students.....	45
3.9.3	Cultural Students	45
3.10	Operating Environment	46
3.10.1	Hardware	46
3.10.2	Operating System.....	46
3.10.3	Internet Connectivity	46
3.11	Design and Implementation Constraints	47
3.11.1	Scheduling System	47

3.11.2	Cross-Platform Compatibility	47
3.11.3	Performance Optimization.....	47
3.11.4	Academic Context and Accuracy	47
3.11.5	User Accessibility	48
3.11.6	Data Security and Privacy.....	48
3.11.7	Scalability.....	48
3.11.8	Device Limitations	48
3.12	User Documentation	49
3.12.1	User Manual	49
3.12.2	User Support.....	49
3.12.3	User Interface Design	49
3.12.4	Feedback System	49
3.13	Assumption	50
3.14	Dependencies	50
3.15	External Interface Requirements	51
3.15.1	User interfaces (Initial Interface)	51
3.15.2	User Interfaces (Secondary Interface)	54
3.16	Hardware Interfaces.....	57
3.16.1	Display Screen	57
3.16.2	Input Mechanisms.....	57
3.16.3	Internet Connectivity	57
3.16.4	Photo Upload Capability	57
3.16.5	GPA	58
3.17	Software Interfaces	58
3.18	System Features	58
3.18.1	Core features	58
3.18.2	User interactions features.....	58
3.18.3	Technical features	59
3.19	Conclusion	59
4	CHAPTER 4 SOFTWARE DESIGN.....	60
4.1	Introduction.....	60
4.2	System Architecture.....	61
4.3	Constraints and Assumptions	62
4.4	Interfaces and Data Stores	63
4.4.1	Dataflow Diagram	63
4.4.2	Class Diagram	64
4.5	Sequence diagram	65

4.5.1	Sequences Homepage	65
4.5.2	Sequence schedule	66
4.5.3	Sequences tracking absence	67
4.5.4	Sequences Ambassadors	68
4.6	ER diagram and relation schema	69
4.6.1	Activity Diagram schedule	69
4.6.2	Activity Diagram Homepage	70
4.6.3	Activity Ambassadors	71
4.6.4	Activity Absences	72
4.7	Conclusion	73
5	CHAPTER 5 System Implementation and Validation	74
5.1	Introduction.....	74
5.2	Technology Stack	74
5.3	Collaboration Tools	78
5.4	User Interfaces (Latest Design).....	81
5.5	Implementation Details	95
5.5.1	Welcome Screen code	95
5.5.2	Registration Screen (Sign Up) code.....	97
5.5.3	Registration Screen (Log in) code	100
5.5.4	Forget Password Screen code	103
5.5.5	Verification Screen code	105
5.5.6	New Password Screen code	107
5.5.7	Success Screen “Password Change Confirmation” code	109
5.5.8	Home Screen Main Screen code	111
5.5.9	Ambassadors Screen code	113
5.5.10	GPA Calculator Screen	115
5.5.11	Attendance Tracker Screen code.....	117
5.5.12	Profile Screen code.....	119
5.5.13	Machine Training Attempts.....	121
5.5.14	ways to create a schedule Screen code	122
5.5.15	Your Way Screen code	124
5.5.16	Popup Box Screen code	126
5.5.17	AI Planner Screen code	128
6	CHAPTER 6 Testing and Evaluation	130
6.1	Introduction.....	130
6.2	Testing.....	130
6.3	Survey Analysis	133

6.3.1	Visual Appeal and Appropriateness for Educational App	134
6.3.2	Necessity of the Login Process	135
6.3.3	Enhancing the educational Experience.....	136
6.3.4	The Possibility of Using the App during Education Planning or During Educational Experiments	137
6.3.5	Recommendation of the Application to Others.....	138
6.3.6	Experiment with calculating the rate and organizing the Schedule.....	139
6.3.7	Potential Improvements for the Schedule Management Experience	140
6.3.8	Identifying Unnecessary or Missing Sections in the Application	140
6.4	Usability Test.....	141
6.4.1	Participant Details	142
6.4.2	Task Performance Details	143
	General Usability Test Results.....	145
6.5	Future Enhancements.....	146
6.6	Conclusion	147
7	CHAPTER 7 Conclusion and Future Work.....	148
7.1	Introduction.....	148
7.2	Future Work.....	148
8	Conclusion	150
9	References	152
10	Appendix.....	158

List of Figures

FIGURE 1 METHODOLOGY	12
FIGURE 2 USE CASE DIAGRAM	30
FIGURE 3 USE CASE LOG IN	31
FIGURE 4 USE CASE SING UP	33
FIGURE 5 USE CASE HOME PAGE	35
FIGURE 6 USE CASE ACADEMIC SCHEDULE	37
FIGURE 7 USE CASE GPA	39
FIGURE 8 USE CASE ATTENDANCE TRACKING	41
FIGURE 9 USE CASE CONTACT AMBASSADOR	43
FIGURE 10 LOGIN METHOD SELECTION SCREEN	51
FIGURE 11 LOG IN SCREEN	51
FIGURE 12 SING UP SCREEN	51
FIGURE 13 VIEW SCHEDULE	51
FIGURE 14 SCHEDULE AI PLANNER.....	51
FIGURE 15 GPA CALCULATOR	51
FIGURE 16 ABSENCE SCREEN SCREEN 2	52
FIGURE 17 ABSENCE SCREEN SCREEN	52
FIGURE 18 GPA CALCULATOR 2.....	52
FIGURE 19 AMBASSADOR SCREEN	52
FIGURE 20 LOGIN METHOD SELECTION SCREEN	54
FIGURE 21 LOG IN SCREEN	54
FIGURE 22 SING UP SCREEN	54
FIGURE 23 VIEW SCHEDULE	54
FIGURE 24 SCHEDULE AI PLANNER.....	54
FIGURE 25 GPA CALCULATOR	54
FIGURE 26 ABSENCE SCREEN SCREEN 2	55
FIGURE 27 ABSENCE SCREEN SCREEN	55
FIGURE 28 GPA CALCULATOR 2.....	55
FIGURE 29 AMBASSADOR SCREEN	55
FIGURE 30 SYSTEM ARCHITECTURE	61

FIGURE 31 DATA FLOW DIAGRAM	63
FIGURE 32 CLASS DIAGRAM	64
FIGURE 33 SEQUENCES HOMEPAGE DIAGRAM	65
FIGURE 34 SCHEDULE SEQUNCE DIAGRAM.....	66
FIGURE 35 SEQUENCES TRACKING ABSENCE.....	67
FIGURE 36 SEQUENCES AMBASSADORS.....	68
FIGURE 37 ACTIVITY DIAGRAM SCHEDULE	69
FIGURE 38 ACTIVITY DIAGRAM HOMEPAGE	70
FIGURE 39 ACTIVITY DIAGRAM AMBASSADORS.....	71
FIGURE 40 ACTIVITY ABSENCES.....	72
FIGURE 41 APPLICATION LOGIN METHOD SELECTION INTERFACE	81
FIGURE 42 SING UP SCREEN	82
FIGURE 43 LOG IN SCREEN	83
FIGURE 44 FORGET PASSWORD	84
FIGURE 45 VERIFICATION.....	85
FIGURE 46 PASSWORD CHANGED SUCCESSFULLY	86
FIGURE 47 NEW PASSWORD.....	86
FIGURE 48 HOME SCREEN.....	87
FIGURE 49 WAYS TO CREATE A SCHEDULE	88
FIGURE 50 SCHEDULE YOUR WAY	88
FIGURE 51 VIEW SCHEDULE	89
FIGURE 52 INFORMATION BOX.....	89
FIGURE 53 SCHEDULE AI PLANNER.....	90
FIGURE 54 SCHEDULE AI PLANNER.....	90
FIGURE 55 ABSENCE SCREEN	91
FIGURE 56 GPA CALCULATOR	92
FIGURE 57 AMBASSADOR SCREEN	93
FIGURE 58 PROFILE	94
FIGURE 59 WELCOME SCREEN CODE	95
FIGURE 60 REGISTRATION SCREEN (SIGN UP) CODE.....	97
FIGURE 61 REGISTRATION SCREEN (LOG IN) CODE	100
FIGURE 62 FORGET PASSWORD SCREEN CODE	103
FIGURE 63 VERIFICATION SCREEN CODE.....	105
FIGURE 64 NEW PASSWORD SCREEN CODE.....	107

FIGURE 65 SUCCESS SCREEN CODE	109
FIGURE 66 HOME SCREEN MAIN SCREEN.....	111
FIGURE 67 AMBASSADORS SCREEN CODE	113
FIGURE 68 GPA CALCULATOR SCREEN.....	115
FIGURE 69 ATTENDANCE TRACKER SCREEN CODE	117
FIGURE 70 PROFILE SCREEN CODE	119
FIGURE 71 WAYS TO CREATE A SCHEDULE SCREEN CODE	122
FIGURE 72 YOUR WAY SCREEN CODE.....	124
FIGURE 73 POPUP BOX SCREEN CODE.....	126
FIGURE 74 AI PLANNER SCREEN CODE	128
FIGURE 75 VISIBLE GRAVITY AND RATIO CHART	134
FIGURE 76 NECESSITY OF THE LOGIN RATIO CHART	135
FIGURE 77 ENHANCING THE EDUCATIONAL EXPERIENCE RATIO CHART.....	136
FIGURE 78 LIKELIHOOD OF APPLICATION USAGE RATIO CHART.....	137
FIGURE 79 RECOMMENDATION OF THE APPLICATION RATIO CHART	138
FIGURE 80 IMMERSIVE EXPERIENCE RATIO CHART.....	139
FIGURE 81 SCREENSHOT OF GOOGLE SHEETS.....	158
FIGURE 82 SCREENSHOT OF MICROSOFT EXCEL	158
FIGURE 83 SCREENSHOT OF DISCORD	159
FIGURE 84 SCREENSHOT OF JIRA	159
FIGURE 85 SCREENSHOT OF GOOGLE DRIVE.....	160
FIGURE 86 SCREENSHOT OF TRELLO	160
FIGURE 87 SCREENSHOT OF GITHUB 1	161
FIGURE 88 SCREENSHOT OF GITHUB 2	161

List of Tables

TABLE 1 OVERVIEW OF EXISTING PROJECTS	16
TABLE 2 FUNCTIONAL REQUIREMENTS	26
TABLE 3 NON-FUNCTIONAL REQUIREMENTS	28
TABLE 4 LOG IN PAGE SCENARIO.....	32
TABLE 5 SIGN UP SCENARIO.....	34
TABLE 6 HOME PAGE SCENARIO	36
TABLE 7 SCENARIO SCHEDULE	38
TABLE 8 GPA SCENARIO	40
TABLE 9 SCENARIO ATTENDANCE TRACKING	42
TABLE 10 SCENARIO CONTACT AMBASSADOR	44
TABLE 11 SOFTWARE TESTING.....	131
TABLE 12 SOFTWARE TESTING2	132
TABLE 13 PARTICIPANT DETAILS.....	142
TABLE 14 TASK PERFORMANCE DETAILS 1	143
TABLE 15 TASK PERFORMANCE DETAILS 2	144

1 CHAPTER 1 INTRODUCTION

1.0 Introduction

The Muad application offers a transformative approach to managing academic life, leveraging cutting-edge AI technology to empower students in achieving academic success. This report details the development of the application, its innovative features, and its potential impact on simplifying education through smart tools and personalized support.

The application is designed to streamline academic processes by providing AI-powered class scheduling, attendance tracking, GPA calculation. These features ensure that students can effectively organize their time, track their progress, and stay on top of their academic commitments. Through its intelligent planning tools, Muad fosters a more efficient and engaging learning experience.

This chapter explores the problem domain, outlines the problem statement, and presents the aims, objectives, and methodology of the proposed system. By integrating advanced technologies and focusing on user-centric design, the application aspires to revolutionize academic management. Muad emphasizes customization and interactivity, offering students a tailored platform that adapts to their unique needs. With its comprehensive approach, the project aims to set a new standard for academic success and time optimization.

1.1 Problem domain

The problem lies in the significant time and effort students spend managing their academic responsibilities, such as organizing schedules, tracking attendance, and monitoring academic progress. Another challenge is to develop a system that enables students to streamline and customize their academic activities using smart tools. This includes creating dynamic and personalized schedules as well as seamlessly tracking their academic performance.

Overall, the goal in this problem domain is to create an engaging and user-friendly platform that allows students to fully manage and optimize their academic lives in a natural and intuitive way, leveraging AI-powered solutions to enhance organization, save time, and boost academic success

1.2 Problem statement

Managing academic responsibilities can be a challenge for students due to the fragmented nature of existing tools. However, we aim to address this issue by offering an intuitive application that simplifies schedule management, attendance tracking, and GPA calculation, enhancing students' academic experiences.

This solution leverages AI technology to provide personalized reminders and insights, empowering students to make smarter decisions and stay organized. Through its user-friendly interface and advanced features, the application streamlines academic tasks, allowing students to focus on their goals and achieve

1.3 Proposed system

Managing academic responsibilities can often feel overwhelming for students, but now there's a solution: a smart academic management application. This app leverages AI technology to simplify essential tasks like organizing schedules, tracking attendance, and calculating GPA.

In addition to these core features, the application provides personalized reminders and actionable insights, empowering students to make informed decisions about their academic journey. In this report, we'll explore how this system enhances the academic experience by application Muad

1.4 Aims and Objectives

An AI-powered application designed to facilitate students' academic management through the following measurable objectives:

1. AI-powered study schedules: The system will create personalized study plans for each student, including notes, preferred study times, and extracurricular activities. The success of this feature will be measured by at least a 20% improvement in students' study time and adherence to the schedule during the first semester of use.
2. Attendance and absence monitoring: The platform will track and report class attendance in real-time, calculating absence rates with 95% accuracy.
3. GPA calculation: The application will provide tools to calculate cumulative and semester GPA with an accuracy rate of at least 90%.

1.5 Proposed system features

1.5.1 Planning and Requirements Gathering:

The system will begin with a structured planning phase aimed at defining the project scope, goals, and deliverables. This phase will involve gathering detailed requirements from stakeholders, establishing timelines, and creating a high-level plan to guide development. By focusing on precise planning, the foundation for a successful and organized project will be established.

1.5.2 System Design:

To ensure a seamless user experience, the system will incorporate a well-structured design phase. This will include creating the system architecture, defining components, and developing interactive prototypes. These designs will be carefully reviewed to ensure alignment with user needs and project objectives, providing a clear blueprint for the development phase.

1.5.3 Development:

The development phase will focus on building the core features of the application, including the back-end, front-end, and AI-powered functionalities. Advanced APIs will be integrated to enable smooth interactions between system components, ensuring the application is robust and scalable.

1.5.4 Testing:

The system will undergo rigorous testing to ensure its reliability and performance. Testing activities will include unit testing for individual components, integration testing for system interoperability, and user acceptance testing to validate the application's functionality and usability from the end-user perspective.

1.5.5 Deployment:

The application will be launched in a phased manner to ensure a smooth transition for users. During this phase, the system's performance will be closely monitored, and initial feedback will be gathered to address any immediate concerns or necessary adjustments.

1.5.6 Maintenance and Updates:

Post-deployment, the system will enter a continuous improvement cycle. This phase will involve regular updates to enhance features, resolve issues, and optimize performance. Support will be provided to users, ensuring the system remains effective and up-to-date over time.

By following these structured development phases, the application will deliver a reliable, user-friendly, and innovative solution to meet student needs effectively.

1.6 Project Methodology

The Agile methodology was used in a project. It is a framework for software development and project management that divides the project into small, manageable units called “Sprints,” lasting from two to four weeks, with the goal of producing usable parts of the final product. Agile focuses on continuous collaboration with the client, quick response to changes, regular delivery, enhanced team collaboration, and continuous performance improvement to achieve goals flexibly and efficiently.

How to Use Agile Methodology in a Muad Project:

1. Dividing the Project into Sprints:

- We divide the graduation project into small tasks completed over a period of two to four weeks, such as initial research preparation, developing prototypes, and system testing.

2. Holding Daily Stand-up Meetings:

- We hold short daily meetings with the team (or conduct a personal review when working individually) to review what we have accomplished, identify upcoming tasks, and address potential challenges.

3. Sprint Reviews:

- We review what we have completed at the end of each sprint with the project supervisor or other members to gather feedback and improve work for the next sprint.

4. Receiving Feedback:

- We receive feedback from supervisors or peers at the end of each sprint to determine if we are on the right track or need adjustments.

5. Continuous Planning (Sprint Planning):

- We plan for each sprint before starting, set the tasks we will work on, and estimate the time needed for completion.

6. Continuous Improvement (Retrospectives):

- We meet at the end of each sprint to discuss what went well and what needs improvement, and apply the lessons learned in the next sprint

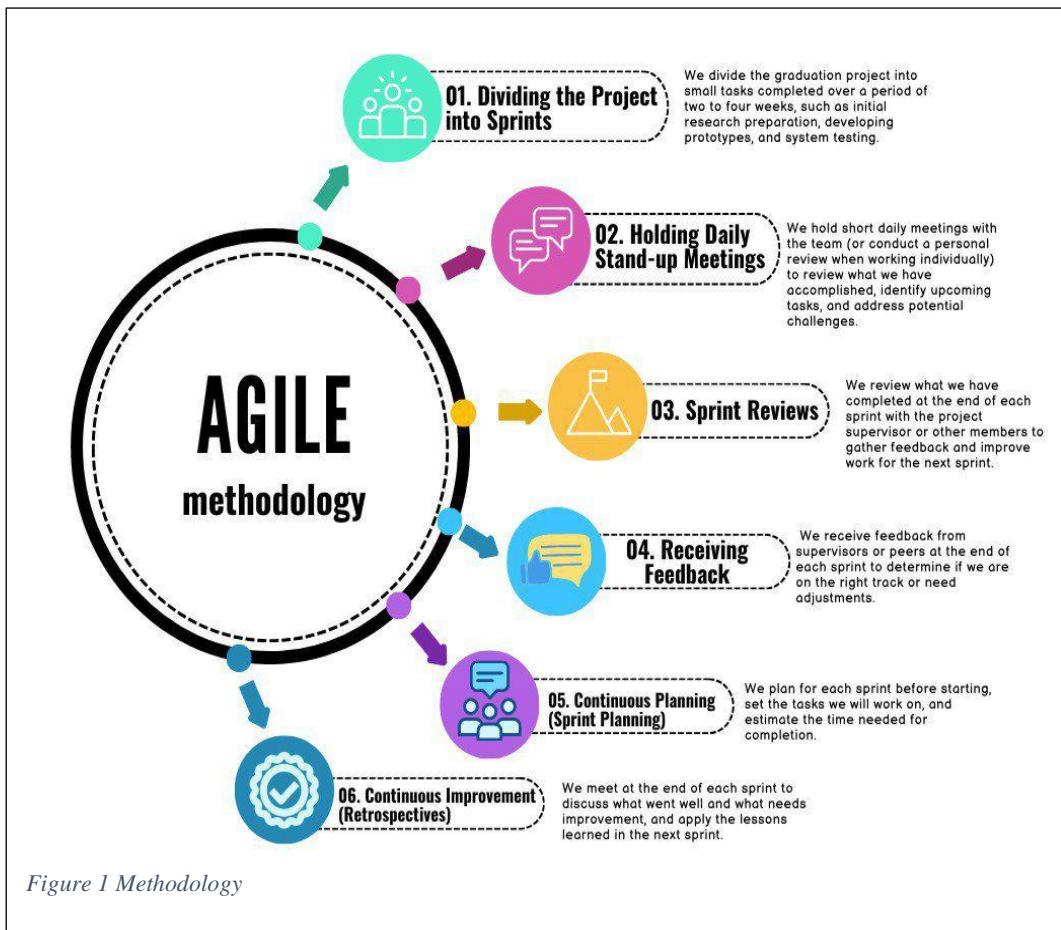


Figure 1 Methodology

1.7 Report Layout

This report is divided into six chapters, each addressing different aspects of the project.

Chapter 1: Introduction

The first chapter introduces the project, outlining the problem it aims to solve, the objectives of the system, and the resources required for its development. This chapter sets the context for the entire report and provides an overview of the project's scope.

Chapter 2: Background and Related Work

The second chapter offers a comprehensive review of the background information and existing work relevant to the project. This includes a comparison of similar projects, presented through a comparison table, to highlight differences and similarities with the proposed system.

Chapter 3: System Scenarios and Requirements

Chapter three details the project's scenarios, describing both functional and non-functional requirements. It provides a clear understanding of the system's expected behavior, ensuring that all necessary features and performance standards are addressed.

Chapter 4: Software Design

The fourth chapter covers the software design, including various diagrams such as activity diagrams, sequence diagrams, use case diagrams, and context diagrams. It also includes the structural design and system architecture, as well as a detailed explanation of the interfaces and data stores used in the project.

1.8 Conclusion

In conclusion, Chapter 1 provided a comprehensive introduction to the Muad application, a project designed to address the challenges students face in managing their academic responsibilities. The chapter highlighted the problem area, emphasizing the difficulties students face in organizing schedules, tracking attendance, and monitoring academic progress.

The proposed system was introduced, showcasing the integration of smart tools for schedule management, attendance tracking, and calculating cumulative and semester GPAs. The system's features ensure that students have a seamless and engaging experience that supports their academic journey.

The chapter outlined the project objectives, focusing on creating an easy-to-use platform that simplifies students' academic management, enhances time management, and promotes overall academic success. The Agile methodology was also adopted, ensuring flexibility and continuous collaboration with stakeholders to meet changing requirements and deliver a high-quality product.

Finally, the report outline was presented, providing a clear structure for subsequent chapters, each focusing on different aspects of system development, design, and evaluation. This structure lays the foundation for a detailed exploration of the background research, system design, and technology implementation, leading to a comprehensive understanding of the project's impact.

2 CHAPTER 2 BACKGROUND/EXISTING WORK

2.1 Introduction

This chapter provides a detailed literature review exploring existing systems and research studies in the field of student support applications. The review is based on extensive research conducted using reputable academic search engines such as Google Scholar , IEEE Xplore , and PubMed , along with other trusted sources.

The review highlights the challenges and limitations of current systems, focusing on how the Muad application offers innovative solutions tailored to the needs of university students. Key features of the application include absence tracking, GPA calculation, schedule creation, and communication with academic ambassadors.

Additionally, this chapter provides a comprehensive background on similar projects and includes a comparison table that highlights the differences and innovations offered by the Muad application. This comparison demonstrates the app's added value in enhancing students' academic experiences and effectively managing their time

2.2 Overview of existing projects

Comparison Criteria	My App	UQU App	GPA Calculator	Schedule App	Calendar & Schedule App	Najran Uni App	Bisha Uni App
Technology	AI-Powered	Basic Features	Basic Features	Basic Features	Basic Features	Basic Features	Basic Features
Scope	All Students (Future Integration with UQU)	Umm Al-Qura	Global Users	General Students	General Students	Najran University	Bisha University
User- Friendly	✓	✓	✓	✓	✓	✓	✓
Interactive	✓	✓	✗	✓	✓	✓	✓
Privacy Policy	✓	✓	✓	✓	✓	✓	✓
Immersive	✓	✗	✗	✓	✓	✗	✗
Attendance Alerts	✓ (Orange & Red Alerts)	✗	✗	✗	✗	✗	✗
Academic Calculation	Semester + Cumulative	Cumulative Only	Semester + Cumulative	✗	✗	✗	✗
Ambassador Communication	✓	✗	✗	✗	✗	✗	✗

Table 1 overview of existing projects

The comparison table provides a detailed analysis of our app compared to similar applications, focusing on aspects such as technology, target audience, and core features. Our app stands out by leveraging AI-powered technology, offering intelligent schedule creation and GPA prediction, unlike other apps that rely solely on basic functionalities. It caters to all students across various universities, with a future plan to integrate with Umm Al-Qura University for tailored services. While all apps feature user-friendly interfaces, our app excels in interactivity. Additionally, it delivers an immersive experience through advanced data analysis and innovative features, setting it apart from competitors that lack such capabilities.

2.3 Limitations of existing projects

Several existing projects, as summarized in Table 1, exhibit notable limitations that impact their effectiveness and usability. These limitations include the following:

1. Limited Technology Advancement:

- Many existing applications, such as the UQU App and Najran Uni App, rely on basic features, lacking the advanced functionality offered by AI-powered systems. This restricts their ability to adapt to user needs dynamically.

2. Restricted Scope:

- Most applications serve a narrow audience. For instance, the UQU App is limited to Umm Al-Qura University students, while the Bisha Uni App and Najran Uni App are university-specific, reducing their utility for a broader range of students.

3. Lack of Interactivity:

- Applications like the GPA Calculator and the Schedule App are not interactive, limiting user engagement. This makes the user experience less engaging and fails to foster a sense of involvement.

4. Insufficient Privacy Measures :

- Privacy policies are often absent or inadequate in many of the applications, which can lead to concerns over the safety and confidentiality of user data.

5. Non-Immersive Experience :

- Most applications fail to provide immersive features, with the exception of "My App," which integrates advanced capabilities. The lack of immersive tools diminishes user engagement, especially for applications targeting tourists or students.

6. Absence of Attendance Alerts:

- Many of the existing applications, such as the UQU App, GPA Calculator, and others, do not include attendance tracking or alert systems. This absence limits their ability to notify students when they approach or exceed attendance thresholds, such as orange and red alerts.

7. Incomplete Academic Calculations:

-While some applications focus on academic calculations, they are limited in scope. For instance, the GPA Calculator handles cumulative calculations only, whereas "My App" provides both semester and cumulative GPA tracking with enhanced precision and prediction capabilities.

8. Lack of Ambassador Communication Features:

-None of the existing applications, apart from "My App," offer features that facilitate direct communication with student ambassadors or mentors. This reduces the ability to provide personalized guidance for students regarding courses, training, and academic planning.

2.4 Innovations of our project

The proposed system, Muad, integrates the key features of existing applications while leveraging AI-powered solutions to provide an advanced, user-friendly, and personalized experience for students. Through smart scheduling, GPA calculation, absence tracking, and enhanced engagement, the application ensures a smooth and engaging academic management process. A strong focus on user privacy through a comprehensive privacy policy ensures secure handling of user data, making Maud a reliable and innovative tool for students to excel in their educational journey.

2.5 Conclusion

In conclusion, Chapter 2 provided an overview of existing projects aimed at supporting students' academic journeys, highlighting their strengths and limitations. The comparison of similar applications, such as GPA Calculator, Schedule App, and university-specific apps like Najran Uni and Bisha Uni, showcased how these systems offer varying levels of functionality. However, critical gaps were identified in many of these projects, such as limited interactivity, a lack of advanced features like AI-based scheduling and Ambassadors contact, the absence of robust privacy policies that ensure data security.

Muad leverages the strengths of these existing systems by integrating advanced AI-driven features, including smart scheduling, GPA calculation, absence tracking, and enhanced interactivity. Its comprehensive privacy policy ensures secure data handling, while its user-friendly interface provides an engaging and efficient academic management experience tailored for students. This unique focus on innovation and personalization distinguishes Muad from similar applications, addressing the gaps found in other solutions and empowering students to manage their academic lives effectively.

This chapter lays the foundation for the detailed discussions in subsequent chapters regarding the functionality, design, and development of Muad, highlighting its potential to transform how students approach academic planning and success.

3 CHAPTER 3 SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Introduction

The Muad application provides an innovative and intelligent platform for students to efficiently manage their academic journey. This document serves as a comprehensive guide for the development team, stakeholders, and users, offering detailed insights into the app's objectives, features, and technical specifications. By leveraging AI-powered tools, Muad aims to simplify complex academic tasks, such as course scheduling, GPA calculation, communicate with ambassadors, and absence tracking, enhancing productivity and user experience.

With its intuitive interface and personalized features, Muad bridges gaps in academic management, empowering students to organize their studies effectively. The app's engaging and user-friendly design ensures accessibility and ease of use for all students, regardless of their academic level or university. Through its innovative approach, Muad strives to redefine the student experience, making academic planning seamless and more efficient

3.1.1 Purpose

The purpose of this chapter is to outline the specific requirements and conventions for the Muad Project Software Requirements Specification document. It will cover the structure, format, and language used throughout the document to ensure clarity and consistency. This chapter will also define the key components and features of the Muad application that will be detailed in subsequent chapters, including AI-powered scheduling, attendance tracking, GPA calculation, and communicate with ambassadors.

3.2 Document Conventions

The Software Requirements Specification (SRS) document for the Muad project will adhere to specific conventions to ensure clarity, consistency, and accessibility for all stakeholders:

3.2.1 Document Format

The SRS document will be created using commonly used tools such as Microsoft Word or Google Docs to ensure it is easily accessible and reviewable by all stakeholders.

3.2.2 Language

The document will use clear, straightforward language, avoiding complex technical jargon. This ensures it is understandable for all stakeholders, including those without a technical background.

3.2.3 Section Headings

Each section will have well-defined and descriptive headings, enhancing readability and making it easier to locate relevant information within the document.

3.2.4 Numbering System

Sections and subsections will follow a consistent numbering system, allowing precise referencing of specific requirements throughout the document.

3.2.5 Version Control

A version control process will be implemented to track document updates and ensure all stakeholders have access to the latest version. This practice promotes transparency and accountability.

3.2.6 Stakeholder Signatures

To confirm agreement and commitment to the specified requirements, team members, as key stakeholders in this project, will sign the document upon completion. This signing process reflects collective approval and ensures alignment with the project's goals and expectations.

3.3 Intended Audience and Reading Suggestions

3.3.1 Project Stakeholders

Individuals or organizations invested in the success of the Muad project. This may include university administrators, academic advisors, and project sponsors who aim to enhance student experiences through innovative academic tools.

3.3.2 Development Team

The team responsible for designing, developing, and implementing the application. This includes software engineers, developers, testers, and quality assurance personnel, working collaboratively to deliver a robust and efficient product.

3.3.3 End Users

Students across various educational institutions who will use Muad to manage their academic schedules, calculate GPAs, communicate with ambassadors, and track attendance. These users have diverse needs, and the app is tailored to offer personalized experiences to meet their academic goals.

3.3.4 Academic Advisors and Educational Experts

Professionals specializing in academic advising, curriculum planning, or student engagement. These individuals can utilize Muad to provide insights, guide students effectively, and analyse academic progress, ensuring alignment with educational objectives. Their feedback is valuable in refining the app's features to meet institutional and student expectations.

3.4 Suggestions for the software requirements specification document include

3.4.1 Provide an Overview

Start the document with a comprehensive summary that outlines the key features, objectives, and advantages of the Muad application. This will give stakeholders a high-level understanding of the app's purpose and the goals of the project.

3.4.2 Use Clear Language

Ensure the document is written in simple, clear language that can be easily understood by both technical and non-technical stakeholders. Avoid unnecessary technical jargon to maintain accessibility.

3.4.3 Organize the Document Effectively

Use headings, subheadings, and a detailed table of contents to structure the document. This will enhance readability and make it easier for stakeholders to locate specific information.

3.4.4 Include Functional Requirements

Define each functional requirement with precision, specifying its purpose, inputs, outputs, and any relevant rules or dependencies. Include diagrams or mock-ups to visualize complex features such as AI scheduling, GPA calculations, and attendance tracking.

3.4.5 Address Non-Functional Requirements

List non-functional requirements such as performance benchmarks, reliability standards, data security, and scalability expectations. Include measurable metrics to evaluate the system's performance and compliance with these requirements.

3.4.6 Provide a Glossary of Terms

Incorporate a glossary defining key terms and acronyms, such as "AI scheduling," "GPA," and "attendance tracking." This will assist readers unfamiliar with technical or academic terminology.

3.4.7 Include Appendices

Attach additional resources such as wireframes, user journey maps, or use cases in the appendices. This will provide stakeholders with supplementary details to deepen their understanding of the application's design and functionality.

3.4.8 Seek Feedback and Clarification

Encourage stakeholders, including students, advisors, and university administrators, to review the document and provide feedback. Address any concerns or questions to ensure the document accurately reflects their expectations and requirements for the Muad application

3.5 Product Scope

The Muad platform aims to revolutionize academic management by providing AI-powered tools for personalized scheduling, GPA tracking, communicate with ambassadors, and attendance monitoring. It connects students with efficient academic planning and progress tracking features, ensuring a seamless and productive educational journey. The Muad application guarantees a comprehensive and engaging experience for students, enhancing their time management and academic performance while offering tailored recommendations to support their goals. Through these innovative features, Muad ensures that every student can achieve their academic aspirations with confidence and ease.

3.6 Product Perspective

The Muad app is a comprehensive AI-powered platform designed to help students manage their academic journey effectively. It offers a range of features such as personalized study schedules, attendance monitoring, communicate with ambassadors, and grade calculation, allowing students to improve their study habits and stay on top of their academic progress. The app also enables seamless communication with ambassadors, providing guidance and support throughout their educational experience. By integrating these features into a single platform, Muad aims to enhance the student experience and foster academic success.

3.7 Project requirements

3.7.1 Functional Requirements

ID	Functional Requirement	Description
1.1	User Registration	The system displays a registration form for the user to fill in his personal information (e.g., first name, last name, email, password).
1.2	Email Verification	The system verifies the email address entered by the user.
1.3	Save User Data	The system securely saves the user's information in the database.
1.4	Redirect to Home	The system redirects the user to the home page after successful registration.
2.1	User Login Form	The system displays a login form for the user to enter his email and password.
2.2	Verify Credentials	The system verifies the user's credentials and redirects him to the home page upon successful login.
3.1	Upload Schedule	The system allows the user to upload an image containing course details to create a schedule.
3.2	Image Analysis	The system analyzes the uploaded image and extracts relevant course information.
3.3	Custom Schedule	The system provides a customized schedule based on the user's preferences and extracted data.
4.1	Track Attendance	The system should allow users to specify their attendance for a particular course.
4.2	Display Attendance	The system should display attendance records in a summary format.
5.1	Input Grades	The system should allow the user to enter their grades and credit hours for the course.
5.2	Calculate GPA	The system should calculate the GPA based on the entered data and display it to the user.
5.3	Store GPA History	The system should store historical GPA data for comparison and tracking.

Table 2 Functional Requirements

(Table 2 – Functional Requirements) outlines the key functional requirements of the application, focusing on essential features such as user registration, login, schedule management, attendance tracking, and GPA calculation.

The registration process allows users to enter their personal details, with the system verifying the provided email, securely saving the user's information, and redirecting them to the homepage. For existing users, the login functionality ensures secure access by verifying credentials and providing seamless navigation to the home page.

Schedule management includes the ability to upload images containing course details, which the system analyses to extract relevant information, ultimately offering a customized schedule based on user preferences. Attendance tracking enables users to record their attendance for specific courses, with the system summarizing these records for easy viewing.

The GPA calculation feature allows users to input their grades and credit hours, with the system computing the GPA and displaying it to the user. Additionally, historical GPA data is stored to facilitate comparison and tracking over time.

3.7.2 Non-Functional Requirements

1	Performance	- The system shall load pages in less than 2 seconds. -The system should ensure a fast and responsive user experience.
2	Performance	-User inputs (e.g., dates, preferences) shall be processed in less than 1 second. -Ensure that all interactions with the application are immediate to improve user experience.
3	Security	-All sensitive data must be encrypted during transmission and storage. -User data like passwords and personal information must be securely encrypted to maintain confidentiality.
4	Scalability	-The backend must be able to handle an increase in traffic without performance degradation. -The backend should be scalable to handle a larger user base without performance issues.
5	Compatibility	-The system shall support a range of devices and operating systems. -The application should work across different platforms (Android)
6	Availability	-The system should be available 99.9% of the time, with scheduled maintenance communicated to users. -The application should be almost always available with prior communication regarding downtime for maintenance.

Table 3 Non-Functional Requirements

(Table 3 – Non-Functional Requirements) outlines the key non-functional requirements that ensure optimal performance, reliability, and an enhanced user experience for the application.

The application should provide fast loading times and responsive interactions to ensure a seamless user experience. It must also support a large number of concurrent users without significant slowdowns.

The system should minimize crashes, maintain session continuity after errors, and safeguard sensitive user data using encryption and access controls.

The application ensures secure operations and prevents unauthorized access, offering robust data protection mechanisms.

The system should accommodate increased user traffic and new features without compromising performance or requiring major architectural changes.

The application adheres to WCAG 2.1 guidelines by providing features such as text alternatives, keyboard navigation, and contrast adjustments, ensuring inclusivity for all users.

Clear navigation and intuitive controls should be implemented based on established design principles to enhance the user experience.

The application should be accessible 24/7, supported by robust backup and recovery systems to minimize downtime, with planned maintenance communicated in advance.

3.8 Use Case Diagram and Scenarios

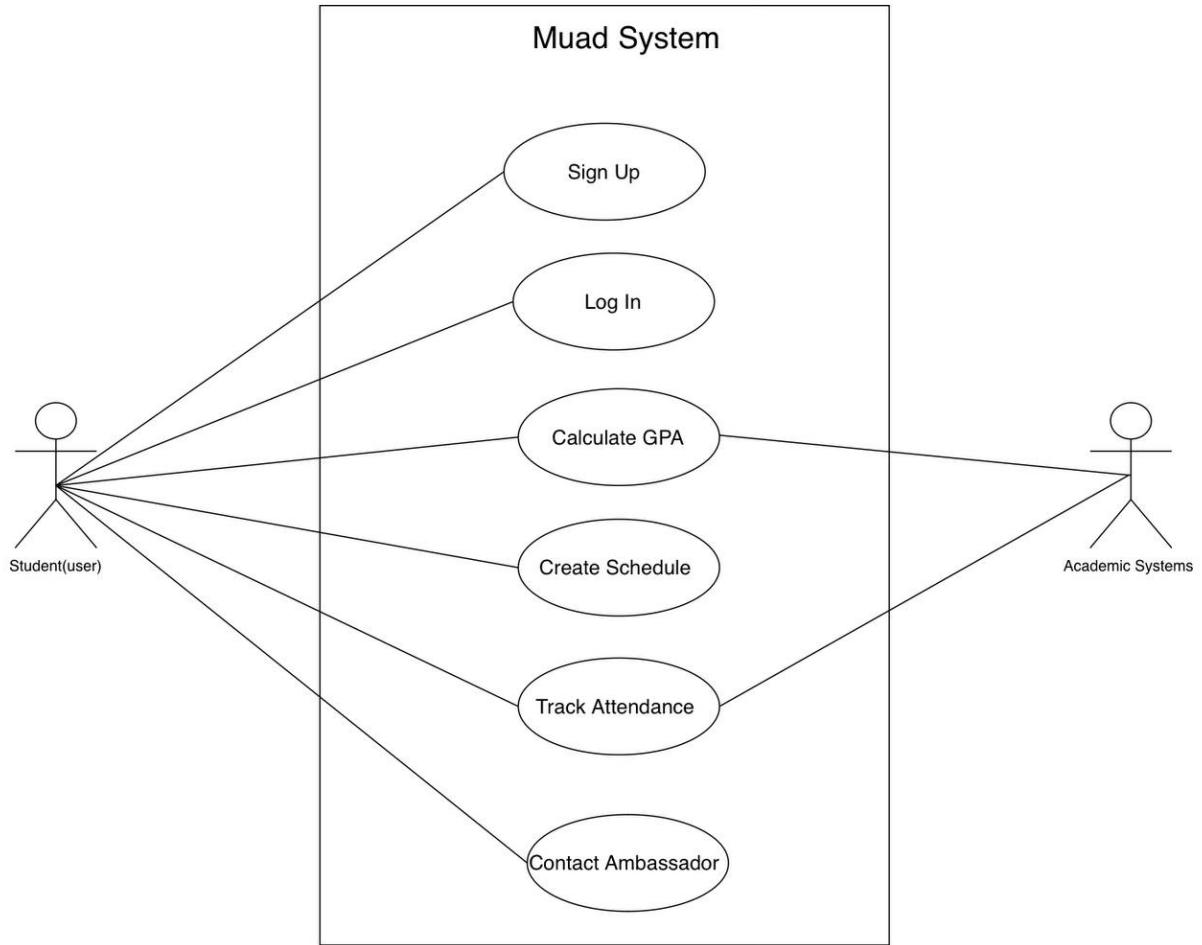


Figure 2 Use Case Diagram

The **use case diagram** illustrates the interactions between the user (student) and the "Muad System," as well as its communication with the system. The diagram includes the core processes that the student can perform, such as registration, login, GPA calculation, schedule creation, attendance tracking, and contacting the ambassador. It highlights the relationship between the student and the academic system and outlines the key functionalities that the system provides to the user.

3.8.1 Use Case Log in:

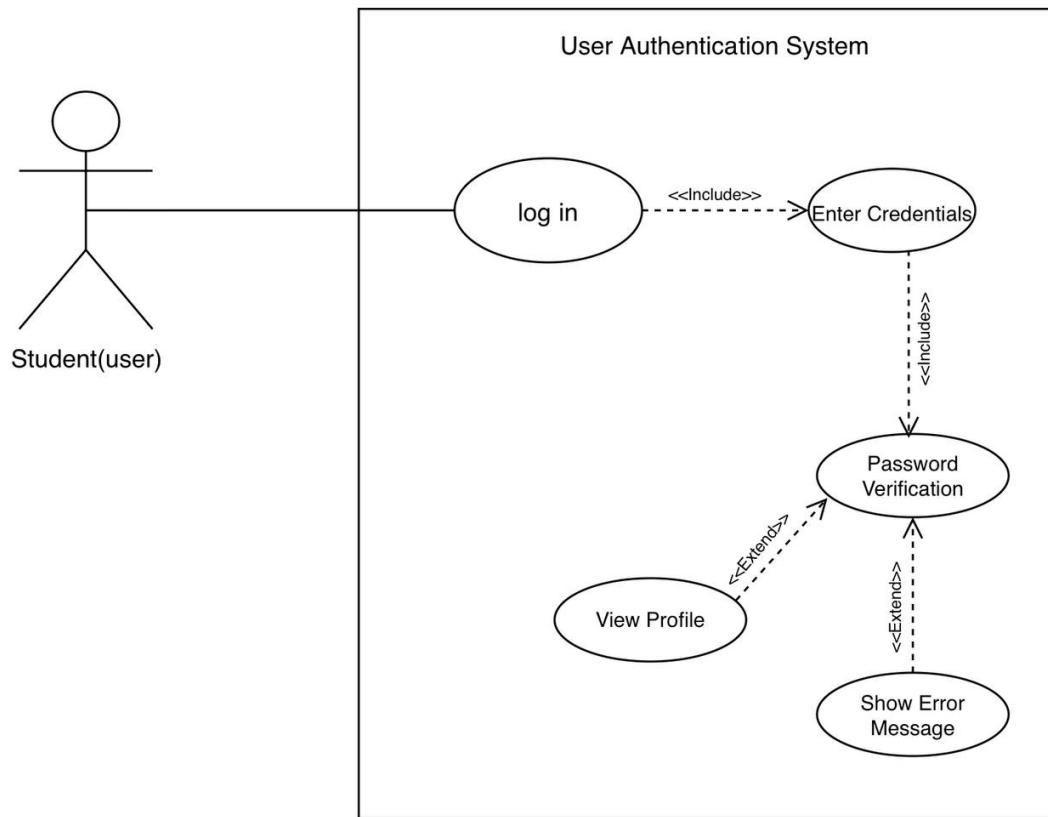


Figure 3 Use case log in

The use case diagram for the Log in illustrates the interactions between the student and the user authentication system. The student initiates the process by selecting the "Log in" option, which includes entering their credentials (email or phone number and password). The system then performs password verification. If the credentials are correct, the student can view their profile. If the verification fails, the system extends to show an error message. This diagram effectively represents a secure and straightforward user authentication process, highlighting both successful and error-handling scenarios.

Principal actor	Student
Description	The system displays a login form with two fields: email or phone number, and password. The system verifies the entered data in the database. If the data matches, the user information is retrieved, and the system navigates to the main interface. If no match is found, a pop-up message appears stating, "There is no such account, try again or sign up." After logging in, a "Forgot your password?" link appears below the password field. Clicking this link displays two boxes for verification, where the user can choose to verify either by phone number or email after checking the provided information.
Data	-Client information (email or phone number, password)
Pre-condition	- The user has an existing account and clicks on the login button
User Action	- The user enters credentials (email/phone and password) and clicks login
System Response	- The system checks the entered credentials against the database
Output	- If valid, the user is redirected to the home page. If invalid, an error message is shown with an option to sign up.
Comments	- All client data must be stored in the database and can only be accessed by authorized managers

Table 4 Log in Page Scenario

Scenario login This table describes how the student interacts with the system to access their account. The system initially displays a login form requiring the user to input their email or phone number and password. The system verifies the entered credentials against the database. If the credentials are valid, the user is directed to the home page. If the verification fails, an error message appears prompting the user to try again or sign up. Additionally, a "Forgot your password?" option below the password field enables users to reset their password through phone or email verification. All client data is securely stored, accessible only by authorized managers, ensuring privacy and data protection.

3.8.2 Use Case Sing UP:

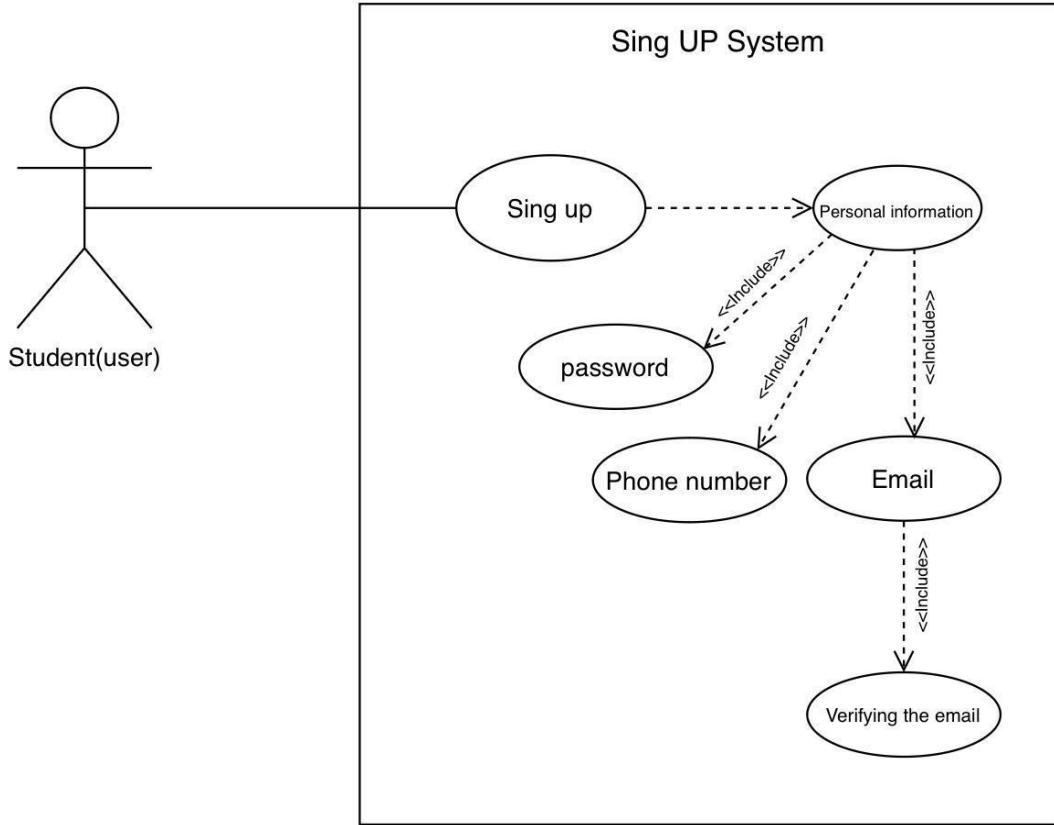


Figure 4 Use Case Sing UP

The use case diagram User Sign Up illustrates the interactions between the student (user) and the system, focusing on the essential steps required to sign up. The student, acting as the primary user, initiates the registration process by selecting the “Sign up” option. This process includes providing personal information, setting a password, entering an email address, and adding a phone number. The system then prompts the student to verify their email and phone number to complete the registration. This sequence ensures that the student’s account is created securely and verified, allowing them to access the system’s features with a confirmed identity.

Principal actor	Student
Description	The system displays a registration form with fields for first name, last name, email, phone number, and password. The system validates the entered data with specific constraints: email and phone number must be unique and not duplicated in the database, and the password must be at least 8 characters long, containing uppercase and lowercase letters, numbers, and at least one special character. The system verifies whether the email or phone number already exists in the database. If the data is valid, the user is redirected to the login page. The system stores the registration information securely in the database.
Data	-Client personal information.
Pre-condition	-clicking on the "Sign Up" button.
Output	-move to the home page
Comments	All client data must be stored in the database and can only be accessed by managers

Table 5 Sign Up Scenario

Scenario Sign Up This table describes the functionality of the Sign-Up feature within the application, which enables new users to create an account securely. The principal actor is the student, who initiates the registration process. The system displays a registration form requiring the user to enter specific information, including their first name, last name, email, phone number, and password. During the process, the system validates the provided data based on certain constraints: the email and phone number must be unique, and the password must be at least 8 characters long, containing both uppercase and lowercase letters, numbers, and at least one special character. If the email or phone number already exists in the database, the system redirects the user to the log-in page. Once all data is validated, the system stores the information in the database. This scenario begins when the student clicks the “Sign Up” button, and upon successful registration, they are redirected to the home page. The system ensures that all user data is stored securely and can only be accessed by authorized managers, thus prioritizing data privacy and security.

3.8.3 Use Case Home Page:

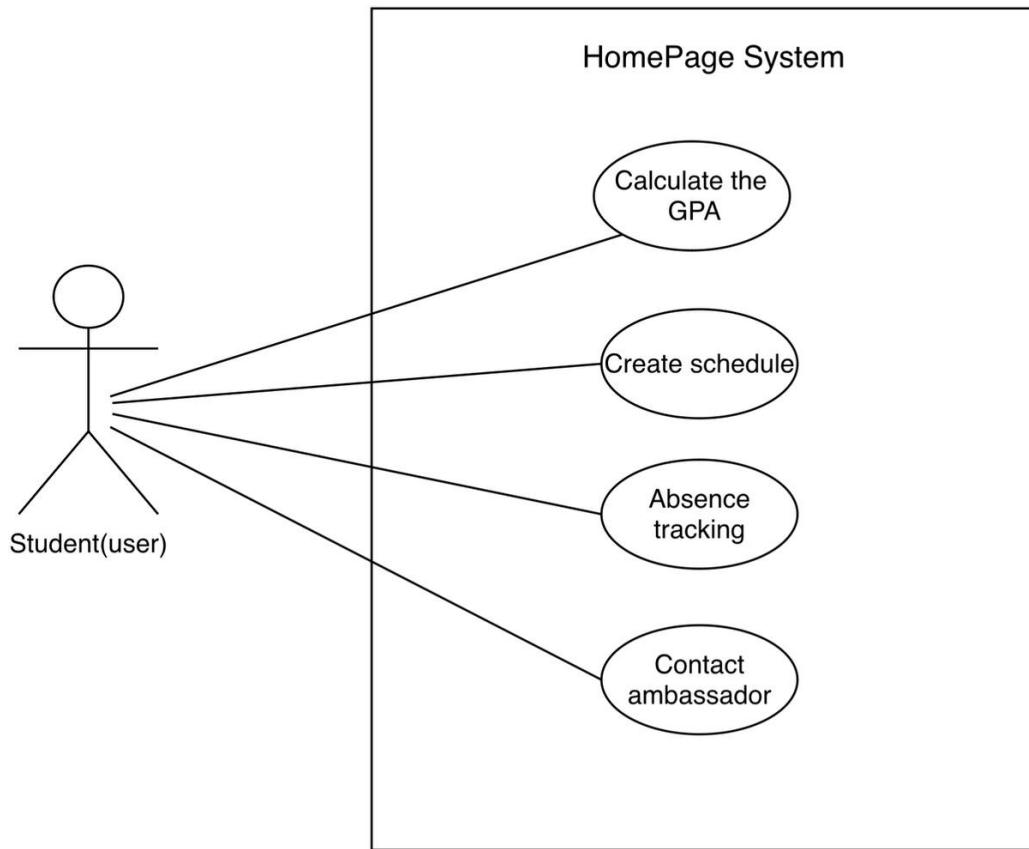


Figure 5 Use Case Home Page

The use case diagram Home Page The system interacts with the student through various use cases such as viewing GPA, contacting the ambassador, viewing the absence rate, going to the university GPA calculator, and creating the schedule. There is an include relationship between “Go to University GPA Calculator” and “Calculate University GPA,” meaning the calculation happens automatically when navigating to the page.

Principal actor	Student
Description	The student accesses the home page, which displays simplified boxes for key features of the application. These boxes allow the student to quickly navigate to main features such as GPA calculation, communication with ambassadors, absence percentage, university GPA calculation, and study schedule creation. Each box provides easy access to the corresponding functionality with minimal effort.
Data	-GPA -Absence Percentage -Ambassador Contact Information - University GPA Data -Study Schedule Data
Pre-condition	- The student must be logged into the application. -The app must be up-to-date with relevant data such as the student's GPA, attendance, and course information.
Output	-A home page with the following interactive boxes: -GPA Calculation Box: Takes the user to the GPA calculation interface. - Ambassador Contact Box: Opens the communication interface with ambassadors. - Absence Percentage Box: Displays the student's absence percentage and leads to the attendance page. -University GPA Calculation Box: Directs the user to the university-specific GPA calculation page. -Create Study Schedule Box: Navigates the user to the page for creating a study schedule.
Comments	This design aims to provide the student with quick access to essential academic features in the app, simplifying navigation and enhancing usability. The home page acts as a dashboard for easy access to core functionalities.

Table 6 Home page Scenario

Scenario Home page This table describes the functionality, student accesses to home page of the application, where simplified boxes representing the main features of the app are displayed. These boxes include GPA calculation, communication with ambassadors, absence percentage, university GPA calculation, and study schedule creation. Each box allows the student to easily navigate to the corresponding function with a single tap. The goal of this design is to make navigation within the app easier and enhance the user experience by providing key functions in a single, easily accessible interface.

3.8.4 Use Case Academic schedule:

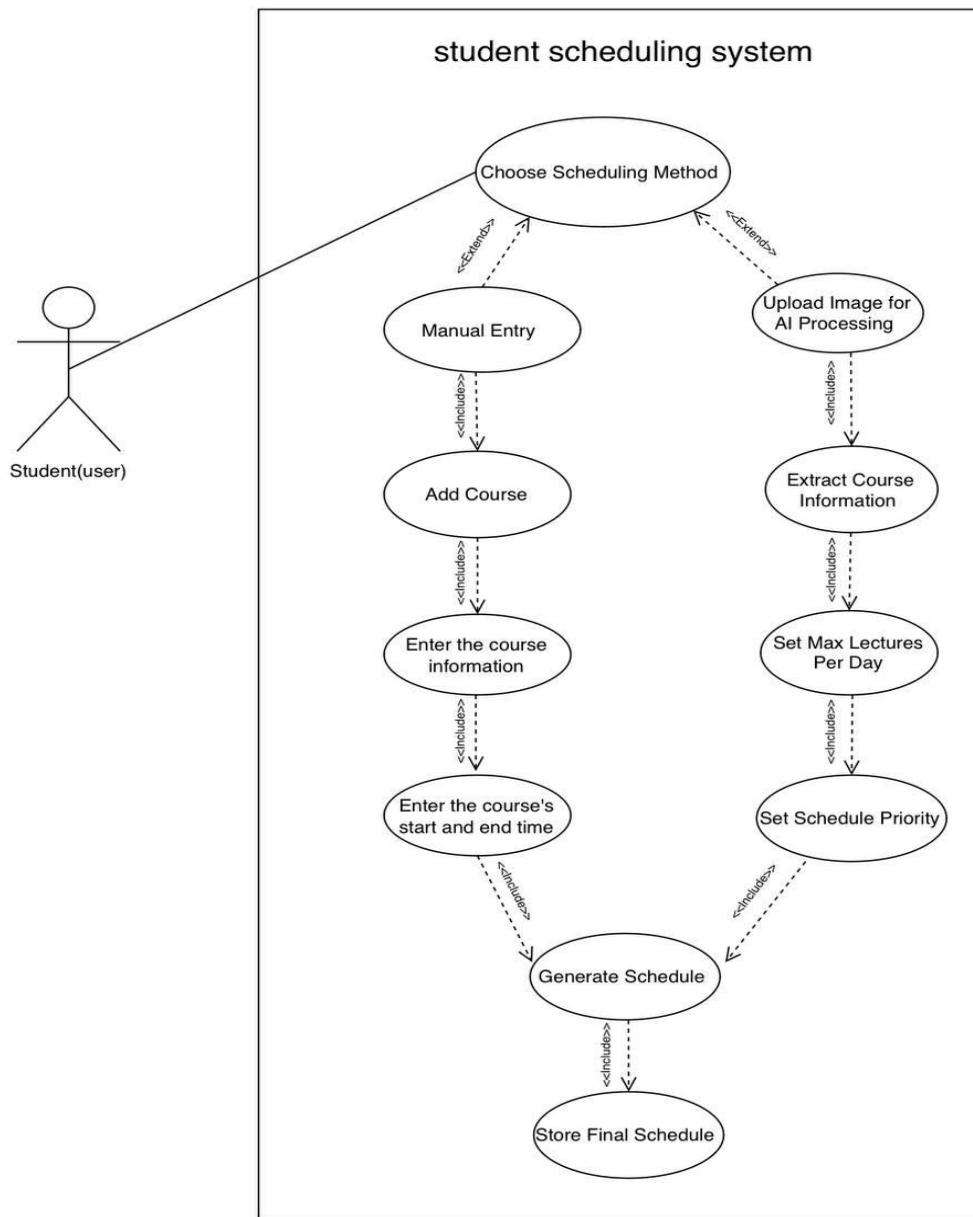


Figure 6 Use Case Academic schedule

The use case diagram for the Academic Schedule illustrates the Scheduling System, allowing students to either manually enter course data or upload an image for AI-based extraction. The system then applies preferences such as setting no-course days, setting schedule priority, and limiting the maximum number of lectures per day before generating and storing the final schedule.

Principal actor	Student
Description	The student interacts with the scheduling system to either manually enter course details or upload an image for AI processing. After input is provided, the system applies scheduling preferences such as setting no-course days, prioritizing courses, and limiting daily lectures before generating and storing the final schedule.
Data	<ul style="list-style-type: none"> - Course details (Name, Hours, Section) - Uploaded schedule image (for AI processing) - No-course day preferences - Schedule priority settings - Maximum lectures per day
Pre-condition	<ul style="list-style-type: none"> - The student must be logged into the system. - The system must have access to course data for validation (if applicable). - The AI processing feature must be functional for image uploads.
Output	<ul style="list-style-type: none"> - A generated schedule based on student preferences. - The final schedule is stored in the system. - The student can view, modify, or export the schedule. - Notifications if scheduling conflicts occur.
Comments	The scheduling system provides flexibility by allowing students to manually input data or leverage AI-based scheduling. This improves efficiency and personalization, ensuring that course schedules align with student preferences.

Table 7 Scenario schedule

Scenario Academic schedule The table describes the functionality of the academic schedule feature within the application, where the main actor is the student. The student interacts with the scheduling system by either manually entering course details or uploading an image for AI-based processing. The system applies preferences such as setting no-course days, prioritizing courses, and limiting daily lectures before generating and storing the final schedule. The main data includes course details, uploaded schedule images, and user-defined preferences.

3.8.5 Use Case GPA:

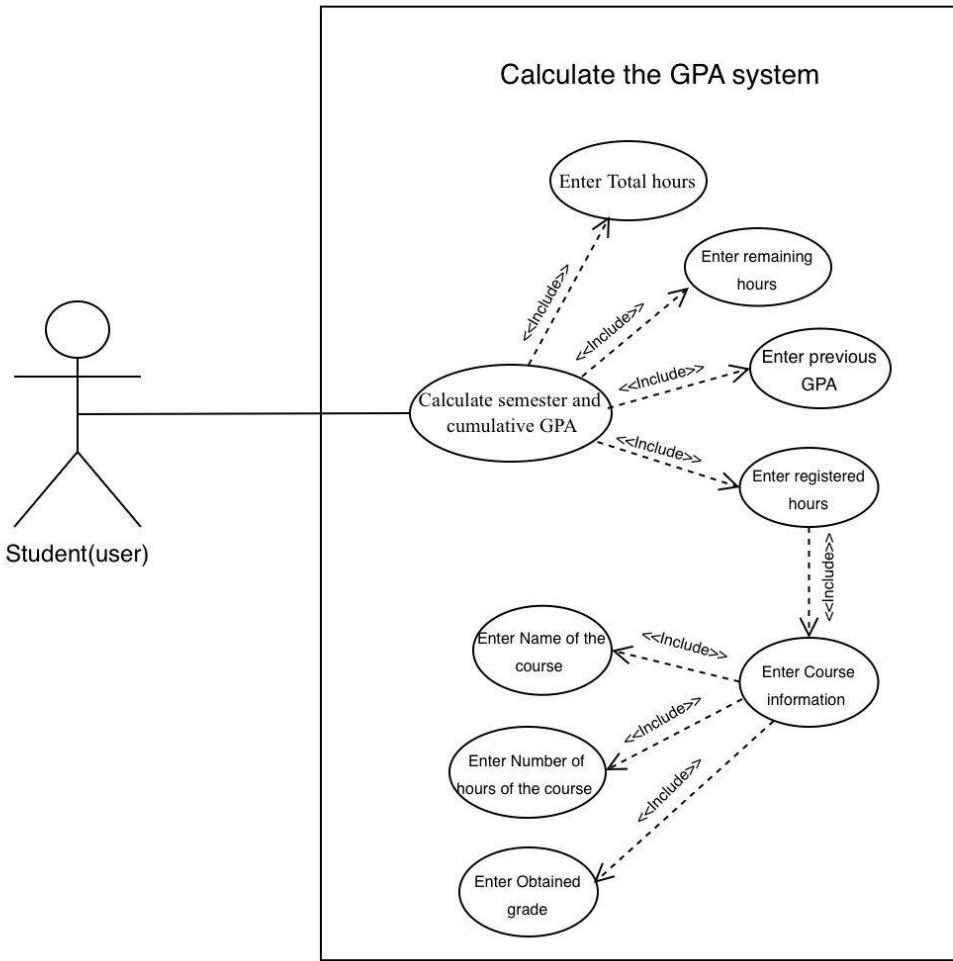


Figure 7 Use Case GPA

The use case diagram for the **GPA Calculator interface** illustrates the GPA Calculator interface, allowing students to manage their academic performance with ease. Students input data such as their previous GPA, recorded hours, remaining hours, and completed hours, along with course details like names, credit hours, and grades. The system processes this data to calculate the semester or cumulative GPA and displays the expected GPA instantly, helping students track their academic progress clearly and efficiently.

Principal actor	Student
Description	The student accesses an interface designed for GPA calculation, where they input essential academic details such as their previous GPA, recorded hours, remaining hours, and completed hours. The interface also allows the student to add specific course details, including course name, credit hours, and the obtained or expected grade. The system calculates the cumulative or semester GPA based on the entered data and displays the result in real time for easy tracking and monitoring of academic performance.
Data	<ul style="list-style-type: none"> - Completed credit hours - Remaining credit hours - Registered credit hours - Previous cumulative GPA - Course name - Course credit hours - Expected grade (e.g., A+, A)
Pre-condition	<ul style="list-style-type: none"> - The student is registered in the system and has access to the GPA Calculator interface. - Accurate data, including grades and credit hours, are provided by the user.
Output	<ul style="list-style-type: none"> - Display of the calculated cumulative or semester GPA. - Real-time updates of the GPA after adding or modifying course details.
Comments	<ul style="list-style-type: none"> - The interface is user-friendly, providing clear fields for input and automatic calculation of GPA. - Real-time feedback ensures the user stays informed of changes to their GPA. - The system includes guidance messages for data entry and alerts for invalid or incomplete inputs.

Table 8 GPA Scenario

GPA Calculation Page Scenario This table describes the functionality of the GPA calculation page within the app, allowing students to calculate and track their academic GPA. When the student accesses the page, the system provides fields to enter completed, remaining, and registered credit hours, along with the previous cumulative GPA. The student can choose to calculate either cumulative or semester GPA. After entering course details, the system calculates and displays the GPA clearly. If any data is missing, the system shows an error message to ensure accurate calculations.

3.8.6 Use Case Attendance Tracking:

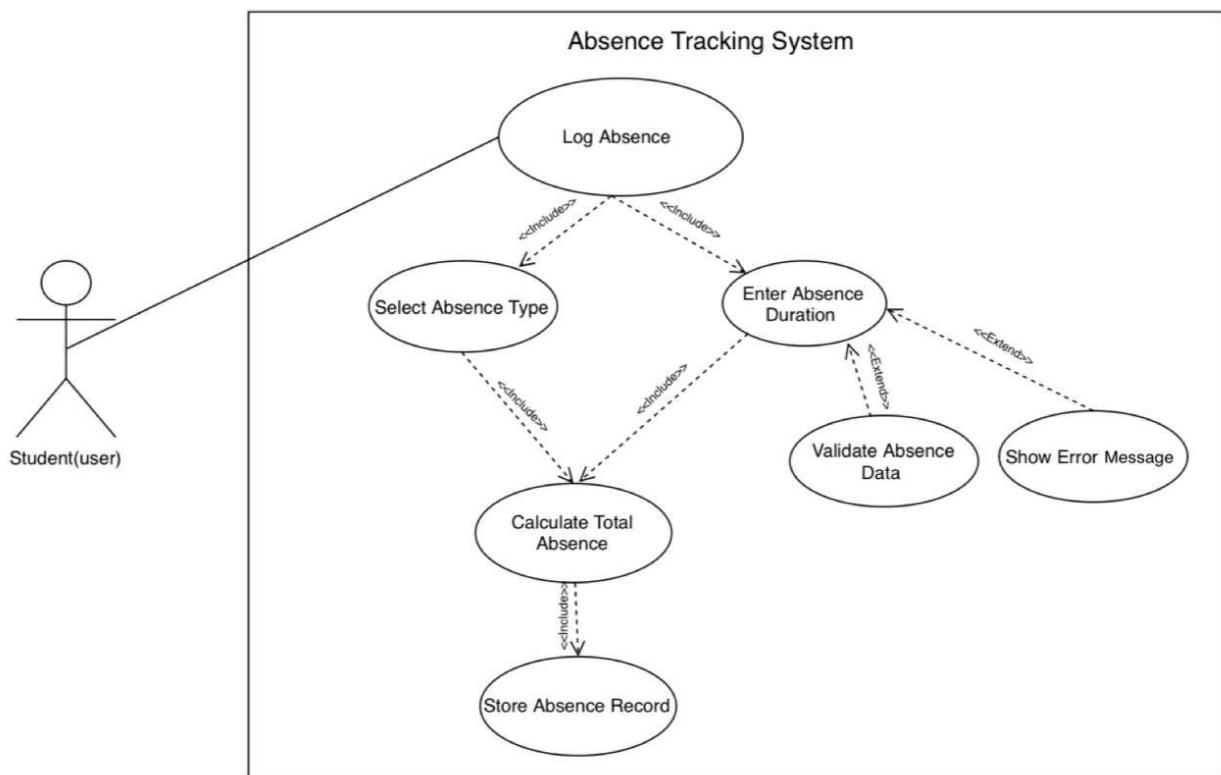


Figure 8 Use Case Attendance Tracking

The use case diagram for the Absence Tracking System illustrates the Absence Tracking System, enabling students to efficiently manage their absence records. Students can log a new absence by selecting the type of absence, entering the duration, and saving the record in the system. The system automatically calculates the total absence and validates the entered data. If any errors are detected, an error message is displayed to guide the student. Additionally, the system ensures accurate record-keeping by structuring tasks into necessary sub-tasks and optional actions, allowing flexibility and ease of use.

Principal actor	Student
Description	The student interacts with the system to add a new absence record. This includes selecting the type of absence (e.g., justified or unjustified), entering the absence duration, and saving the record in the system. The system automatically calculates the total absence hours based on the provided duration.
Data	-Absence type (e.g., justified, unjustified) -Duration (hours or days)
Pre-condition	The student must be registered in the system.
Output	The updated total absence hours displayed to the user.
Comments	This use case relies on key functionalities, including selecting the absence type, calculating total hours, and saving the record. If the student attempts to enter invalid data, the system displays an error message and requests correction. This ensures accurate absence tracking while providing a user-friendly experience.

Table 9 Scenario Attendance Tracking

Scenario Attendance Tracking This table describes the functionality the student interacts with the system to add a new absence record. The process includes selecting the type of absence (justified or unjustified), entering the absence duration (in hours or days), and saving the record. The system automatically calculates the updated total absence hours based on the input provided. The student must be registered in the system to access this functionality. Once the process is complete, the updated total absence hours are displayed to the user, providing a seamless and efficient absence tracking experience.

3.8.7 Use Case contact Ambassador

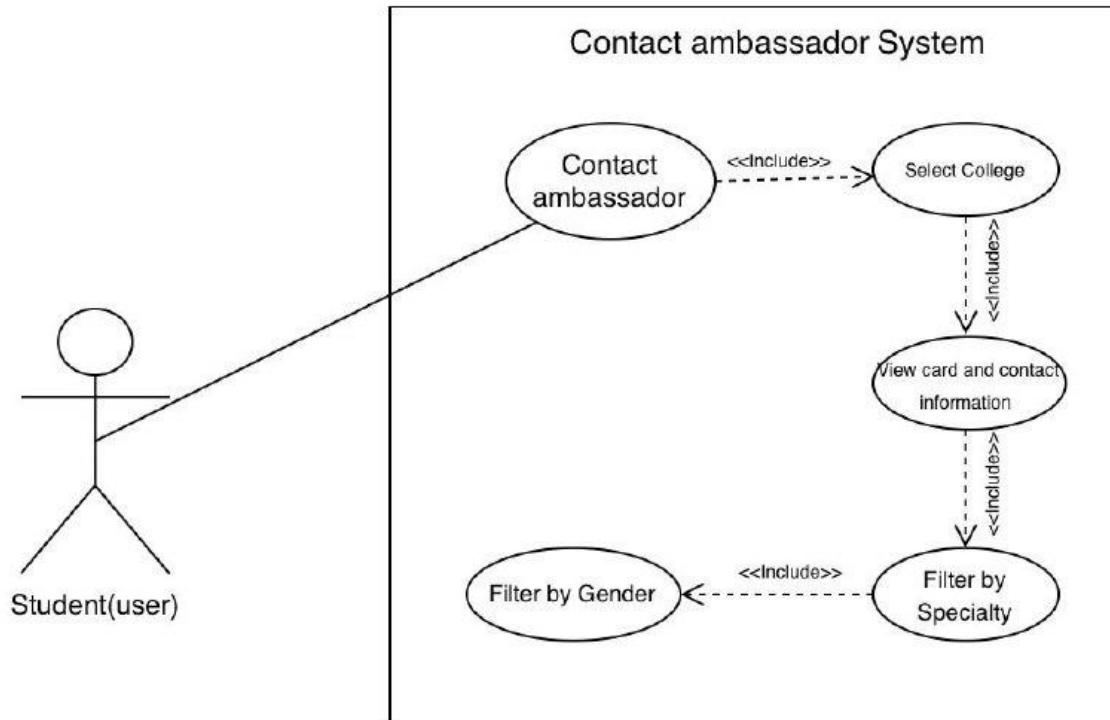


Figure 9 Use Case Contact Ambassador

The use case diagram for Contact Ambassador In this diagram, there is one key actor: the user (student). This actor is associated with two primary use cases:

1. Search for Students by Major: This use case allows the user to search for other students based on their major by accessing the search feature within the application.
2. View Card and Contact Information: This use case is part of the “Search for Students by Major” use case. After finding students based on their major, the user can view detailed cards and contact information for the selected students. The system’s response to these actions includes processing the search request, retrieving the relevant student information, and displaying the detailed contact information as needed. This diagram effectively maps out the user’s interaction with the system to achieve their goal of finding and contacting other students based on their major.

Principal actor	Safra (student helping other students)
Description	The system allows students to search for contact information of other students by their academic major.
Data	<ul style="list-style-type: none"> - Student name - Major - Contact information
Pre-condition	<ul style="list-style-type: none"> - Ambassadors (platform administrators) have added their information. - Ambassadors do not need login access.
Output	Displays the selected student's basic information and contact details.
Comments	The search process must be efficient, fast, and ensure the privacy of contact information.

Table 10 Scenario Contact Ambassador

Scenario Contact Ambassador This table describes the functionality "Student Ambassador Communication" allows students to search for ambassadors based on their academic major. Ambassadors provide their information without the need for login access, which is managed by platform administrators. The feature displays the ambassador's name, major, and available contact links (such as Twitter and LinkedIn), ensuring privacy. When the student accesses this page, they can choose the desired academic major from the available selection list. After selecting the major, a list of ambassadors associated with that major is displayed, along with their personal details and available contact links. If any data is missing, the system will display an alert.

3.9 User Classes and Characteristics

The primary user class for the Muad app consists of students, who use the app daily to manage their study schedules, track absences, and calculate GPA. Students require an intermediate level of technical expertise to navigate the app effectively and utilize its features to optimize their academic performance. As the main focus of the app, students represent the most critical user class.

3.9.1 Leisure Students

students who seek to simplify their academic lives using the app. They are interested in managing their time and schedules in a convenient and seamless way, without worrying about daily tasks such as calculating GPA or tracking attendance or communicate with ambassadors. Their goals include easily calculating their academic GPA without complexity, using attendance tracking tools and alerts to avoid any academic issues. They also aim to create customized study schedules based on personal preferences such as break times and holidays, and to improve their academic experience through organized and easy-to-use tools.

3.9.2 Family Students

A well-organized and flexible academic experience through the app. These students aim to manage their study schedules and academic activities in a way that fits their daily needs. They also prefer to take advantage of communication options with academic ambassadors and find educational materials related to their academic journey.

3.9.3 Cultural Students

The app also provides options to connect with cultural ambassadors who offer advice and guidance on how to integrate cultural activities with the academic path.

3.10 Operating Environment

3.10.1 Hardware

The Muad application is designed to run on commonly available hardware, including:

- Mobile Devices: Smartphones with at least 2GB RAM and a dual-core processor or higher.
- Storage: Minimum of 100MB free space for application installation and temporary data storage.
- Servers: Cloud-based or on-premises servers with sufficient computational power to support AI features, real-time calculations, and database operations.

3.10.2 Operating System

The application supports the following operating systems:

- Mobile:

Android version 8.0 (Oreo) and above.

- Server-Side:

Linux-based servers (e.g., Ubuntu, CentOS) for backend operations.

Compatibility with modern web server environments like Apache or Ngin

3.10.3 Internet Connectivity

- Online Functionality: Requires a stable internet connection for features like schedule synchronization, real-time, GPA calculations, and communication with ambassadors via external platforms.
- Minimum required speed: 2 Mbps for optimal performance.
- Offline Support: Certain features, such as viewing saved schedules and GPA history, are accessible offline but will sync automatically when reconnected.

3.11 Design and Implementation Constraints

3.11.1 Scheduling System

The Muad application must provide a flexible and efficient scheduling system that allows users to create customized study schedules based on their preferences. The application should include an easy-to-use interface for adding courses, specifying lecture times, and organizing academic activities.

3.11.2 Cross-Platform Compatibility

The Muad application should be designed to work seamlessly across multiple platforms, including Android, to ensure accessibility for a wide range of users and provide a consistent user experience across all devices.

3.11.3 Performance Optimization

- The Muad application must deliver high-speed performance, ensuring that all features, including GPA calculations, schedule generation, communicate with ambassadors, and absence tracking, are processed in real time without noticeable delays.
- Efficient backend operations are required to minimize server response time during peak usage.

3.11.4 Academic Context and Accuracy

- The Muad app ensures alignment with the academic context and needs of its users. This includes features such as scheduling adjustments based on the university's academic calendar and supporting event filters relevant to student activities, ensuring the app remains accurate and tailored to users' requirements.

3.11.5 User Accessibility

- - The Muad app ensures accessibility by providing a user-friendly design that supports all students. The app adheres to accessibility standards, such as WCAG 2.1, by incorporating features such as clear navigation, keyboard compatibility, and readable text sizes.

3.11.6 Data Security and Privacy

- The application must comply with privacy standards (e.g., GDPR) to protect users' personal and academic data.
- Secure authentication, data encryption, and limited data sharing ensure user privacy and safety.

3.11.7 Scalability

- The system must be scalable to accommodate an increasing number of users, features, and data load without affecting performance.
- Cloud-based infrastructure should support dynamic scaling for high user demands.

3.11.8 Device Limitations

- The application must be optimized to work efficiently on devices with limited processing power, storage, or older operating systems, ensuring inclusivity for a broader range of users.
- Lightweight design and efficient resource usage will help maintain performance on such devices

3.12 User Documentation

3.12.1 User Manual

A comprehensive user manual will be provided, explaining all features of the Muad application, including how to create study schedules, calculate GPA, track absences, and communicate with academic ambassadors. The manual will include step-by-step instructions with screenshots to guide users through each feature.

3.12.2 User Support

The Muad app provides a dedicated support feature through academic ambassadors. Students can view ambassador profiles, which include a brief introduction, their area of expertise, and links to their social media accounts, such as LinkedIn and Twitter. This allows students to contact ambassadors directly for guidance on specific majors, study plans, or academic concerns, ensuring personalized and efficient support.

3.12.3 User Interface Design

The user interface will be designed to be simple and user-friendly to ensure a seamless experience. The focus will be on clear organization of buttons and options, with colors and layouts that enhance the app's clarity and ease of navigation.

3.12.4 Feedback System

The application will include a feedback system allowing users to provide suggestions and feedback on features and functionalities. Users will be able to submit ratings, report bugs, and offer suggestions for improvement. This system will enable continuous development of the application based on user needs

3.13 Assumption

- 1. User Access:** It is assumed that users have basic knowledge of using smartphones and apps, which will facilitate the adoption of the Muad application.
- 2. Device Compatibility:** The app assumes that users have devices running at least the minimum required version of the Android operating system for optimal performance.
- 3. Internet Connectivity:** It is assumed that the majority of users will have access to the internet for functionalities that require real-time data, such as course registration and communication with academic ambassadors.
- 4. Data Accuracy:** The application assumes that the data entered by users, such as course details and grades, will be accurate and up-to-date.

3.14 Dependencies

- 1. Platform Dependency:** The Muad application depends on the Android operating system to run smoothly, requiring compatibility with multiple versions within the supported range.
- 2. Third-Party APIs:** The application relies on third-party APIs for certain features like real-time notifications, cloud storage for data, and communication tools (e.g., messaging for academic ambassadors).
- 3. Internet Connectivity:** The full functionality of Muad depends on an active internet connection for features such as schedule synchronization, notifications, and accessing academic resources.
- 4. Data Providers:** The application depends on accurate data from the university or relevant academic bodies for course schedules, grades, and attendance information.
- 5. Cloud Storage:** For storing user data and academic progress, Muad depends on reliable cloud storage solutions to ensure data is safe and accessible at all times

3.15 External Interface Requirements

3.15.1 User interfaces (Initial Interface)



Figure 10 Login Method Selection Screen

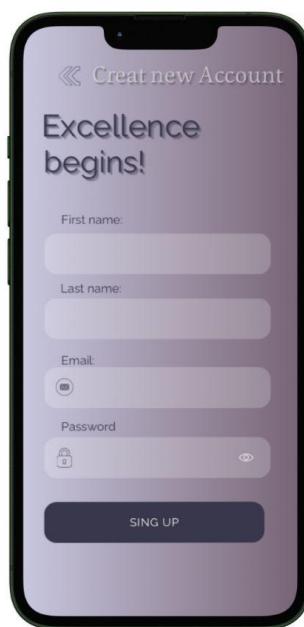


Figure 12 Sign up screen



Figure 11 log in screen

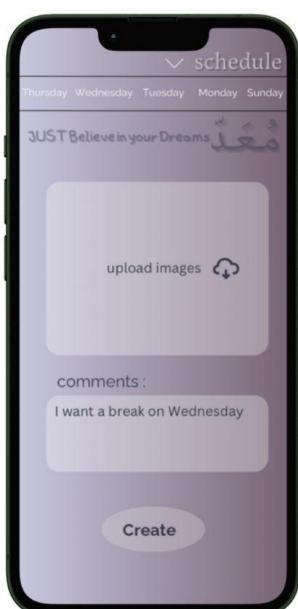


Figure 14 schedule AI Planner

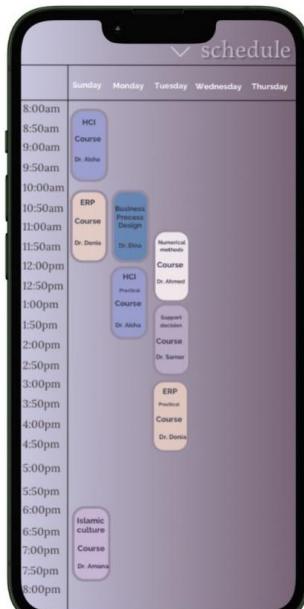


Figure 13 view schedule



Figure 15 GPA calculator



Figure 18 GPA calculator 2



Figure 17 Attendance and absence screen

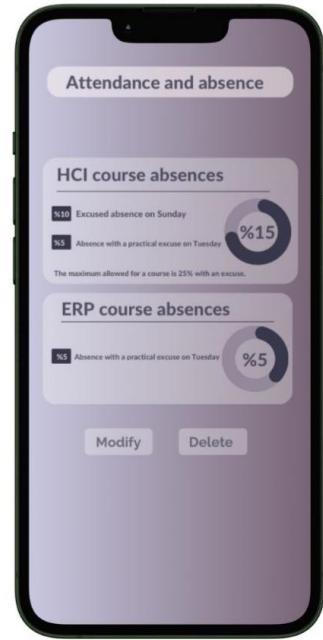


Figure 16 Absence screen screen 2

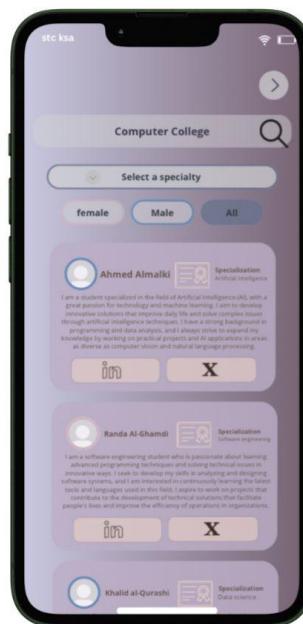


Figure 19 ambassador Screen

Initial Interface Design Observations

1. Colors Not Aligned with Umm Al-Qura University's Identity

- The initial design lacked alignment with the visual identity of Umm Al-Qura University.
- The university typically uses calm and formal colors that reflect its academic and official nature. However, the initial design employed bright or random colors that did not reflect this identity.

2. Inconsistent Fonts and Mixed Languages

- The initial design used varied fonts across the interfaces, which weakened the visual consistency and negatively impacted the user experience.
- Mixing Arabic and English in the same interface can cause confusion and should be avoided. Instead, ensure that the interface supports language selection with consistent translation.

3.15.2 User Interfaces (Secondary Interface)



Figure 20 Login Method Selection Screen

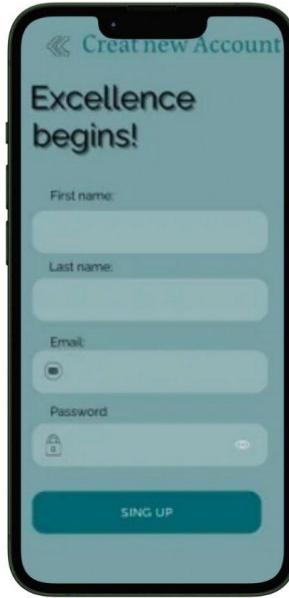


Figure 22 Sign up screen

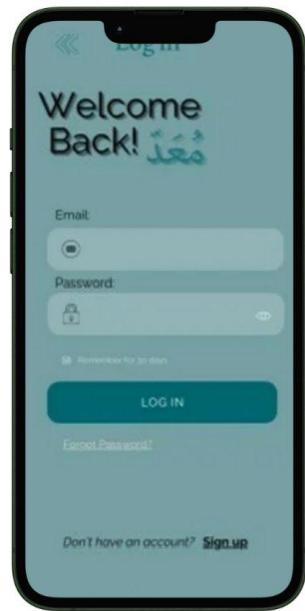


Figure 21 log in screen



Figure 24 schedule AI Planner

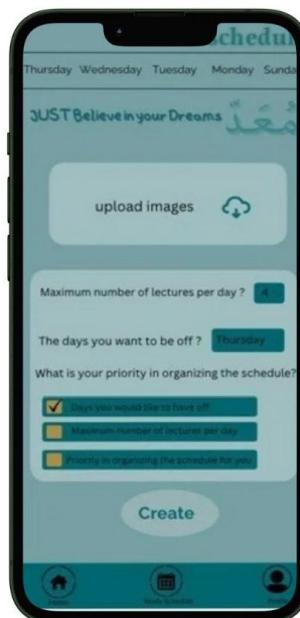


Figure 23 view schedule

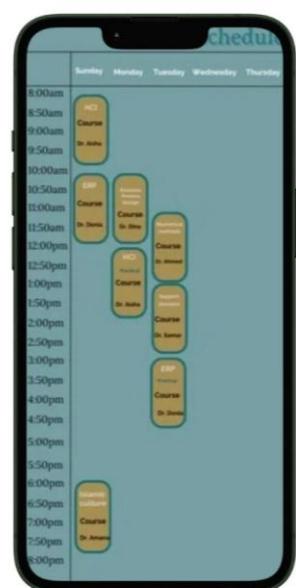


Figure 25 GPA calculator



Figure 28 GPA calculator 2

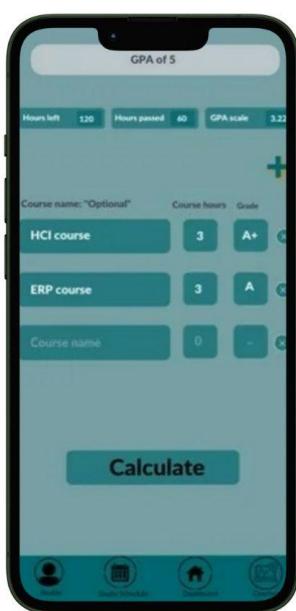


Figure 27 Absence screen screen

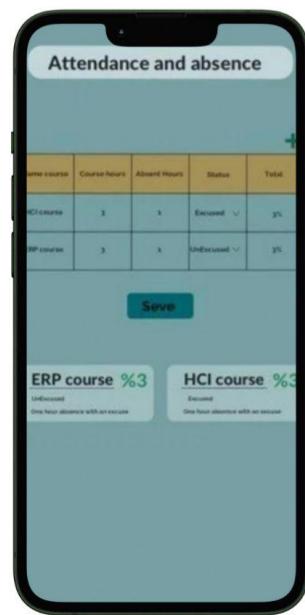


Figure 26 Absence screen screen 2



Figure 29 ambassador Screen

Secondary Interface Design

1. Design Misalignment with Umm Al-Qura University's Identity

- Colors : Colors inspired by Umm Al-Qura University's identity were used but did not achieve full alignment.
- White Background : A white background is a key element in the university's visual identity, making it the appropriate choice.
- Header, Icons, and Bottom Navigation : These elements were not fully aligned with the university's visual theme.

2. Missing Key Interfaces

- The "Forgot Password" interface was not included, which is an essential feature for any academic application.
- This interface should include two main options: password recovery via registered email or phone number.
- Its design requires a simple layout with clear instructions to guide users through the password recovery process

3.16 Hardware Interfaces

3.16.1 Display Screen

The device must have a display screen to present the application's user interface, study schedules, and academic progress. The screen should support standard resolutions for clear visibility and usability.

3.16.2 Input Mechanisms

The device should have input mechanisms such as a touchscreen or keyboard, enabling users to interact with the application effectively. Features like grade entry, schedule editing, and navigation require precise input.

3.16.3 Internet Connectivity

The device must have internet access to facilitate communication with the application's backend servers. This ensures real-time data synchronization for schedule updates, grade tracking.

3.16.4 Photo Upload Capability

The application allows users to upload existing images from their device's storage, such as profile pictures or scanned documents (e.g., course schedules). The device does not require a camera for capturing photos but should have access to the device's storage system to enable file selection and upload.

3.16.5 GPA

- Utilizes hardware processing power on both the user's device and backend servers to perform:
 - Real-time GPA calculations.
 - Data validation for accurate and reliable academic performance tracking.

3.17 Software Interfaces

In the future, the Muad app will include enhancements such as the integration of additional university campuses and academic facilities, allowing students to explore them virtually. The app will also support additional languages to cater to a wider student audience. Furthermore, new features will be continuously added to improve the user experience and academic support, ensuring the app remains relevant and effective for its users.

3.18 System Features

3.18.1 Core features

The system's core features include interfaces for GPA calculation and attendance tracking, where users can input their credit hours and GPA. It also includes an attendance interface displaying a table with course names. The "Ambassadors" interface allows students to select their specialization and communicate with ambassadors via social media. Additionally, a schedule creation interface uses AI to analyze an uploaded image of available courses and generates a schedule based on the student's preferences. Finally, the homepage provides quick access to these features.

3.18.2 User interactions features

The user interaction features focus on providing intuitive navigation with a clean, simple interface and clear labels for each feature. Users can interact by inputting data for GPA calculations, attendance tracking, and schedule creation through forms and tables. The system also allows customization, enabling users to personalize their experience, such as selecting days off or adjusting GPA calculation preferences.

3.18.3 Technical features

The technical features include AI-powered image recognition for extracting course and time information from uploaded images when creating a schedule. User inputs, such as GPA history, attendance data, and preferences, should be securely stored for future use. Additionally, a robust backend algorithm will process GPA and attendance calculations efficiently and accurately.

3.19 Conclusion

This chapter provided a comprehensive overview of the functional and non-functional requirements for the “Muad” application. It detailed key features such as ambassador profiles, GPA calculations, academic scheduling, and attendance tracking, all designed to enhance the user experience. The performance, security, and scalability requirements ensure that the app will perform efficiently and securely, even with a growing user base. The chapter also discussed the importance of accessibility, data privacy, and compliance with educational regulations, emphasizing the app’s commitment to user safety and data protection. By addressing these core elements, this chapter establishes a clear direction for the continued development and refinement of the “Muad” application, ensuring that it aligns with both user needs and technical specifications, ultimately fostering an engaging and reliable platform for students.

4 CHAPTER 4 SOFTWARE DESIGN

4.1 Introduction

Chapter 4 focuses on the system design and architecture of the Muad app, providing a detailed analysis of its components and their interactions. This chapter presents several diagrams and models that illustrate how the system is organized, how its components interact, and how it achieves its intended functions.

The use case diagram defines the key interactions between users and the system, highlighting essential functions such as user registration, login, academic progress tracking, and schedule management. The context diagram provides an overview of how the system interacts with external entities, including databases, users, and external services, ensuring smooth data flow and interaction.

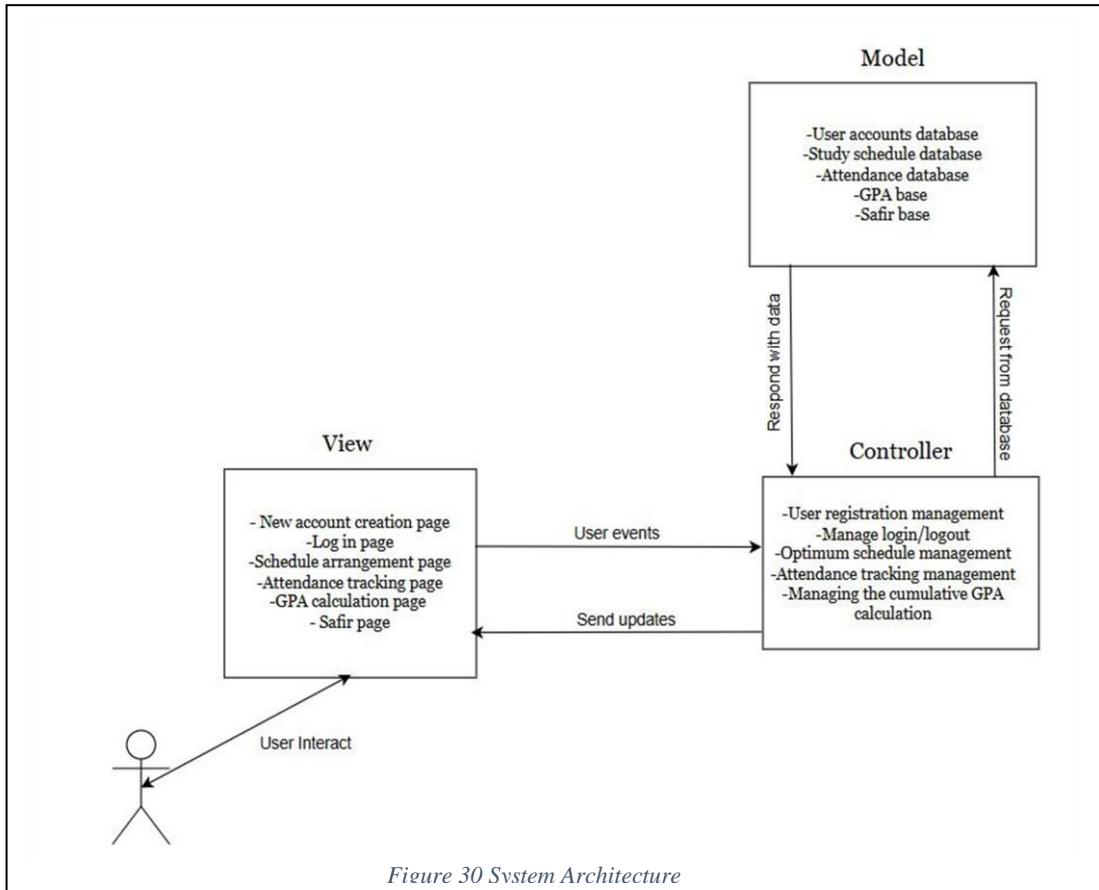
Sequence diagrams visualize the order of interactions between system components during major processes, such as user login, registration, and schedule optimization. Similarly, activity diagrams describe the workflow of important system functions, such as generating personalized study schedules and tracking academic progress.

The system architecture describes the overall structure of the application, including backend services, frontend interfaces, and the communication between them. The structural design outlines the components that make up the system and their relationships, ensuring modular and maintainable development.

This chapter also details the user interfaces that enable interaction between users and the system, ensuring usability and intuitive navigation. Additionally, it describes the data storage mechanisms where user information, course details, and academic data are securely stored and managed.

Together, these models and diagrams provide a clear representation of the system's structure and behaviour, serving as a blueprint for implementation and ensuring that the application meets both functional and non-functional requirements outlined in previous chapters.

4.2 System Architecture



This diagram System Architecture This system architecture diagram illustrates a design based on the Model-View-Controller (MVC) pattern. In this structure, the View component represents the user interface, which includes pages for creating a new account, logging in, scheduling, tracking attendance, calculating GPA, and accessing the Safir page. Users interact with these pages, and their actions are transmitted as events to the Controller. The Controller acts as a bridge between the View and the Model, managing processes such as user registration, login/logout, optimal schedule arrangement, attendance tracking, and GPA calculation. It handles user requests, processes them, and communicates with the Model to retrieve or update data as needed. The Model layer contains databases for user accounts, study schedules, attendance records, GPA calculations, and Safir data. It responds to the Controller's requests by providing the necessary data, which the Controller then relays back to the View. This architecture facilitates a structured flow where user actions in the interface trigger data processing and retrieval, creating an efficient and organized system for managing educational tasks

4.3 Constraints and Assumptions

Constraints:

- 1. Data Privacy Constraints:** The system must comply with data protection regulations (e.g., GDPR) to ensure that students' academic records and personal information are secure and confidential.
- 2. Integration with Educational Institutions:** The application must be compatible with existing databases and learning management systems used by educational institutions, which may have varying data structures and APIs.
- 3. User Accessibility:** The application must be designed to work on various devices and operating systems (Android) to ensure user accessibility.
- 4. Performance Limitations:** The system should efficiently handle multiple users accessing their academic records simultaneously without significant lag or downtime.

Assumptions:

- 1. User Engagement:** Students are assumed to actively use the application to track their academic performance, provide feedback, and input data as required.
- 2. Availability of Data:** In the future, the Muad app will integrate with university systems to automatically retrieve academic data, such as course schedules, attendance, and GPA. Currently, students manually enter their academic data into the app, ensuring that it remains accurate and up-to-date for features like schedule management, attendance tracking, and GPA calculation.
- 3. User Familiarity with Technology:** It is assumed that users (students, teachers, and administrators) are familiar with technology and comfortable using mobile and web applications.
- 4. Quality of Data:** The system assumes that the academic data provided by institutions will be of high quality, accurate, and reflective of the student's actual performance.

4.4 Interfaces and Data Stores

4.4.1 Dataflow Diagram

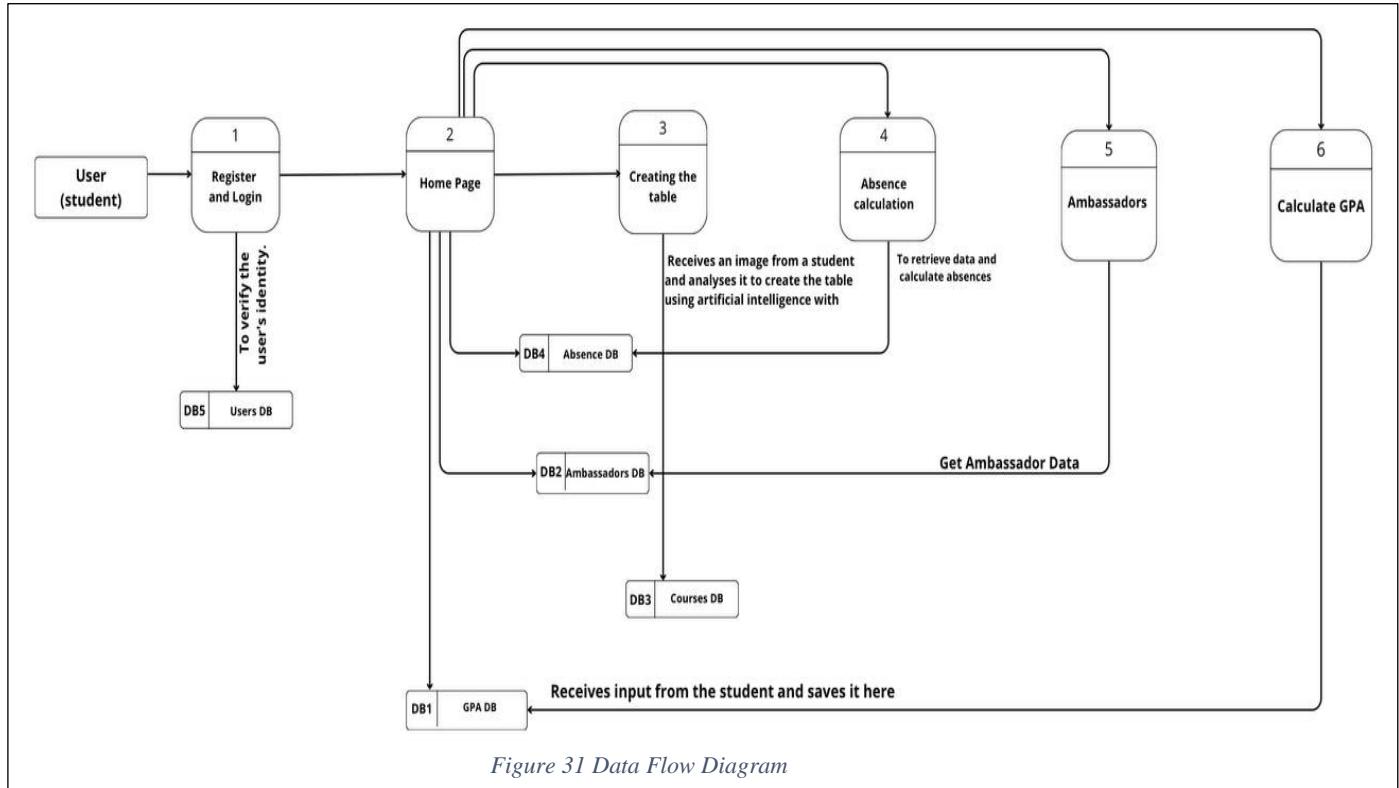


Figure 31 Data Flow Diagram

This diagram Data Flow Diagram Descriptions

- Users DB (DB5): Verifies user identity during registration and login.
- GPA DB (DB1): Provides data for GPA calculation.
- Absences DB (DB2): Stores and retrieves attendance records for absence calculation.
- Courses DB (DB3): Contains course information for creating the schedule table.
- Ambassadors DB (DB4): Stores information about ambassadors for user reference.

Key Data: - The system interacts with five main databases: Users, Courses, Absences, Ambassadors, and GPA. - The home page centralizes access to all features, providing a streamlined user experience with efficient data retrieval and processing.

4.4.2 Class Diagram

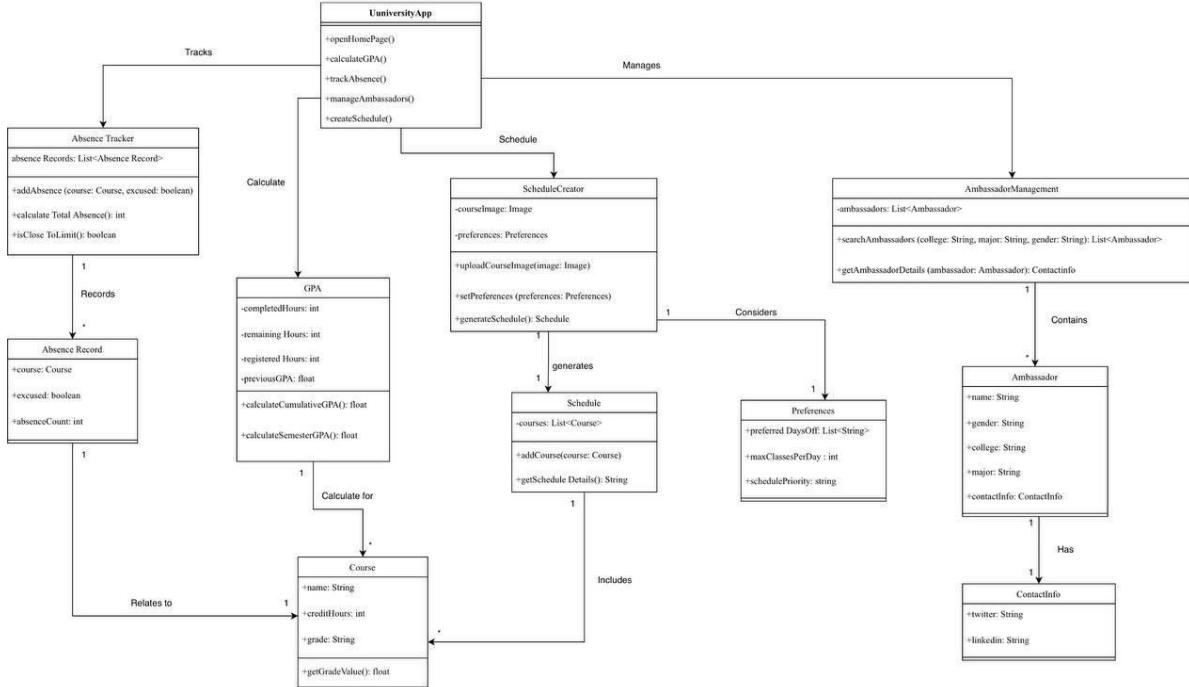


Figure 32 Class Diagram

This diagram class **Diagram** represents a university student management app. The central University App manages functionalities like GPA calculation (GPA), absence tracking (Absence Tracker), ambassador management (Ambassador Management), schedule creation (Schedule Creator). GPA calculates based on multiple Course objects. Absence Tracker records absences in Absence Record, which is linked to Course. Ambassador Management manages Ambassador objects, each with Contact Info. Schedule Creator generates a Schedule based on student Preferences and includes multiple Course objects. The diagram shows how these components interact to provide an integrated user experience.

4.5 Sequence diagram

4.5.1 Sequences HomePage

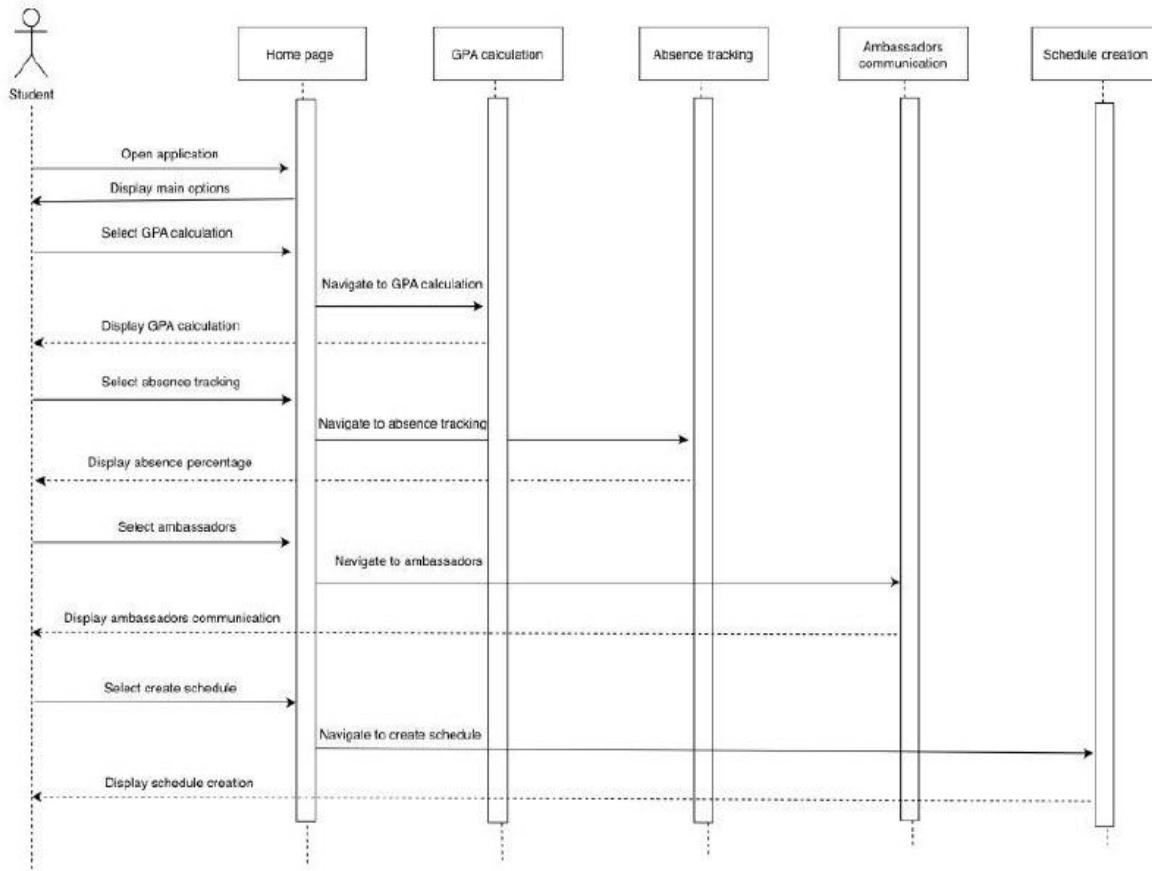


Figure 33 Sequences HomePage Diagram

This diagram **Sequences HomePage** illustrates the Homepage Navigation process within the application. When the student opens the app, the homepage is displayed, presenting several key options: GPA Calculation, Attendance Tracking, Communication with Ambassadors, and Schedule Creation. The student can select any of these options to navigate to the corresponding interface. Upon selection, the system directs the student to the relevant module, where they can view their GPA results, attendance percentage, ambassador communication interface, or study schedule creation tool. This structured navigation ensures a seamless user experience, allowing students to efficiently access and manage their academic and communication needs.

4.5.2 Sequence schedule

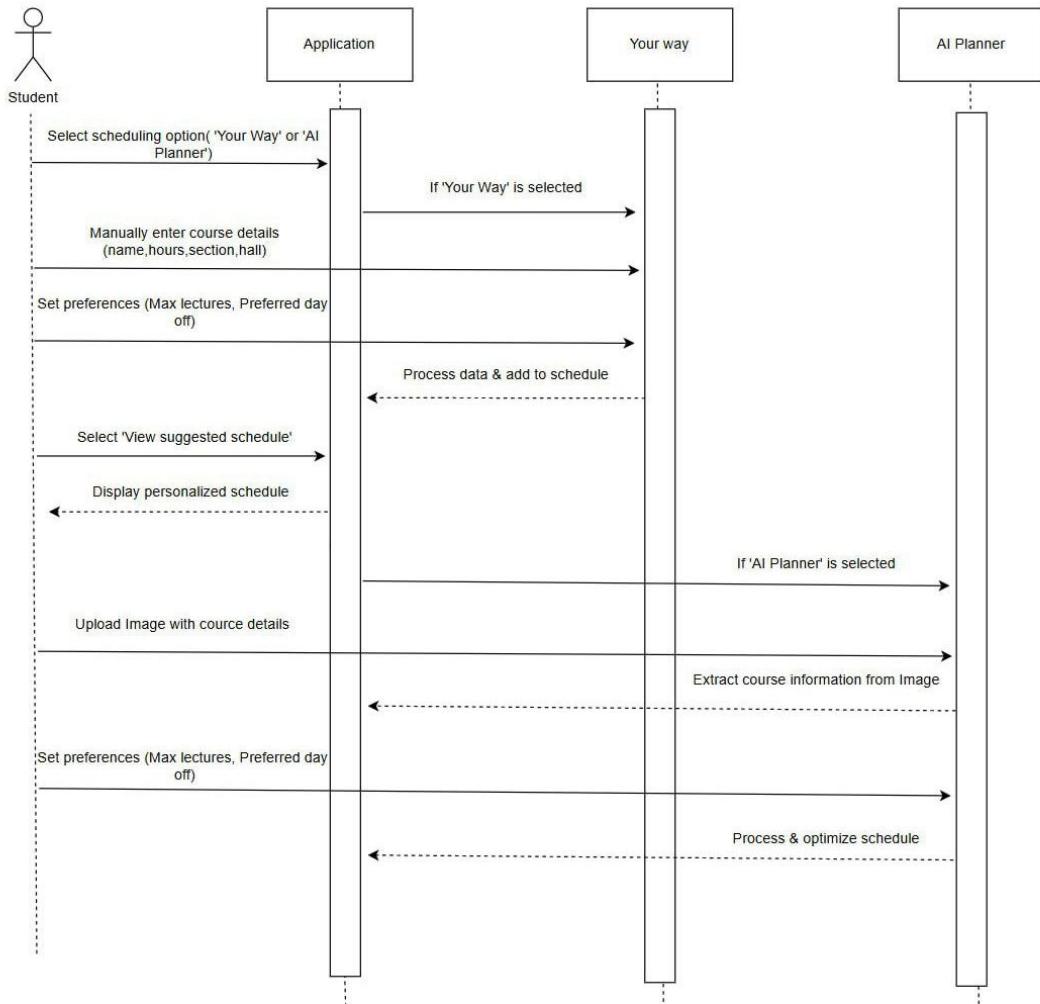


Figure 34 Schedule sequence Diagram

This diagram Sequences Schedule illustrates how a student interacts with a scheduling application using two options: “Your Way” or “AI Planner”. The process begins with the student selecting a scheduling option. If “Your Way” is chosen, the student manually enters course details (such as name, hours, section, and hall) and sets preferences (maximum lectures, preferred day off). The application processes the data and adds it to the schedule. The student can then view the suggested schedule, which is displayed accordingly. If “AI Planner” is selected, the student uploads an image containing course details. The AI extracts the necessary course information from the image, processes the data, and optimizes the schedule based on the student’s preferences. Finally, the personalized schedule is displayed for the student to review.

4.5.3 Sequences tracking absence

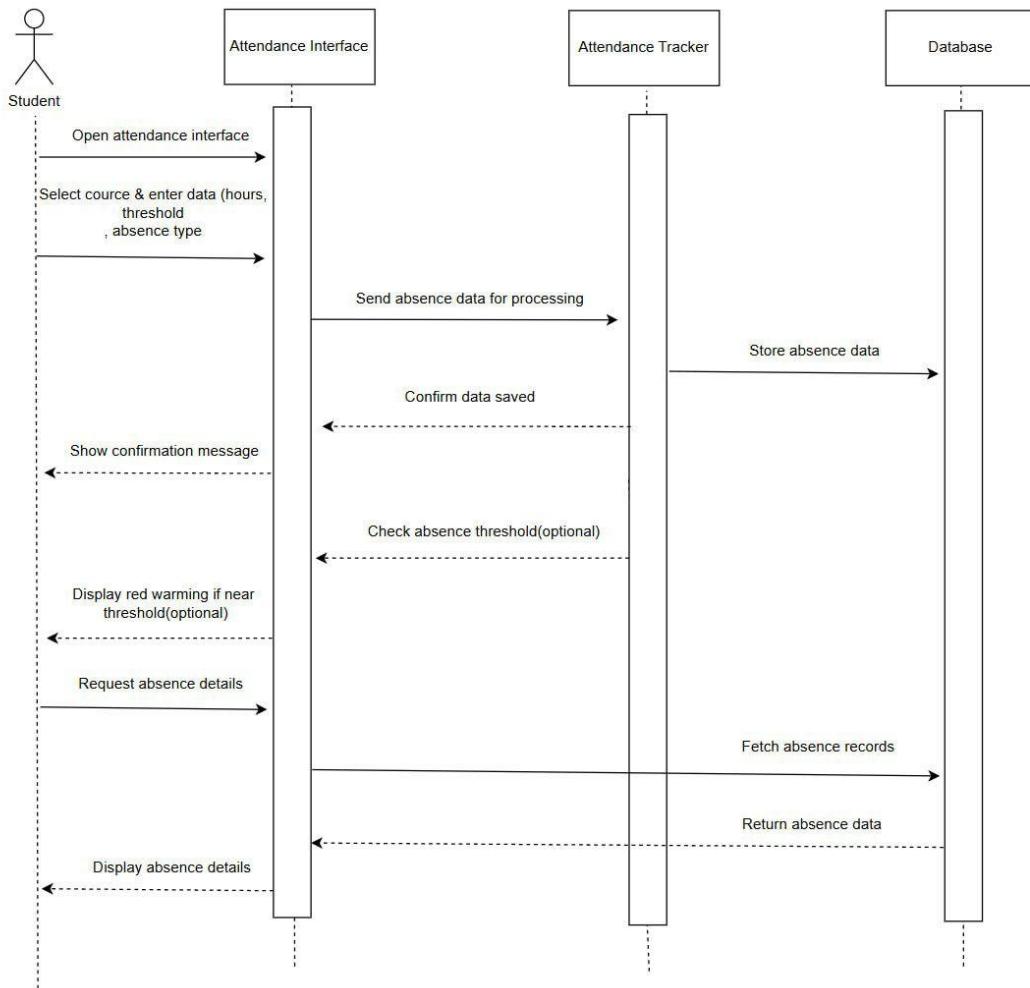


Figure 35 Sequences tracking absence

This diagram Sequences tracking absence This diagram illustrates how a student interacts with an attendance tracking system. The process begins with the student opening the Attendance Interface, where they select a course and enter details such as absence hours, threshold, and absence type. This data is then sent to the Attendance Tracker for processing. The Attendance Tracker forwards the absence data to the Database, where it is stored. Once the data is saved, a confirmation message is sent back to the student. If the system detects that the student is close to the absence threshold, an optional red warning is displayed. Additionally, the student can request their absence details, prompting the Attendance Tracker to fetch the absence records from the Database and return the relevant information to the student. Finally, the Attendance Interface displays the retrieved absence details for the student to review

4.5.4 Sequences Ambassadors

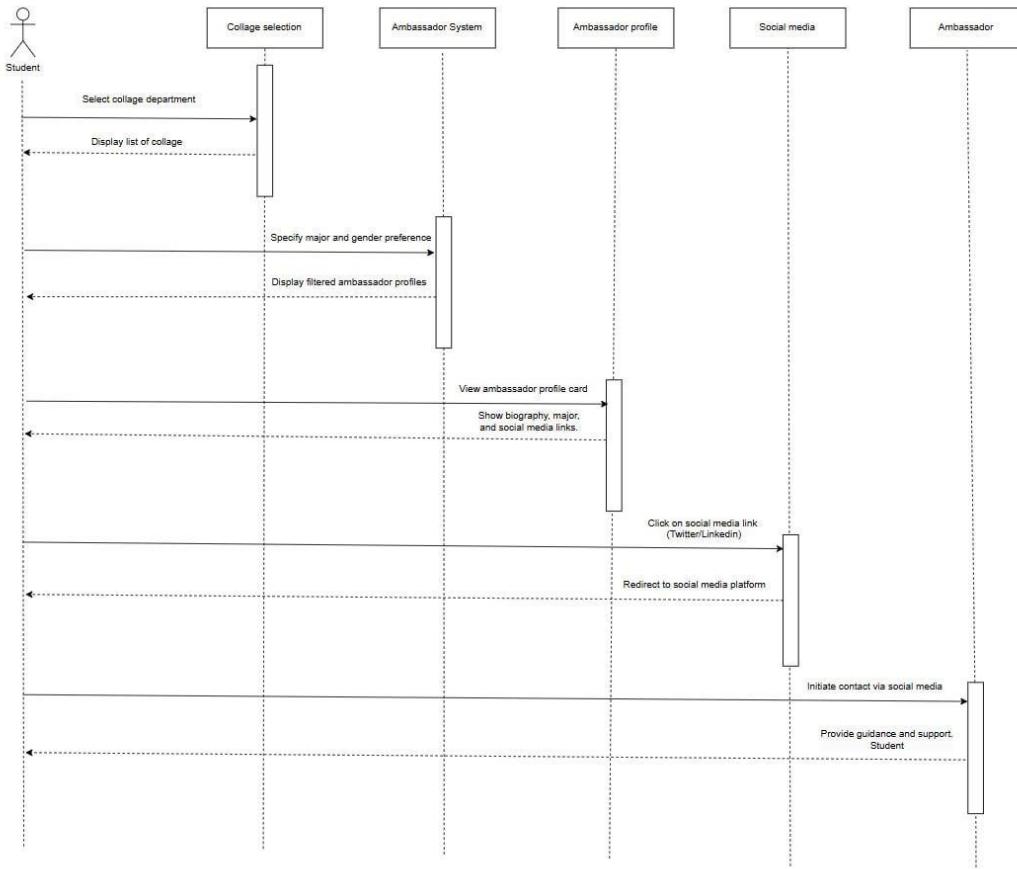


Figure 36 Sequences Ambassadors

This diagram Sequences Ambassadors illustrates the Ambassador Selection and Interaction process within the system. The student begins by inputting preferences such as major and gender, allowing the system to filter and display relevant ambassador profile cards. Each profile includes details like the ambassador's major and social media links (e.g., Twitter or LinkedIn). If the student chooses to interact further, they can click on a social media link, which redirects them to the respective platform for direct communication. This streamlined process ensures that students can easily find and connect with ambassadors based on their preferences.

\

4.6 ER diagram and relation schema

4.6.1 Activity Diagram schedule

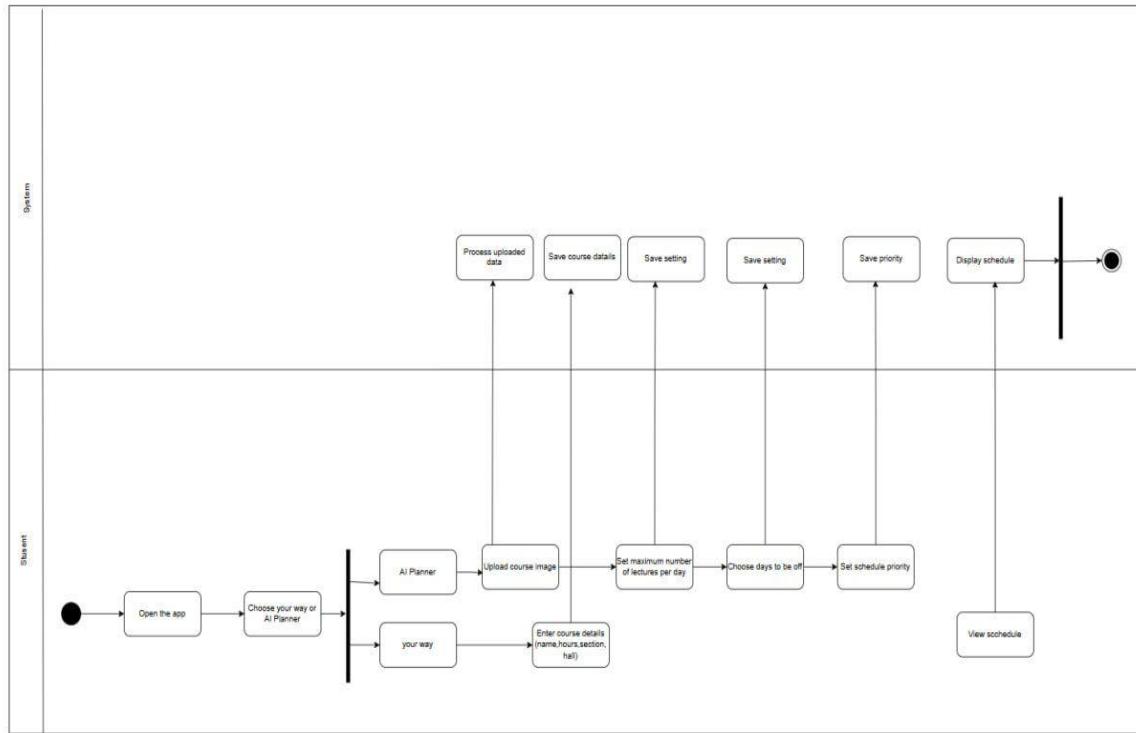


Figure 37 Activity Diagram schedule

The activity diagram for creating a study schedule illustrates how the student interacts with the system to create a study schedule using an application with two options: “Your Way” and “AI Planner.” The student begins by opening the app and selecting the preferred method. If the student chooses “Your Way,” they manually enter course details such as course name, number of hours, section name, and room number, with the option to view the schedule after adding the course. On the other hand, if the student selects “AI Planner,” they upload an image containing the desired courses, then answer customization questions such as the number of daily lectures (1-6), the preferred day off (from Sunday to Thursday), and schedule priorities (either reducing the number of study days or the number of daily lectures). The system processes the entered data, saves the settings, and displays the final schedule to the student. The process ends with the schedule being displayed, outlining the roles of both the student and the system in the process.

4.6.2 Activity Diagram Homepage

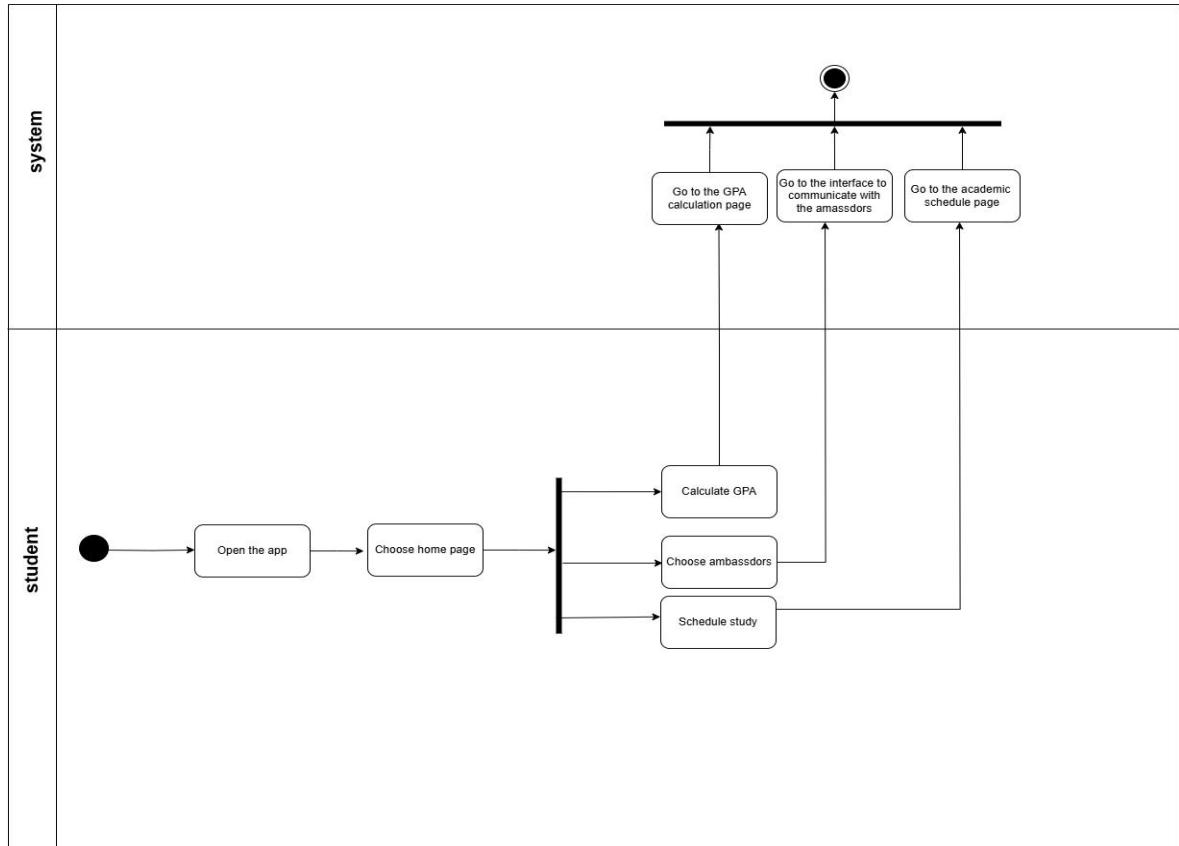


Figure 38 Activity Diagram Homepage

The activity diagram for Homepage, the student starts by opening the application on their device, which represents the starting point. After opening the application, the student navigates to the home page, which provides several options for interacting with the system. The activities available on the home page include:

GPA Calculation: The student can navigate to the GPA calculation page, where the system directly redirects them to this page upon selecting this option.

Communicate with Ambassadors: This option allows the student to access an interface for communicating with ambassadors. The system instantly directs them to the appropriate interface.

Study Scheduling: The student can choose this option to organize their study plan, leading them to a dedicated study scheduling page.

When the student selects any of these available options, the system immediately responds and directs them to the required page or interface, whether it's the GPA calculation page, ambassador communication interface, or study scheduling page.

The activity ends once the student selects one of the options and is redirected to the required page, where they can continue further interactions.

4.6.3 Activity Ambassadors

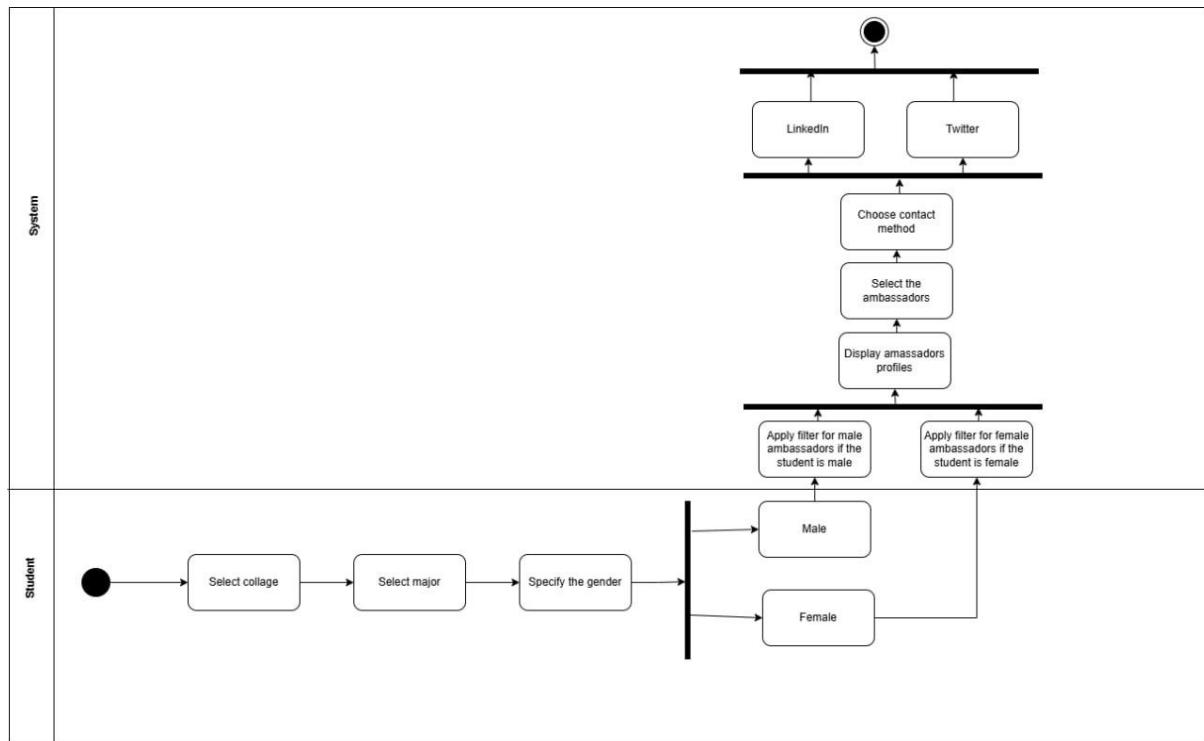


Figure 39 activity diagram ambassadors

The activity diagram for the ambassadors the student goes through several steps to choose an ambassador and communicate with them. The process starts with selecting the college they wish to contact an ambassador from, followed by choosing the academic major. Next, the student is asked to select their gender (male or female), and based on this choice, the system applies a search filter. If the student selects “male,” the system searches for male ambassadors, and if “female” is selected, it searches for female ambassadors. After applying the filter, the system displays the profiles of matching ambassadors, and the student is allowed to select the ambassador they wish to communicate with. The student is then prompted

4.6.4 Activity Absences

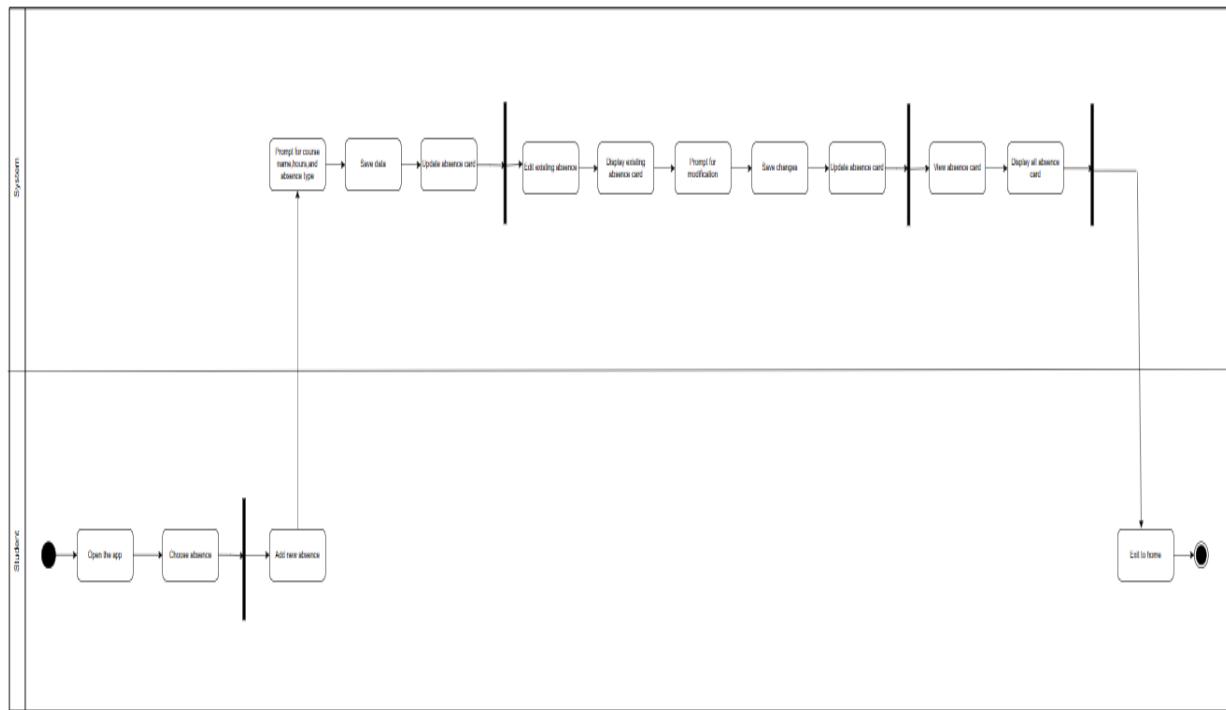


Figure 40 Activity Absences

The **activity diagram Absences** illustrates the workflow of managing absences within the app through three main options. The user (student) starts by opening the app and selecting “Manage Absences.” They can then add a new absence by entering details such as the course name, number of hours, and absence type, after which the data is saved, and the absence card is updated. The second option is to edit an existing absence, where the system displays previously recorded absence cards, allowing the user to make modifications and save the changes. The third option is to view absence cards only, where the system displays all registered cards. Finally, the student can either return to the home page or exit the app.

4.7 Conclusion

Muad aims to deliver a comprehensive academic management experience for students while adhering to the highest standards of security and privacy. The application ensures compatibility with existing systems used by educational institutions and provides user-friendly interfaces that enhance productivity and simplify time management for students.

Muad offers a seamless experience for both users and developers. Additionally, leveraging visual representations like System Architecture and Data Flow Diagram improves understanding of the system's structure and functionality.

Taking into account the constraints and assumptions related to data privacy and accuracy, this project provides a clear roadmap that focuses on user needs and institutional challenges, paving the way for an innovative and effective application that revolutionizes the educational experience.

5 CHAPTER 5 System Implementation and Validation

5.1 Introduction

In Chapter 5, we present the system implementation details by showcasing the main execution screens of the application, accompanied by detailed descriptions of the implementation steps. This chapter focuses on how the theoretical design is translated into practical application, explaining the stages and processes followed to meet the system's intended objectives.

5.2 Technology Stack

The following section provides a detailed overview of the technologies used in our project. We describe the programming languages, frameworks, libraries, and tools selected for the implementation of our solution. By explaining the reasoning behind these choices, we aim to showcase how each technology was effectively leveraged to meet the project's needs.

For our project, we selected a technology stack consisting of **Flutter** for application development and a robust backend powered by **Laravel**. Laravel is a PHP framework known for its elegant syntax and extensive ecosystem, which simplifies the development of web applications by providing ready-made features such as authentication, session management, and routing.

To manage our data, we used **MySQL**, a reliable and efficient database management system that ensures optimal performance and accurate data analysis. We also utilized **Table Plus** for database management, offering a user-friendly graphical interface for organizing and viewing data effectively.

To facilitate API testing and documentation, **Postman** was employed. This tool allowed us to verify client-server interactions and analyse data with ease. Additionally, **Visual Studio Code** served as our primary code editor, providing essential features like Git integration and debugging to support professional-grade coding and project management.

Together, these technologies enabled the seamless development of a comprehensive solution tailored to our project's requirements.

Flutter

Flutter is an open-source UI framework created by Google, known for its ability to support cross platform development. It allows us to build mobile, web, and desktop applications using a single codebase, which saves time and effort. With its extensive collection of pre-built UI components, fast development cycles thanks to hot-reload, and strong performance, Flutter makes it easier to create high-quality applications efficiently.

Self-Learning Material:

To learn Flutter, I followed the course "Flutter - Build iOS and Android Apps with Google's Flutter & Dart" on the "Academind" channel and "Flutter Crash Course" on the "Traversy Media" channel. Additionally, I followed a YouTube playlist called [Flutter Course Playlist by Code With Andrea](#), which covers advanced topics on building Flutter applications, improving performance, and designing responsive interfaces for developers at all levels.

Visual Studio Code

Visual Studio Code (VS Code) is a lightweight and flexible code editor developed by Microsoft, highly popular among Flutter developers. Its strong support for Flutter and Dart, combined with an extensive plugin ecosystem and user-friendly interface, makes it a powerful development tool. VS Code offers features like code completion, debugging tools, and Git integration, all of which contribute to a smoother and more efficient development experience. By utilizing Flutter, Firebase, Visual Studio Code, and GitHub, we can leverage the strengths of each technology to build a robust, cross-platform application with a scalable backend. Flutter allows us to create a visually appealing and responsive user interface, while Firebase provides a reliable and open-source backend solution. Visual Studio Code enhances the development process with Flutter-specific tools, making it easier and more productive.

Android Studio Code

Android Studio is the official Integrated Development Environment (IDE) for Android application development, created by Google. It provides a comprehensive suite of tools and features that streamline the process of building, testing, and deploying Android applications for phones, tablets, wearables, and more.

Self-Learning Material:

To learn how to integrate Android Studio with Flutter, I watched a series of YouTube videos, including tutorials like "[Flutter & Android Studio Setup](#)" and "[Building Flutter Apps in Android Studio](#)" on the "Flutter" channel and "CodeWithChris." These videos provide detailed steps on how to set up Android Studio for Flutter development, including installing the Flutter plugin, configuring the Android emulator, and running Flutter apps directly from Android Studio. This helped me understand how to leverage Android Studio's features while working on Flutter projects.

Postman

Postman is a tool used for testing and documenting APIs. It allows students to verify the functionality of the client-server interface, send requests, view responses, and analyse data effectively, making it essential for application testing and debugging.

Self-Learning Material:

To learn how to use Postman, I watched the course "Postman Tutorial for Beginners" on the "Academind" channel and "Postman - API Testing Tutorial" on the "Tech With Tim" channel. I learned how to send requests, test APIs, interact with server responses, and analyze data effectively.

MySQL with TablePlus

MySQL is a powerful database management system used for storing and retrieving data. When combined with TablePlus, it offers a seamless experience for students to execute queries, edit tables, and analyse data using an easy-to-navigate graphical interface, making database management more efficient.

Self-Learning Material:

To learn MySQL and TablePlus, I followed the course "MySQL Tutorial for Beginners" on the "Programming with Mosh" channel and "TablePlus Tutorial" on the "LearnCode.academy" channel. I learned how to manage databases, execute queries, and organize tables using a graphical interface to simplify database management.

Laravel Package

A Laravel Package is an extension for the Laravel framework written in PHP. It provides pre-built features like authentication, session management, and data handling, streamlining the development of web applications and reducing the time needed to implement complex functionality.

Self-Learning Material:

To learn Laravel, I followed the official documentation from [Laravel Documentation](#), which provides a comprehensive guide on installing and setting up Laravel. This guide covers everything from environment setup to configuration, and it's an essential resource for understanding Laravel's core features. Additionally, I watched tutorials on YouTube that explained how to use Laravel packages, including authentication and session management, to streamline web application development.

5.3 Collaboration Tools

Collaboration is vital for the success of any project. In this section, we outline the tools and platforms that facilitated smooth communication, task coordination, and teamwork among team members. From task management and scheduling to communication channels and progress tracking, we detail the resources that played a crucial role in fostering effective collaboration throughout the project.

Google Sheets

Google Sheets was the primary tool used to organize and manage tasks. It allowed the team to clearly outline and assign responsibilities for front-end development, back-end development, and machine learning tasks. The real-time collaboration features enabled team members to update and track progress seamlessly, ensuring transparency and accountability.

Screenshot of **Google Sheets** provided in **Appendix - Figure 81**

Microsoft Excel

Microsoft Excel was utilized to create a Gantt Chart for visualizing task timelines and deadlines. The chart provided a clear representation of the project schedule, highlighting the start and end dates of tasks, dependencies, and milestones. This tool was instrumental in maintaining a structured workflow and ensuring timely task completion.

Screenshot of **Microsoft Excel** provided in **Appendix - Figure 82**

Discord

Discord served as the primary communication platform for team meetings and discussions. Its voice channels and messaging features enabled efficient interaction among team members, fostering collaboration and idea exchange. Discord ensured that all team members remained connected and updated, regardless of their location.

Screenshot of **Discord** provided in **Appendix - Figure 83**

Jira

As the project grew and the volume of tasks increased, we needed a more advanced solution to accommodate the evolving requirements. Therefore, we moved to an upgraded version of Trello, which offered enhanced features to better handle the complexity of the project and ensure smoother collaboration between team members. This shift allowed us to maintain productivity and achieve project goals effectively.

Screenshot of **Jira** provided in **Appendix - Figure 84**

Google Drive

We use Google Drive to store and manage all data related to our work. It serves as a central repository for project documents, reports, and other important files. This cloud storage solution ensures that all team members have easy access to the necessary resources and can collaborate efficiently.

Screenshot of **Google Drive** provided in **Appendix - Figure 85**

Trello

Trello was used for task management and tracking project progress in the early stages. It allowed us to organize tasks visually, ensuring that team members could easily monitor project status and deadlines.

Screenshot of **Trello** provided in **Appendix - Figure 86**

GitHub

GitHub is a widely used web-based platform for version control and collaborative software development. We have chosen GitHub as our version control system to manage our project's source code. It provides features like code repositories, branches, pull requests, and issue tracking, facilitating seamless collaboration among team members, version control management, and code review processes.

Screenshot of **GitHub** provided in **Appendix - Figure 87-88**

5.4 User Interfaces (Latest Design)

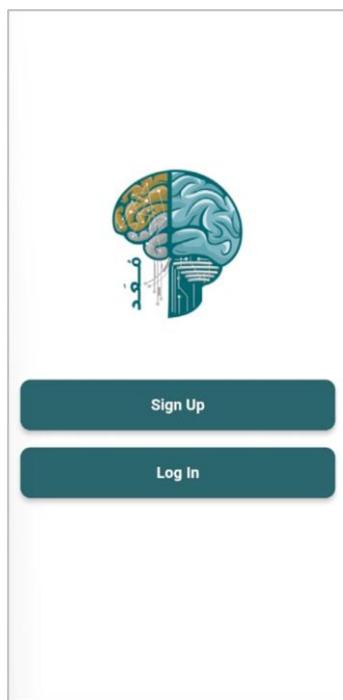


Figure 41 Application login method selection interface

The welcome screen of the application (Figure 41– Welcome screen) offers a clean and visually appealing interface designed for simplicity and ease of use. Below the logo, two primary action buttons are presented in a clear and user-friendly layout. The first button labelled “SING UP” (intended to be “SIGN UP”), allows new users to create an account, while the second button, labelled “LOG IN,” provides returning users with direct access to their accounts. Both buttons are styled with a dark green background and white text, ensuring high visibility and a professional appearance. The use of ample white space and the centered alignment of elements create a minimalist and modern design. This layout ensures that the user’s focus remains on the logo and primary navigation options, enhancing the overall user experience and providing a seamless entry point into the application.



Figure 42 Sing up screen

The sign-up screen of the application (**Figure 42 – Sign up screen**) provides a clean and intuitive interface to allow users to create new accounts efficiently.

Below the logo, the screen presents a series of input fields, each clearly labelled to guide the user through the registration process. The fields include: First name, Last name, Email , Phone number ,Password Icons beside the email, phone number, and password fields visually support their functions, improving clarity and user experience. Additionally, a password visibility toggle icon is placed next to the password field, allowing users to view or hide their input for accuracy. At the bottom of the screen, a prominent “SING UP” button (with a typographical error; should be “SIGN UP”) enables users to complete the registration process. Below this button, a subtle prompt reads “have account? Log in”, with “Log in” styled as a clickable link. This link allows existing users to navigate easily to the login screen, ensuring smooth interaction between registration and login processes.

The minimalist layout, ample white space, and centralized alignment of elements provide a modern and user-friendly design, enhancing the overall registration experience.

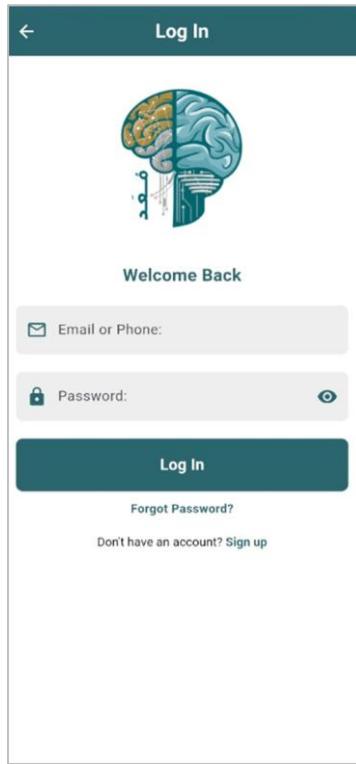


Figure 43 log in screen

The login screen of the application (**Figure 43-Log in screen**) features a clean and user-friendly design for secure and convenient access. At the top, the title "Log in" and a welcoming "Welcome Back" message set a friendly tone. The center displays the app's logo. Users can enter their email/phone and password in two clearly labeled fields, with icons for guidance and a toggle to show/hide the password. A bold "LOG IN" button enables access, while a "Forgot Password?" link provides recovery options. At the bottom, a prompt reads, "Don't have an account? Sign up," with the link leading to registration. The intuitive layout and ample white space ensure a seamless login experience.

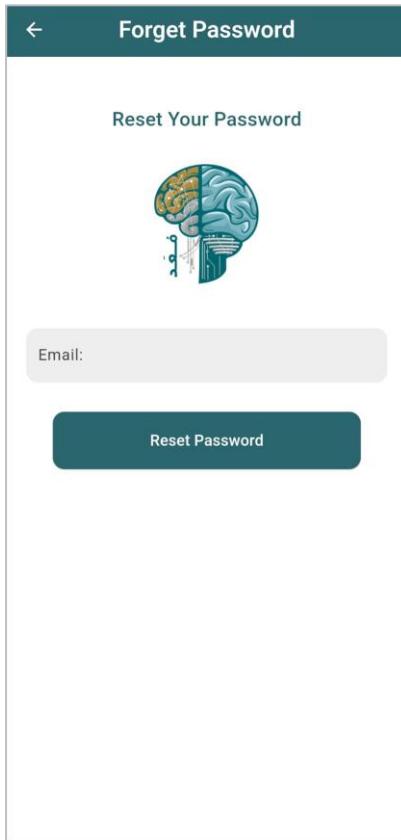


Figure 44 Forget password

The Forget Password screen of the application (Figure 44 – Forget Password screen) provides a straightforward and user-friendly interface to help users reset their passwords. At the top of the screen, a back button allows users to navigate to the previous screen seamlessly. The main section of the screen displays the text “Reset Your Password” to inform users of the screen’s purpose. In the centre of the screen, an input field labelled “Enter your email” allows users to input their email address, ensuring a smooth recovery process. The screen maintains a clean layout and uses a calming colour scheme, ensuring clarity and ease of use for the user.

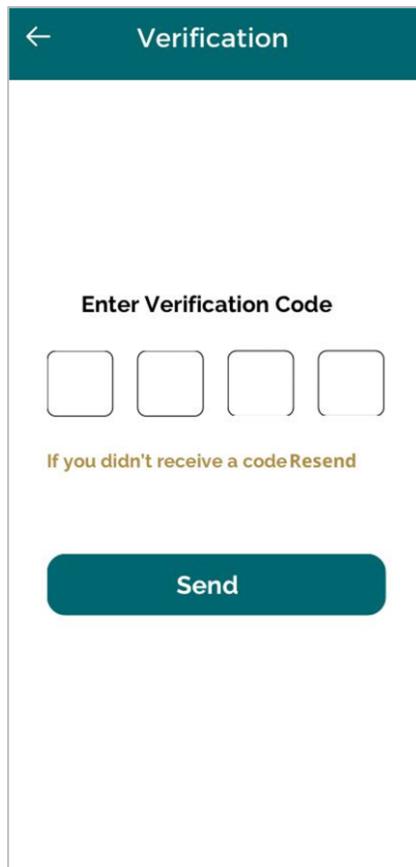


Figure 45 Verification

The Verification screen (Figure 45 – Verification screen) offers a minimalist and user-friendly design, helping users securely verify their accounts. At the top of the screen, a back button allows users to return to the previous screen. The main section features a bold header titled “Verification” to indicate the purpose of the screen. Below the header, a prompt instructs users to “Enter Verification Code” into four clearly displayed input boxes, ensuring a structured and easy input process. Beneath the input fields, a helpful message reads, “If you didn’t receive a code”, followed by a “Resend” link that allows users to request a new verification code. This feature ensures a smooth and supportive user experience. At the bottom of the screen, a prominent “Send” button serves as the primary call-to-action, allowing users to submit their verification code. The clean layout, clear labels, and user-focused design make the verification process simple and intuitive.

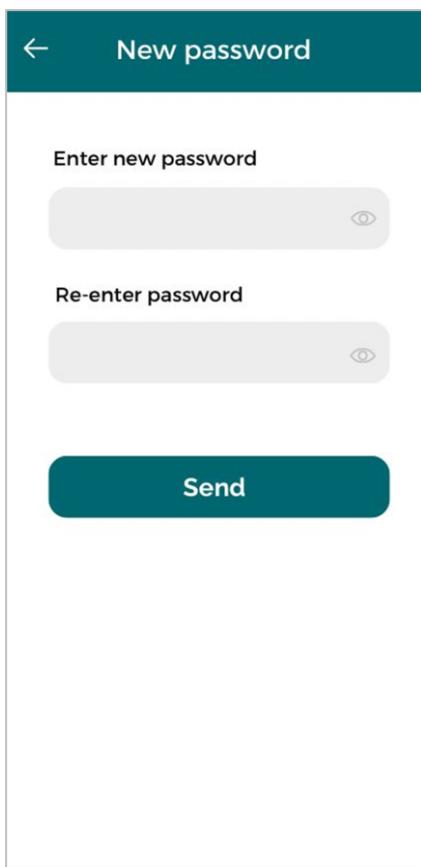


Figure 47 New Password

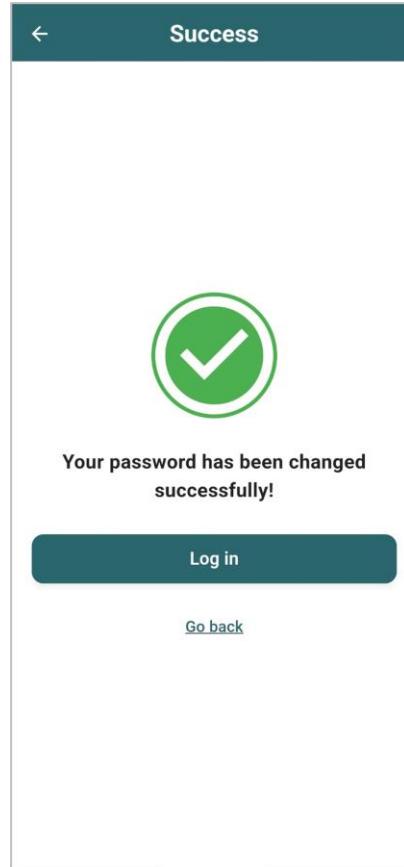


Figure 46 Password changed successfully

The New Password screen of the application (Figure 46-47 – New Password screen) offers a clean and intuitive interface, enabling users to securely update their password. At the top of the screen, a back button allows users to navigate to the previous screen. The main section of the screen contains two input fields labelled “Enter new password” and “Re-enter password” to ensure accuracy when setting the new password. Both fields include an eye icon that allows users to toggle the visibility of their password input. Beneath the input fields, a prominently displayed “Send” button provides a clear call-to-action for users to submit their new password. The layout is simple and organized, with ample white space, ensuring a smooth user experience while maintaining clarity and focus on the task.



Figure 48 Home screen

The Home Page of the application (Figure 48 – Home Page) offers a clean, user-friendly, and organized interface to help users easily access key services. At the top of the screen, the title “Home Page” is displayed, clearly identifying the current screen. Below the title, a personalized welcome message greets the user, displaying their name, which creates a welcoming and engaging experience. Centered on the screen is a series of four service icons, each accompanied by a descriptive label. These include Attendance, Student Schedule, Calculate GPA, and Ambassador. Each option is presented in a visually distinct, clickable box, ensuring users can quickly navigate to the desired service. At the bottom of the screen, a navigation bar provides access to the app’s primary sections: Home, Study Schedule, and Profile. The current tab, “Home,” is highlighted, giving users clear visual feedback. This streamlined layout ensures smooth navigation and enhances the overall user experience.

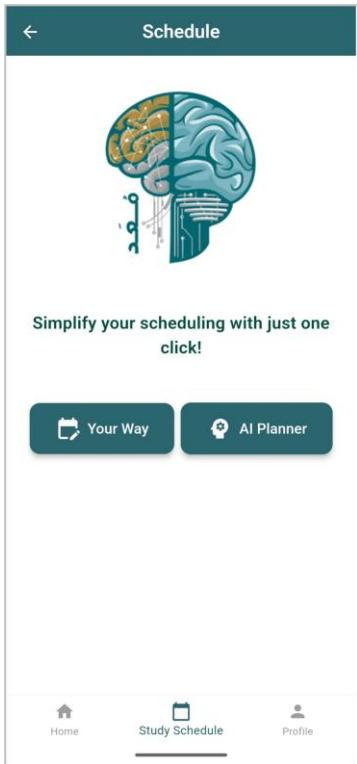


Figure 49 ways to create a schedule

Figure 50 schedule your way

The ways to create a schedule (Figure 49 – ways to create a schedule) provides users with an interface to guide users. Two primary buttons are displayed: “Your Way,” which enables personalized scheduling, and “AI Planner,” which provides AI-powered assistance for effective planning. These options give users the flexibility to customize their schedules based on their unique needs.

The “Your Way” screen (Figure 50– schedule your way) provides users with a personalized and straightforward interface to create their schedules based on personal preferences and course details. At the top of the screen, the “Your Way” heading is prominently displayed, indicating the primary focus of this feature. The interface is designed with ease of use in mind, and features input fields for basic course details. Users can enter the course name, number of course hours, and select preferred weekdays from a drop-down list. Additional fields allow for specific details, such as section number and lecture hall number, to be entered, providing a comprehensive structure for course planning. Two primary buttons are placed at the bottom: “Add Course,” which allows users to save course details, and “View Schedule,” which allows users to review their planned schedule.

Academic Schedule						
Time	Sunday	Monday	Tuesday	Wednesday	Thursday	
8:00-8:50						
9:00-9:50		ERP		HIC		
10:00-10:50		ERP		HIC		
11:00-11:50				HIC		
12:00-12:50						
1:00-1:50						
2:00-2:50						
3:00-3:50						
4:00-4:50						
5:00-5:50						

Figure 51 view schedule

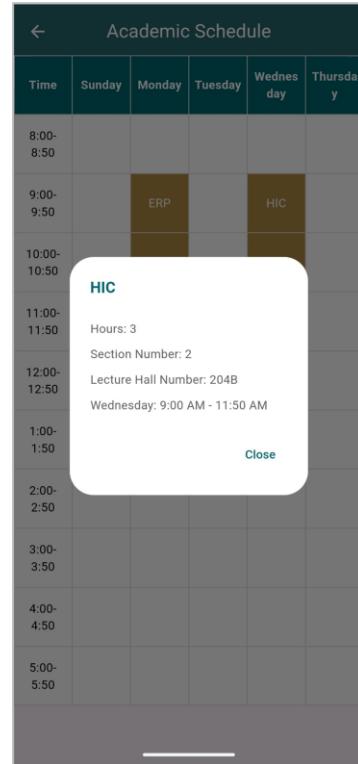


Figure 52 information box

The view Schedule Screen application (Figure 51-52 view Schedule Screen) provides users with a clear and organized interface to manage their weekly academic timetable. The screen is divided into columns representing each day of the week from Sunday to Thursday, with rows indicating specific time slots starting from 8:00 AM to 5:50 PM. This grid structure allows users to quickly identify their scheduled courses, such as "HCI" and "ERP," at their respective times and days. Upon clicking on a course field, an information box (as shown in Figure [21]) appears, displaying details about the course, including the course name, section, number of hours, day and time, and lecture hall. The use of clear labels for days and times ensures that users can easily navigate and understand their schedule. The interface is designed to be intuitive, with each course name placed in the corresponding time slot and day, providing a visual representation of the user's weekly commitments. This straightforward and organized layout enhances usability, allowing students to efficiently manage their academic responsibilities and plan their study time effectively.

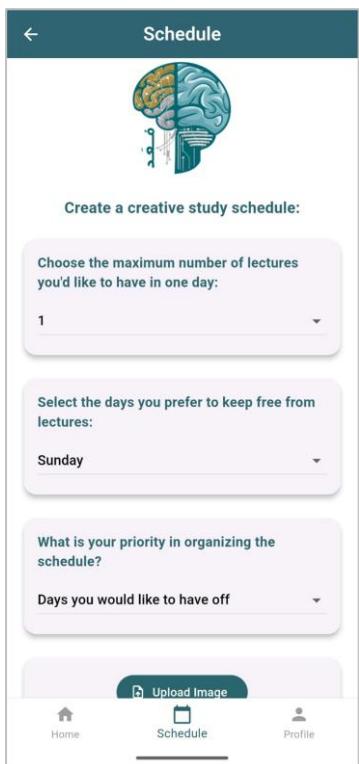


Figure 54 schedule AI Planner

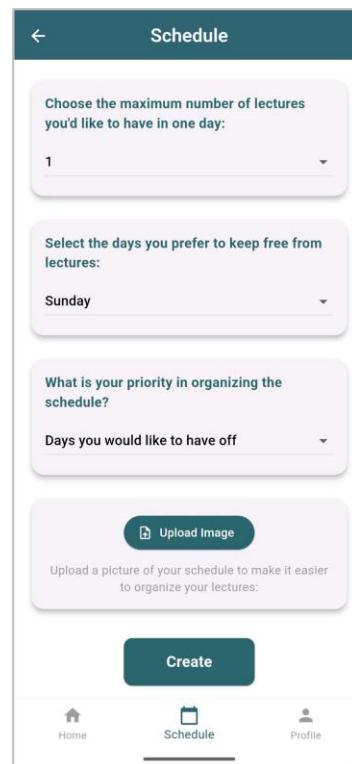


Figure 53 schedule AI Planner

The schedule AI Planner the application (Figure 53-54– schedule AI Planner) provides users with an intuitive and interactive interface to create a personalized study plan. At the top of the screen, the title “Schedule” is displayed, clearly indicating the purpose of the page. The screen guides the user through customizing their schedule with clear, well-labeled sections. Users can choose the maximum number of lectures they want in a single day using a dropdown menu, ensuring flexibility based on their preferences. Additionally, another dropdown menu allows users to select specific days they prefer to keep free from lectures, such as Sunday. Further down, users are prompted to indicate their priorities for organizing the schedule, ensuring a tailored experience that meets their study goals. At the bottom of the screen, a navigation bar provides quick access to other main sections, including Home, Study Schedule, and Profile, with the current tab highlighted for clarity. This thoughtful layout enhances usability, guiding users smoothly through the scheduling process.

The screenshot shows the Attendance Tracker application interface. At the top, there's a header bar with the title "Attendance Tracker". Below the header is a table with four columns: "Name course", "Hours", "Absent", and "Status". Two rows of data are present: one for "HCI course" (Hours: 6, Absent: 5, Status: UnExcused) and another for "ERP course" (Hours: 3, Absent: 8, Status: Excused). Below the table are two buttons: "Save" (grey) and "Edit" (blue). At the bottom, there are two cards: a grey card for "HCI course" (5 hrs absent, Status: UnExcused) and an orange card for "ERP course" (8 hrs absent, Status: Excused). The bottom navigation bar includes icons for "Home", "Schedule", and "Profile", with "Schedule" being the active tab.

Name course	Hours	Absent	Status
HCI course	6	5	UnExcused
ERP course	3	8	Excused

Figure 55 Absence screen

The Attendance Tracker screen in the application (Figure -55 Attendance Tracker Screen) provides a user-friendly interface for managing attendance records effectively. At the top, the title “Attendance Tracker” clearly indicates the purpose of the page. The main section features a structured table with four columns: Name, Course, Hours, Absent, and Status. Users can input or edit attendance details directly, with dropdown menus available to select the attendance status (e.g., “Excused” or “Unexcused”). This layout ensures that information is displayed in a clean and easily readable format. Below the table, two prominently placed buttons—Save and Edit—enable quick and efficient updates. At the bottom of the screen, color-coded cards summarize attendance for each course. These cards display the course name, the number of hours absent, and the attendance status. Red cards represent excused absences, while grey cards indicate unexcused absences. Each card includes a trash icon, allowing users to delete specific records easily. The screen’s intuitive and visually appealing design ensures a seamless experience for users to monitor and manage their attendance records efficiently.

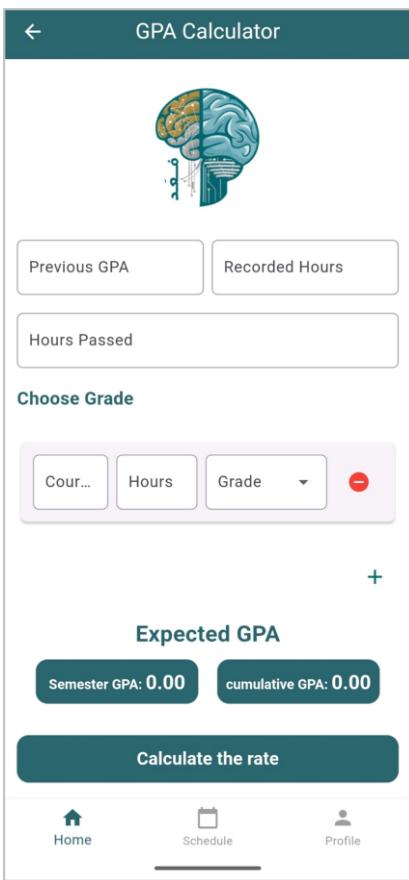


Figure 56 GPA calculator

The **GPA Calculator screen (Figure 56– GPA Calculator screen)** provides a user-friendly interface designed to help students calculate their semester and cumulative GPAs with ease. At the top, the title “GPA Calculator” is prominently displayed, accompanied by a back arrow for easy navigation. The main section is well-organized, featuring input fields for entering previous GPA, recorded hours, hours left, and hours passed, all clearly labelled for simplicity. Additionally, a course entry section allows users to input course details such as course name or identifier, credit hours, and grade, which can be selected from a dropdown menu. Users can add more courses using a "+" icon or remove them with a "-" icon for greater flexibility. To calculate GPA, two prominent buttons, “Semester GPA” and “Cumulative GPA,” are provided for instant results. At the bottom centre of the screen, a large circular display shows the calculated GPA, starting at 0.0. A navigation bar at the very bottom includes icons for Home, Study Schedule, and Profile, ensuring smooth transitions between different sections of the app. The overall design emphasizes clarity, usability, and engagement, allowing students to input their data effortlessly and obtain accurate GPA calculations.



Figure 57 ambassador screen

The Ambassadors screen of the application (Figure 57 – Ambassadors screen) offers a structured and visually appealing interface, making it easy for users to find and connect with ambassadors. At the top of the screen, the title “Ambassadors” clearly indicates the purpose of the page. Below the title, a search bar allows users to look for a specific ambassador by name, enhancing the user experience with a quick search functionality. Two filter buttons—All Specializations and All Genders—are positioned below the search bar, enabling users to refine their search based on criteria such as field of study or gender.

The main section of the screen displays ambassador profiles in card format. Each card includes the ambassador’s name, specialty, and a brief description. This concise format provides users with essential details at a glance. Additionally, each card features two icons: a LinkedIn logo for viewing the ambassador’s profile on LinkedIn and a cross icon for dismissing or hiding the card. These interactive elements make it easy for users to connect with or manage the list of ambassadors. The overall design ensures a seamless and efficient browsing experience, with a clean layout that balances functionality and aesthetics.

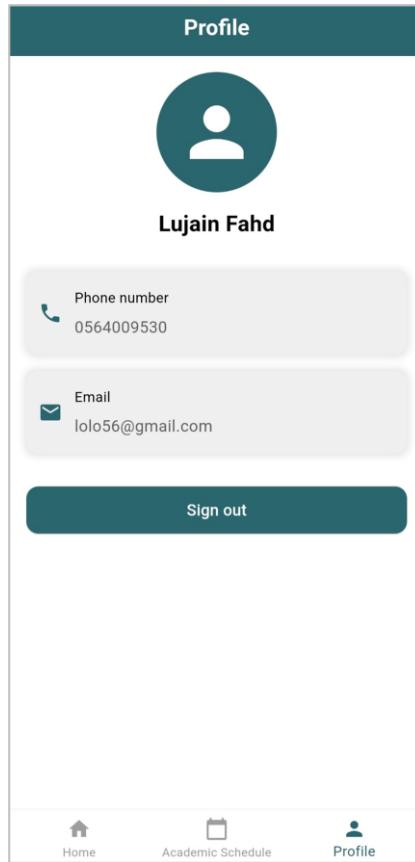


Figure 58 profile

The Profile screen of the application (Figure 58 – Profile screen) provides a minimalistic and straightforward interface for users to view and manage their personal information. At the top of the screen, the title “Profile” is displayed prominently, ensuring users are aware of the page’s purpose. Below the title, a circular icon is placed at the centre, symbolizing the user’s avatar. Directly underneath the avatar, the user’s name, “Nour Khaled,” is displayed in bold, providing a clear identification of the profile. The main section contains two fields displaying the user’s contact details: Phone Number and Email. These details are presented in labelled boxes with a clean and organized design, making them easy to read.

At the bottom of the screen, a Sign Out button is prominently positioned, allowing users to log out of their accounts securely. The button’s large size and distinct colour make it easily accessible. The overall layout of the Profile screen is simple and functional, ensuring a user-friendly experience with an emphasis on clarity and accessibility.

5.5 Implementation Details

5.5.1 Welcome Screen code

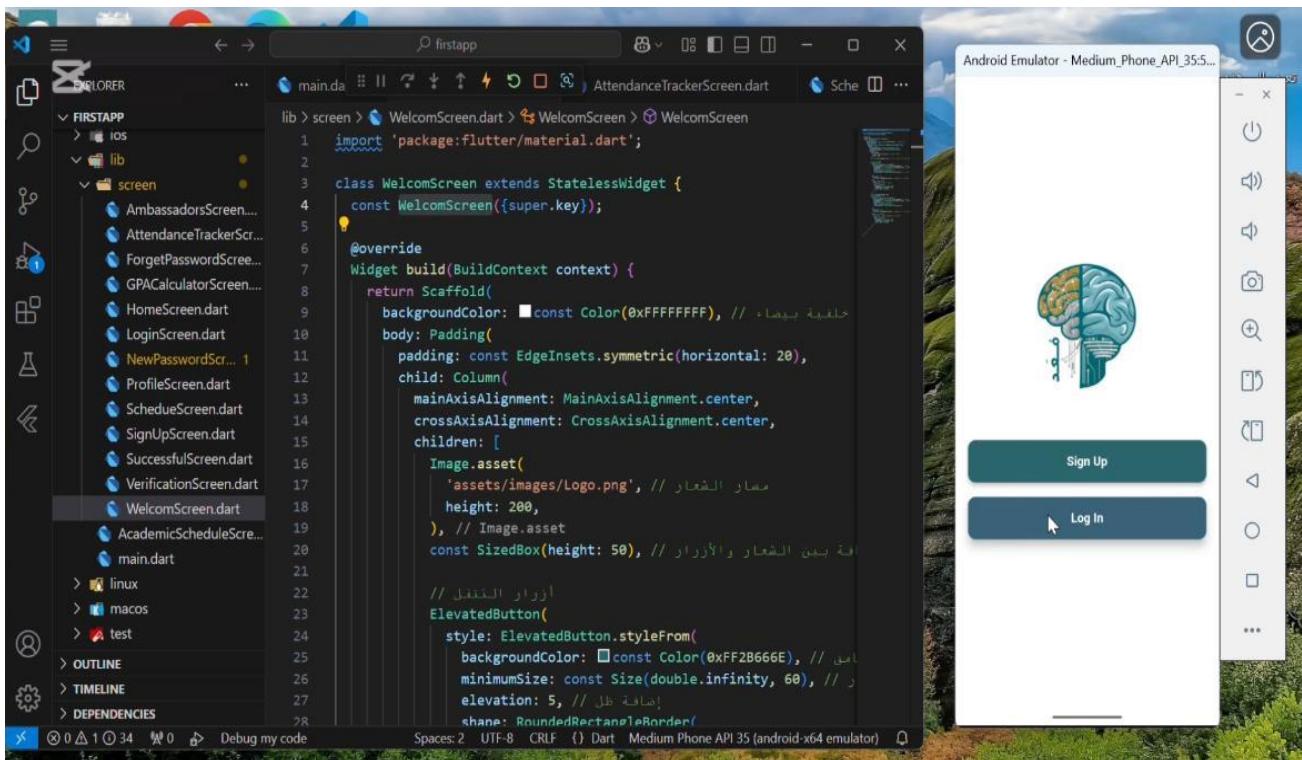


Figure 59 Welcome Screen code

The Welcome Screen is the first interaction point for users, offering a simple and visually appealing design that welcomes both new and returning users.

Main Features :

1. App Logo Display

- The application logo is prominently displayed using the [Image.asset] widget.
- The image path is specified as [assets/images/Logo.png], and the logo is shown at a height of 200 pixels to ensure a clear and recognizable visual identity for the app.

2. Interactive Buttons

- **Sign Up Button:** This button directs new users to the account registration screen to create an account.

- **Log In Button:** This button directs returning users to the login screen to access their accounts.
- Both buttons are implemented using the [ElevatedButton] widget, styled consistently with the app's design. Customizable properties such as [style] are used to define the button shape and size, ensuring an intuitive and user-friendly experience.

Responsive Layout

- The layout is structured using a [Column] widget, ensuring that all elements are centered both horizontally and vertically.
- Properties like MainAxisAlignment.center and CrossAxisAlignmentAlignment.center are used to create a balanced and responsive design that adapts to various screen sizes.

Main Functionalities

- **User Experience :** The screen provides a welcoming and user-friendly interface with clear navigation for both new and returning users.
- **Backend Integration :** The buttons are designed to connect to backend services such as Firebase Authentication to handle account creation and login processes.
- **Customizability :** The design is easily customizable to support additional features, such as personalized welcome messages or multi-language support.

5.5.2 Registration Screen (Sign Up) code

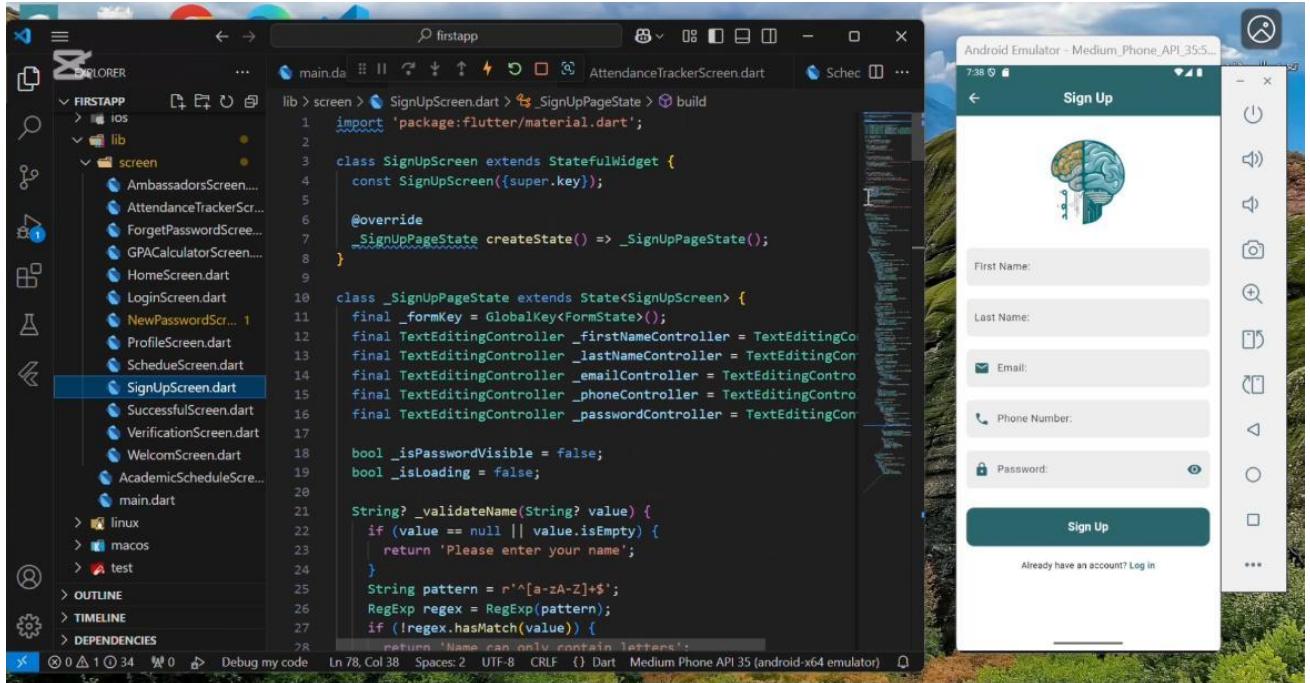


Figure 60 Registration Screen (Sign Up) code

The Sign Up Screen enables users to create a new account by filling out a simple, user-friendly form with clear labels and robust validation for accurate data entry.

Main Features:

1. Form Fields

- **First Name and Last Name:** Text input fields where users can enter their first and last names. These fields ensure proper formatting and validation to avoid empty submissions.
- **Email:** An email input field that validates the input against standard email formats (example@domain.com). If the entered email is invalid, an error message is displayed: "Please enter a valid email address."
- **Phone Number:** A phone number input field that ensures the number starts with "05" and contains exactly 10 digits. If the input does not meet these criteria, an error message is displayed: "Phone number must start with '05' and contain 10 digits."

- Password:

A password input field with validation for the following criteria:

- At least one uppercase letter.
 - At least one lowercase letter.
 - At least one number.
 - At least one special character (e.g., @, #, !).
- If the password does not meet these requirements, an error message is displayed:
"Password must include uppercase, lowercase, numbers, and at least one special character."

Validation Rules

- Fields cannot be left empty. If any field is blank, the app displays the error message:
"This field cannot be empty."
- **Real-time Validation:** Immediate feedback is provided as users fill out the form to guide them in correcting invalid inputs before submission.

Registration Process

1. Once the user fills out all fields and validation is successful:

- The `[_register()]` function is triggered to create a new user account using Firebase Authentication.
- The user's details are securely stored in Firestore, enabling future retrieval and app integration.
- The app updates the `[UserProvider]` with the user's data, ensuring seamless access across the app.

2. Upon successful registration:

- A toast message confirms the action: "Registration successful!"
- The app redirects the user to the **Home Screen** using the `[checkOnBoarding()]` function.

3. If errors occur (weak passwords, duplicate emails):

- Appropriate error messages are displayed using `FlutterToast`, ensuring the user is informed and can take corrective action.

User Experience

- **Clear and Accessible Design:** The form layout ensures all input fields are intuitive and easy to understand.
- **Error Prevention:** Real-time validation helps users fix errors before submission, streamlining the registration process.
- **Secure and Reliable:** User data is validated and stored securely in compliance with best practices.

5.5.3 Registration Screen (Log in) code

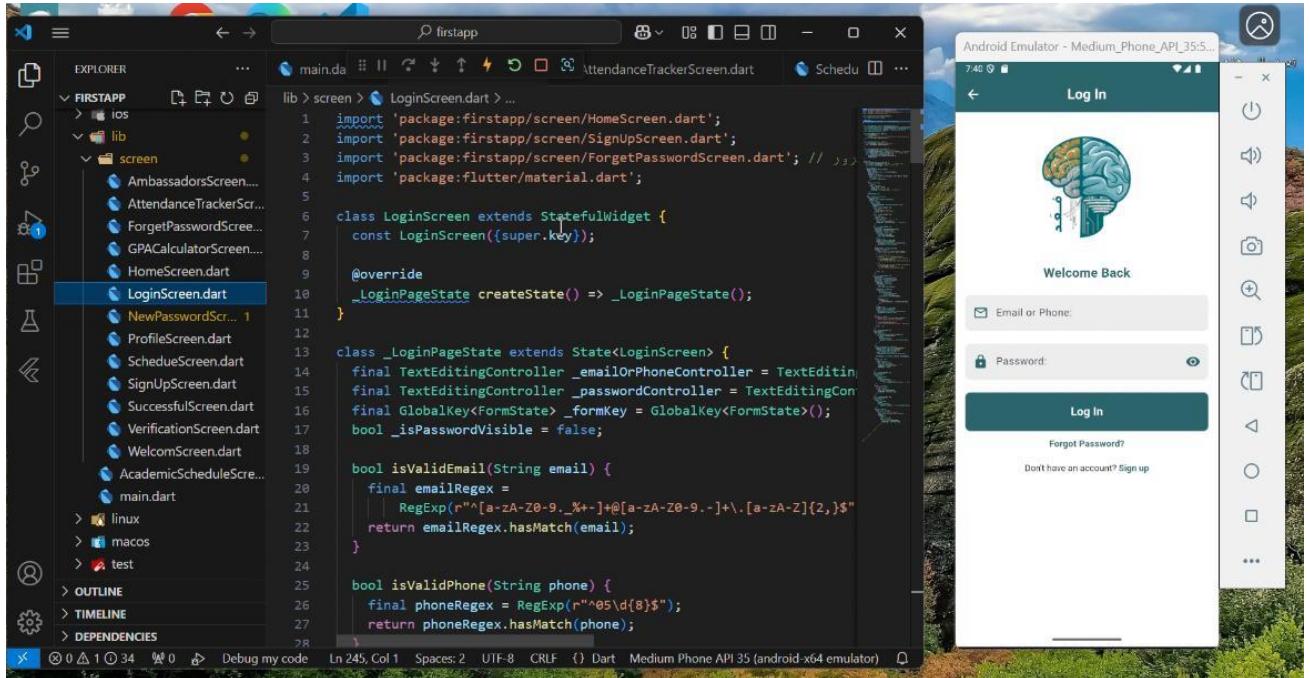


Figure 61 Registration Screen (Log in) code

The **Login Screen** provides users with a secure and user-friendly interface to access their accounts. The design focuses on clarity and ease of use, with real-time validation and a guided login process.

Main Features

1. Form Fields

- Email or Phone Number:

Users can input either their registered email or phone number.

- **Email Input:** Ensures the entered email follows standard email formatting (example@domain.com).

- If invalid, the app displays the error message: "Please enter a valid email address."

- **Phone Number Input:** Ensures the number starts with "05" and contains exactly 10 digits. If invalid, an error message is displayed: "Phone number must start with '05' and contain 10 digits."

- **Password:** A password input field with validation for the following criteria:
 - At least one uppercase letter.
 - At least one lowercase letter.
 - At least one number.
 - At least one special character (e.g., @, #, !).
- If the password does not meet these criteria, an error message is displayed:
"Password must include uppercase, lowercase, numbers, and at least one special character."

Validation Rules

- All fields are required. If a field is blank, the app displays the error message: "This field cannot be empty."
- Real-Time Validation: Validation is performed as the user enters their credentials, ensuring errors are corrected before submission.

Login Process

1. Once the user enters valid credentials and submits the form:
 - The `[_login()]` function connects to Firebase Authentication to validate the credentials.
 - Upon successful authentication:
 - The user's details are retrieved from Firestore.
 - The UserProvider is updated with the user's data for seamless app integration.
 - The user is redirected to the Home Screen.
 - If authentication fails (e.g., incorrect credentials), clear error messages are displayed to guide corrective actions.

Additional Features

- **Forgot Password:** A "Forgot Password" option is available to help users recover their credentials. This feature allows users to reset their password through a secure, email-based recovery process.

Success and Error Feedback

- **Successful Login:** A toast message confirms the action: "Login successful!"

The user is then redirected to the Home Screen.

- **Error Handling:** Comprehensive error messages are displayed for:

- Empty fields: "This field cannot be empty."
- Invalid inputs: "Please enter a valid email address." or "Phone number must start with '05' and contain 10 digits."
- "Invalid email, phone number, or password. Please try again."

User Experience

- **Guided Error Prevention:** Real-time validation and detailed error messages guide users to enter accurate credentials.
- **Security Focus:** Credentials are securely validated using Firebase Authentication to ensure user data protection.

5.5.4 Forget Password Screen code

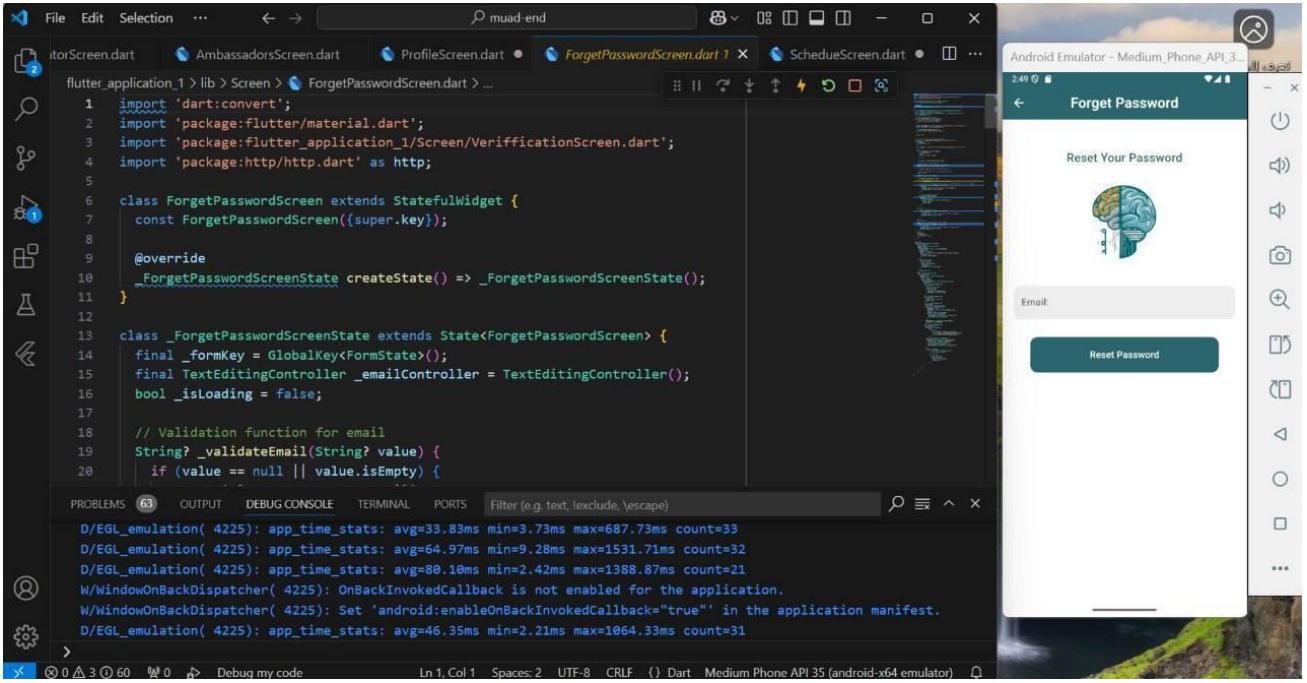


Figure 62 Forget Password Screen code

The **Password Reset Screen** allows users to recover their account access by requesting a password reset link. The design is responsive, user-friendly, and ensures a seamless experience across devices.

Main Features

1. Form Fields

- **Email Address:** A text input field where users enter their registered email address.
- The app validates that the email follows standard formatting (example@domain.com).
- If the email is invalid, an error message is displayed: "Please enter a valid email address."

Validation Rules

- **Required Field:** The email field cannot be left empty. If empty, an error message is displayed: "This field cannot be empty."
- **Real-Time Validation:** The app provides immediate feedback as the user enters the email to ensure the correct format.

Password Reset Process

1. Once the user enters a valid email address:
 - The "Reset Password" button becomes active.
 - Upon clicking the button, the app triggers a password reset request via Firebase Authentication.
 - A reset link is sent to the provided email address.

Success and Error Feedback

- **Success:** If the reset link is successfully sent, a toast message confirms the action: "Password reset email sent successfully!"
- **Error Handling:** If the email is not associated with any account or another error occurs, an appropriate error message is displayed using FlutterToast: "Email not found. Please check your email and try again."

Responsive Layout

- The screen is fully responsive and adjusts for various device sizes using MediaQuery, ensuring a smooth and consistent user experience across different screens.

User Experience

- **Streamlined Design:** The design emphasizes clarity and ease of use, allowing users to easily request a password reset with minimal steps.
- **Guided Error Prevention:** Real-time validation and error messages help users enter the correct email address and understand any issues promptly.
- **Security Focus:** The password reset process is securely handled via Firebase Authentication, ensuring that users can recover their accounts safely.

5.5.5 Verification Screen code

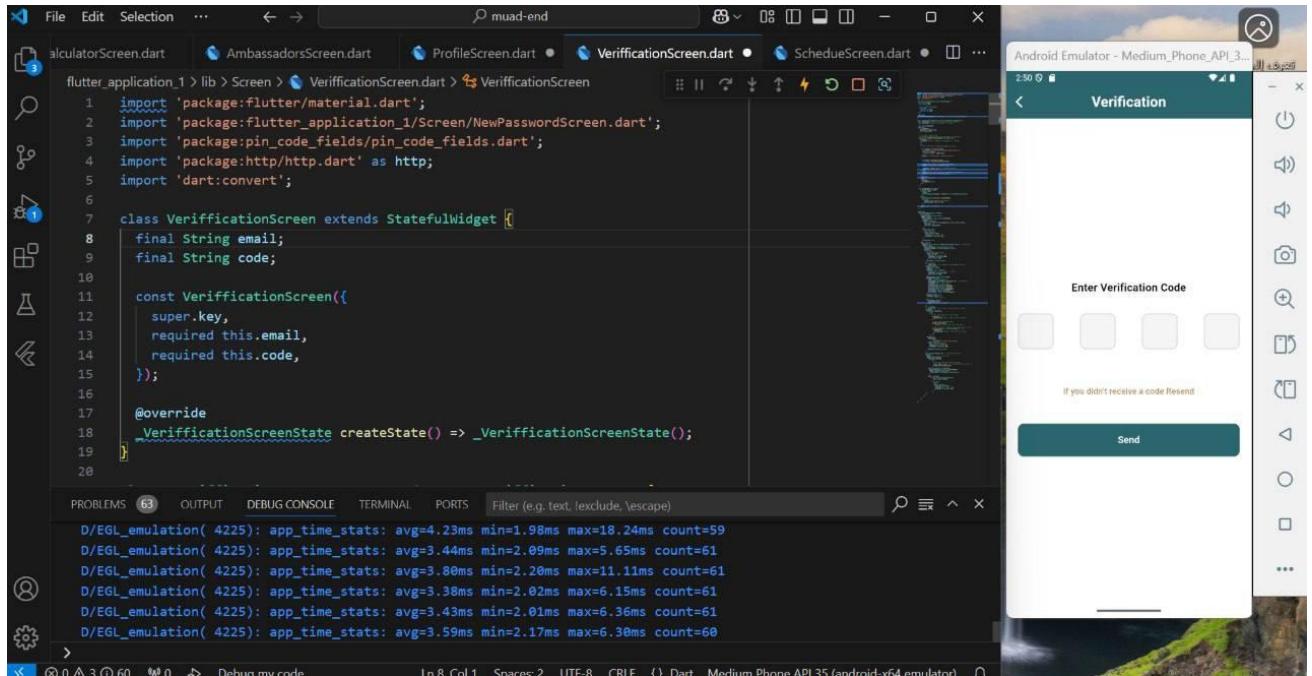


Figure 63 Verification Screen code

The Verification Screen is a key component of the password recovery process, providing users with a secure and user-friendly way to verify their identity using a code.

Main Features

1. Code Input:

- The screen displays four aligned input fields for entering the verification code sent to the user's email.
- Seamless navigation between input fields ensures an intuitive user experience.
- If the entered code is invalid or incomplete, an error message is displayed:
"Please enter the complete verification code."

2. Resend Code Option:

- A clickable "Resend Code" link is provided below the input fields.
- This allows users to request a new verification code if they did not receive the initial one, ensuring uninterrupted access to the recovery process.

3. Submission:

- Once the user enters the complete verification code:
- The "Send" button is activated.
- Clicking the button triggers a backend process to validate the code.
- Upon successful verification, users are redirected to the next step in the recovery process, such as the New Password Screen.

Success and Error Feedback

1. Successful Verification: A toast message confirms the success: "Verification successful!"

2. Invalid Code: If the code is invalid, an error message is displayed using FlutterToast:

"Invalid verification code. Please try again."

Navigation and User Interaction

- **Interactive Code Entry:** The screen supports smooth transitions between the four input fields as users enter the verification code.
- **Dynamic State Management:** The "Send" button is enabled only when all fields are filled with a valid code.
- **Seamless Navigation:** Using Navigator.push(), the app transitions users to the next step (**New Password Screen**) after successful verification.

User Experience

- **Clear and Accessible Design:** The interface ensures clarity, with clear labels and instructions guiding users throughout the process.
- **Immediate Feedback:** Success and error messages provide users with instant clarity on the verification status.
- **Error Prevention:** The **Resend Code** option ensures users can continue the process without disruption if they encounter issues with the original code.

5.5.6 New Password Screen code

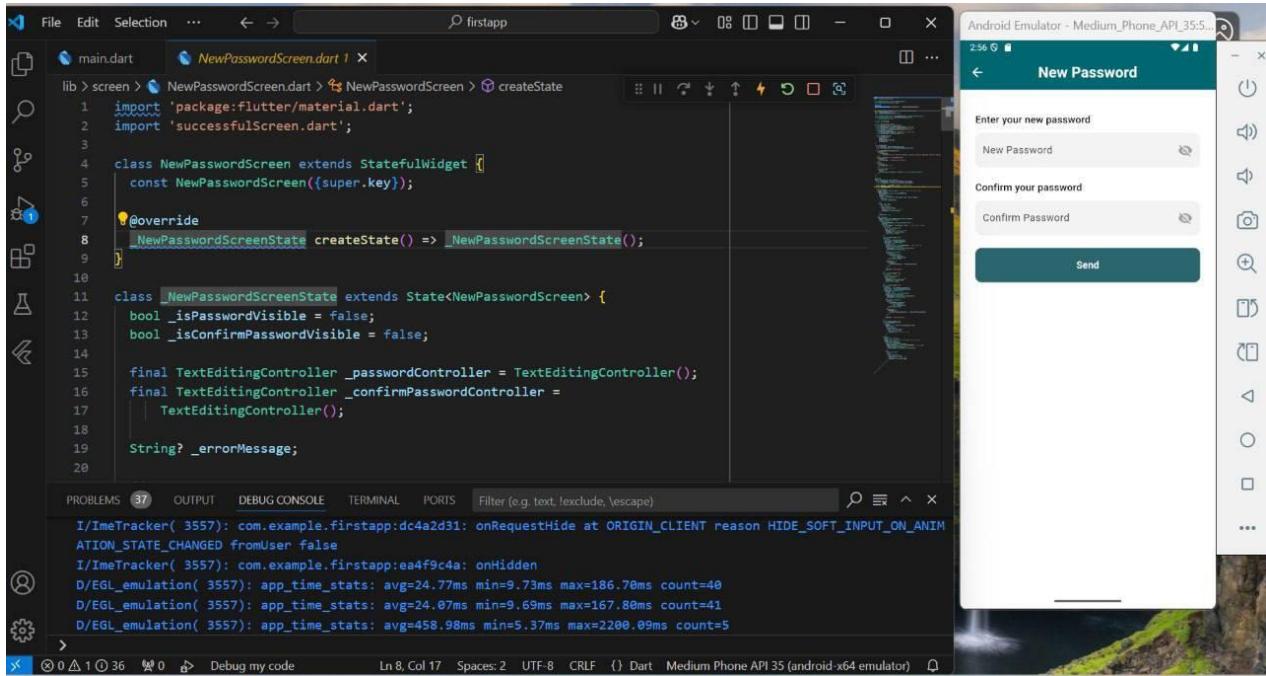


Figure 64 New Password Screen code

The New Password Screen provides users with a secure and intuitive interface to set a new password. The screen includes robust validation rules to ensure password security and prevent errors, offering a smooth and efficient experience.

Main Features

1. Form Fields:

- **New Password:** A password input field designed to enforce strong security requirements.
 - The password must meet the following criteria:
 1. At least 8 characters long.
 2. At least one uppercase letter.
 3. At least one lowercase letter.
 4. At least one number.
 5. At least one special character (@, #,!).
 - If the password does not meet these criteria, the following error message is displayed:
"Password must be at least 8 characters long and include an uppercase letter, lowercase letter, number, and special character."

- Confirm Password:

- A second password input field requiring the user to re-enter their new password.
- If the passwords do not match, the following error message is displayed: "Passwords do not match. Please enter the same password."

2. Validation Rules:

- Fields cannot be left empty. If a field is blank, the error message "This field cannot be empty." is displayed.
- Real-time validation is implemented, providing users with immediate feedback while entering their passwords.

Password Reset Process

1. Once both password fields pass validation, the password reset process begins.
2. The system securely updates the user's password in the authentication system (e.g., Firebase Authentication).
3. If the password reset is successful, users are redirected to the Success Screen with the confirmation message: "Your password has been successfully updated!"
4. If an error occurs (e.g., weak password or server issues), an appropriate error message is displayed using FlutterToast to notify the user.

Navigation and User Interaction

- Dynamic Navigation:** The app employs state management functions like `_onItemTapped()` to handle user input and navigation efficiently.
- Seamless Transitions:** Navigation between the New Password Screen and Success Screen is implemented using `Navigator.push()`, ensuring smooth transitions.

User Experience:

- Accessible Design:** The interface is user-friendly, making it easy for users to securely update their password.
- Clear Feedback:** Clear validation rules and error messages guide users effectively.

5.5.7 Success Screen “Password Change Confirmation” code

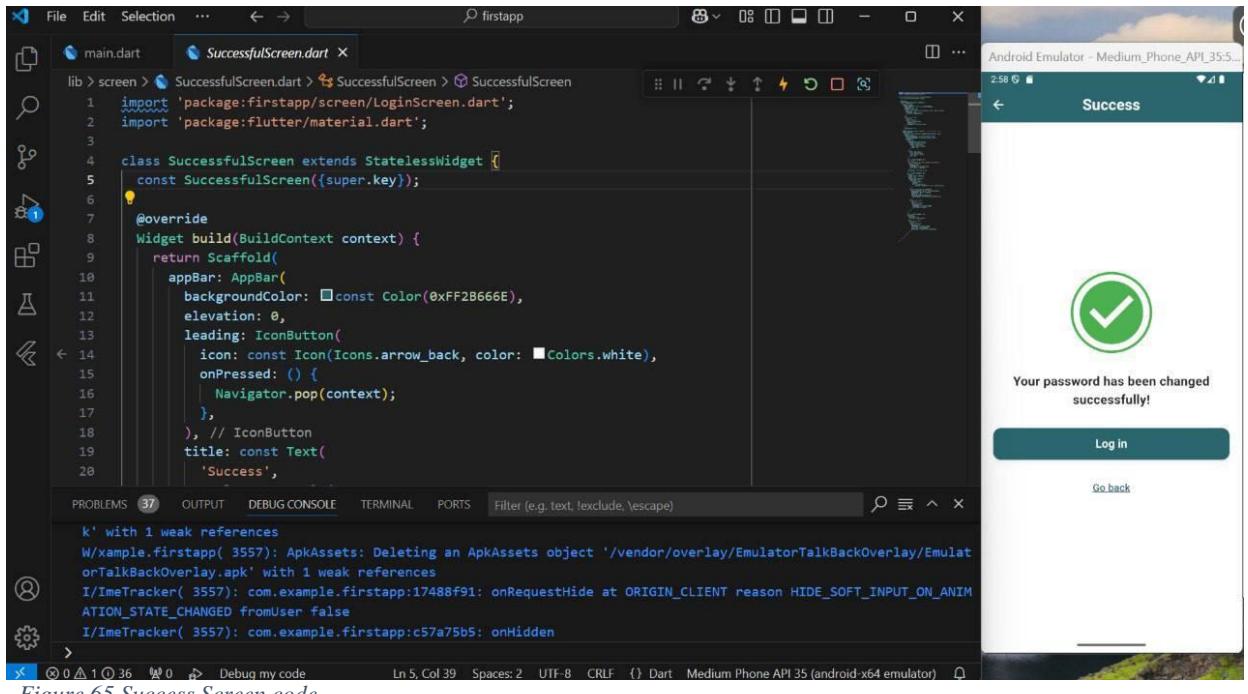


Figure 65 Success Screen code

The **Success Screen** serves as a confirmation hub, ensuring users are informed that their password has been successfully updated. This screen provides a seamless user experience with clear feedback and easy navigation for the next steps.

Main Features

1. Success AppBar (Navigation Bar):

- A back button (←) is displayed on the left, allowing users to return to the previous screen if needed.
- The title "Success" is centered at the top, ensuring clarity and focus.

2. Success Icon:

A prominent green checkmark icon is displayed at the center of the screen, visually confirming the successful password update.

3. Confirmation Message:

A clear and reassuring message is displayed: "Your password has been changed successfully!"

- This informs users that their action has been completed without errors.

4. Primary Action Button: A large and clearly visible button labeled "Log in" is provided to redirect users to the Login Screen, where they can access their account using the updated password.

5. Secondary Action: A text link labeled "Go back" is displayed below the primary action button, offering users the option to return to the previous screen if they need to review any details or information.

Navigation and User Interaction

- **Dynamic Navigation:** The app uses the `_onItemTapped()` function to manage user interaction and navigation efficiently.
- **Smooth Screen Transitions:** The `Navigator.push()` function is utilized to enable fluid and intuitive navigation between the Success Screen, Login Screen, and previous screens.

User Experience:

- **Accessible Design:** The layout is straightforward, ensuring users can quickly understand the screen's purpose.
- **Clarity in Functionality:** Every element on the screen is designed to guide users seamlessly, whether they want to log in or review prior steps.

5.5.8 Home Screen Main Screen code

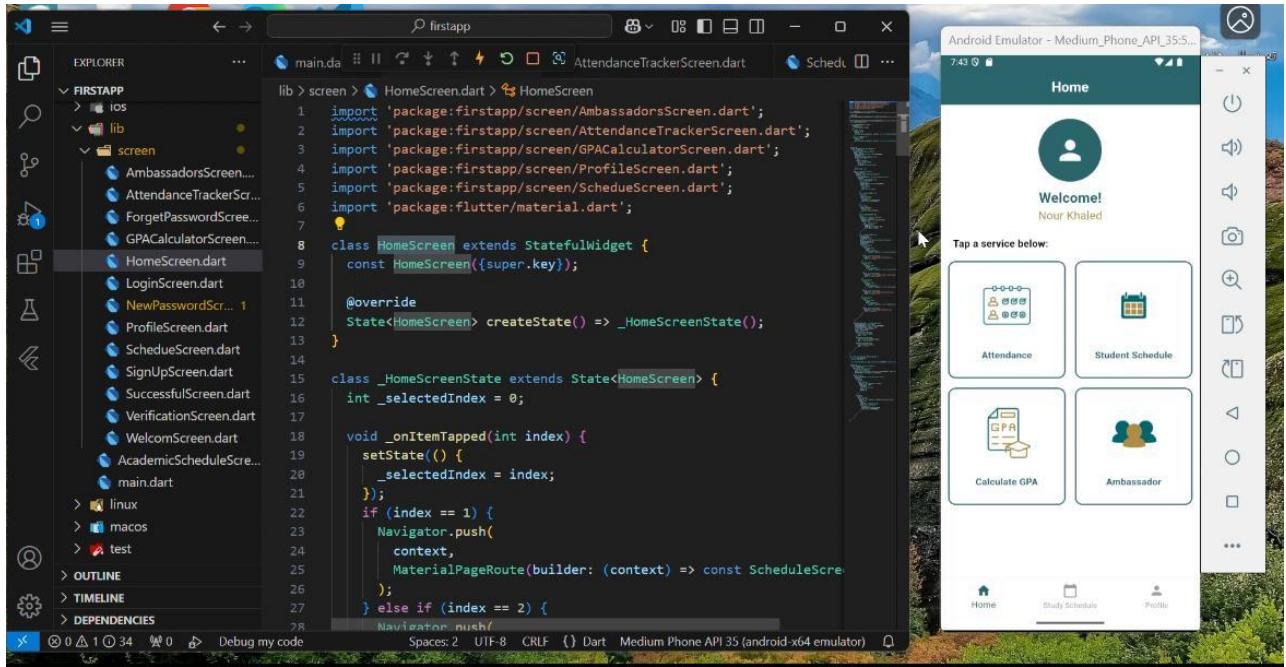


Figure 66 Home Screen Main Screen

The **Home Screen** serves as the central hub of the app, offering users a seamless and personalized experience. Upon logging in, users are welcomed with a friendly message and presented with essential app features, designed for easy navigation and efficiency. Below is a detailed description of the screen's functionality:

Main Features:

1. Personalized Welcome Message:

- The screen displays a custom greeting with the user's name, creating a sense of connection and uniqueness.

2. Guiding Message:

- A clear message at the top of the screen reads:
"Press on the service below."
- This helps users understand that the options below provide access to the app's features.

3. Core Services:

- Users can access the following essential features:

1. Attendance Tracking:

- An advanced tool to monitor attendance records accurately and effortlessly.

2. Student Schedule:

- A feature to create, manage, and view personalized study schedules.

3. GPA Calculator:

- A simple and intuitive tool for calculating grade point averages (GPA).

4. Ambassadors Section:

- A platform for connecting with distinguished students in their fields, providing guidance and support.

Bottom Navigation Bar:

The screen includes a navigation bar at the bottom for quick access to key sections of the app:

1. Home: Returns users to the main dashboard.

2. Study Schedule: Directs users to manage and view their study timetables.

3. Profile: Provides access to personal details and the option to log out.

Navigation and User Interaction:

- Interactive Navigation:

- The app uses dynamic state management with the `_onItemTapped()` function to ensure smooth transitions between sections.

- Screen Transitions:

- Navigation between screens is implemented using the `Navigator.push()` function, delivering a comfortable and intuitive experience.

User Experience:

- The design ensures accessibility and clarity, making it easy for users to find and use the app's core features.
- Each feature and navigation element is crafted to enhance engagement and usability.

5.5.9 Ambassadors Screen code

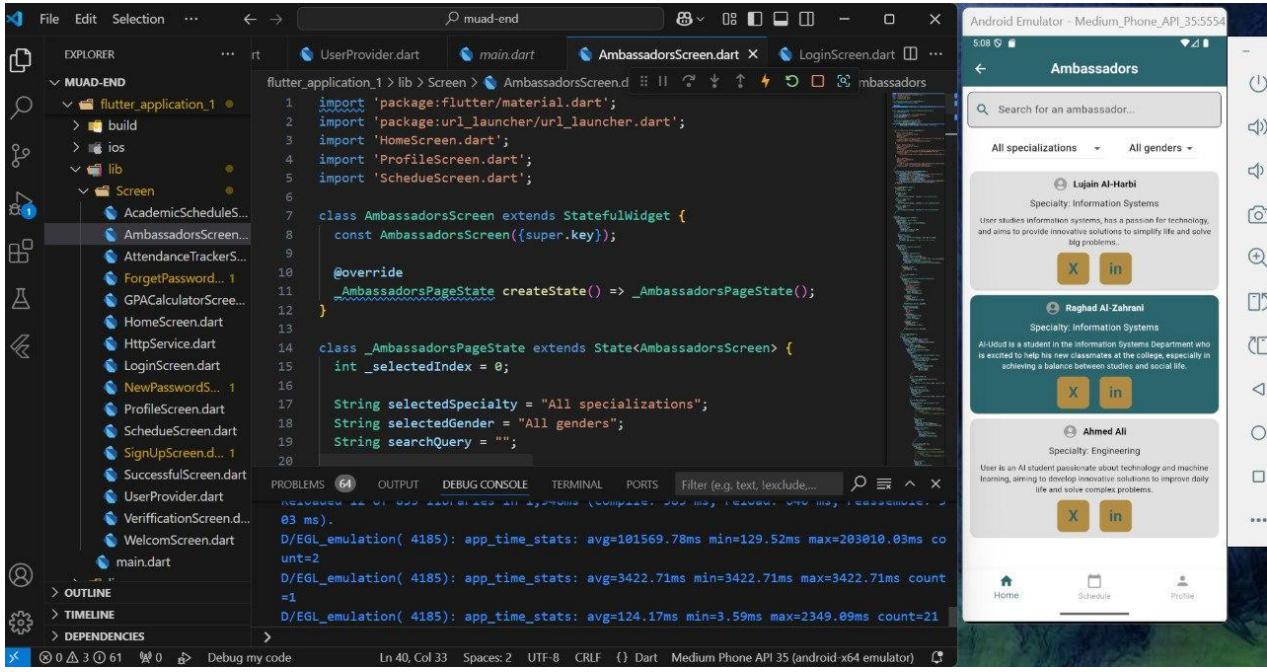


Figure 67 Ambassadors Screen code

The **Ambassadors Screen** is designed to facilitate connections with distinguished students who excel in their respective fields, offering users the opportunity to seek guidance and support. This user-friendly interface includes search and filter functionality to ensure users can quickly find the most relevant ambassadors. Below is a detailed description of the screen's functionality:

Main Features:

1. Search Bar:

- Positioned at the top of the screen, the search bar allows users to search for ambassadors by name or keyword.
- Users can input queries to refine the displayed list dynamically.

2. Filters:

- Two dropdown filters are provided to narrow down the list of ambassadors:
 - **Specialization Filter:** Allows users to select a specific field of expertise (default: All specializations).
 - **Gender Filter:** Enables filtering by gender (default: All genders).

3. Ambassador Cards:

- Each ambassador is displayed in an individual card containing:
 - **Name:** The ambassador's full name.
 - **Specialty:** The ambassador's area of expertise (e.g., Information Systems, Engineering).
 - **Description:** A brief introduction highlighting the ambassador's achievements and goals.
 - **Social Links:** Buttons for direct access to the ambassador's LinkedIn and other social platforms.

Bottom Navigation Bar:

The screen includes a bottom navigation bar for seamless access to key sections of the app:

- 1. Home:** Returns users to the main dashboard.
- 2. Schedule:** Directs users to manage and view their study timetables.
- 3. Profile:** Provides access to personal details and the option to log out.

Interactive Features:

- 1. Dynamic Search and Filters:** The displayed ambassador list updates instantly as users input search queries or adjust the filters.
- 2. Responsive Design:** The layout adapts to different screen sizes, ensuring a consistent user experience across devices.

User Experience:

- The screen is designed to enhance accessibility and efficiency, ensuring users can connect with ambassadors effortlessly.
- The clean layout and intuitive navigation provide a smooth and engaging experience.

5.5.10 GPA Calculator Screen

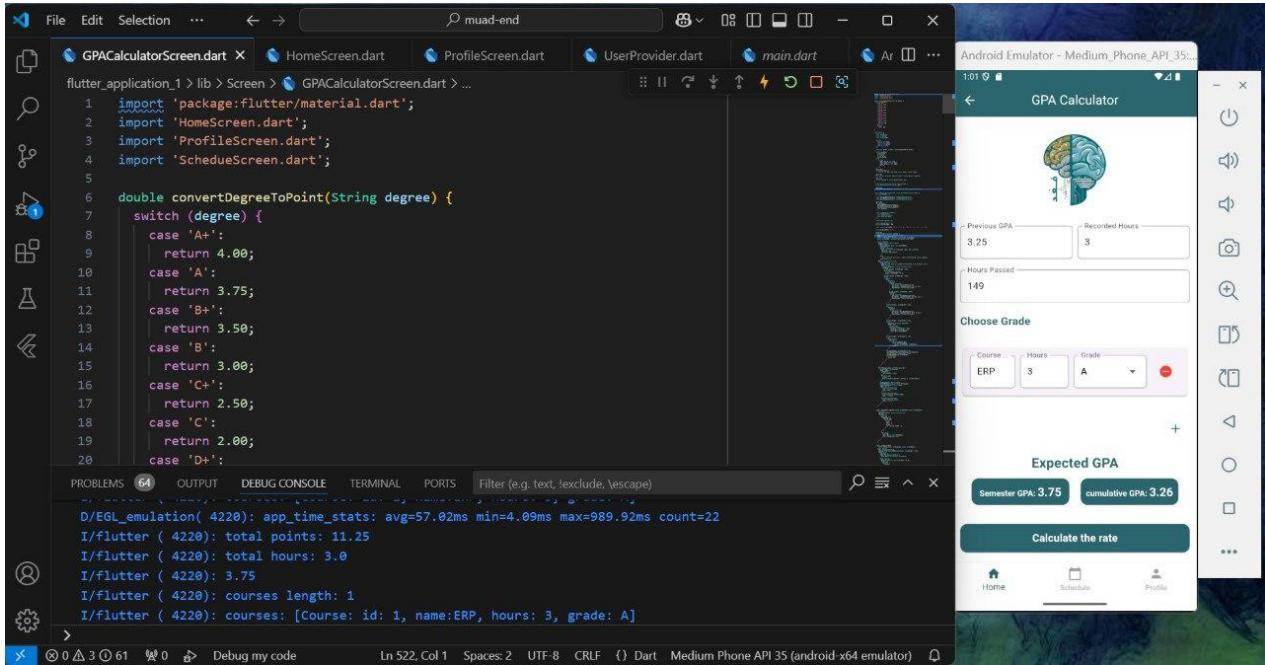


Figure 68 GPA Calculator Screen

The **GPA Calculator Screen** is designed to help users calculate their semester and cumulative GPA with precision and ease. This feature is intuitive and user-friendly, catering to students seeking to monitor their academic performance.

Main Features:

1. Input Fields

- **Previous GPA:** Users can input their cumulative GPA from previous semesters for accurate calculations.
- **Recorded Hours:** Users enter the total credit hours completed in previous semesters.
- **Hours Passed:** This field allows users to input the total number of credit hours they have passed.

2. Grade Input Section

- Users can add course details, including:
 - **Course Name:** A text field to enter the course title.
 - **Hours:** A numeric input for the credit hours of the course.

- **Grade:** A dropdown menu to select the achieved grade (e.g., A, B+, C).
- A "+" button enables users to add multiple courses, while a red delete icon allows removal of unwanted entries.

3. Output Section

- **Expected Semester GPA:** Displays the GPA calculated for the current semester.
- **Cumulative GPA:** Shows the updated cumulative GPA based on the entered data.

4. Calculate Button

- A large, clearly labeled "Calculate the rate" button initiates the calculation process.

Bottom Navigation Bar:

The screen includes a bottom navigation bar for seamless access to key sections of the app:

- 1. Home:** Returns users to the main dashboard.
- 2. Schedule:** Directs users to manage and view their study timetables.
- 3. Profile:** Provides access to personal details and the option to log out.

Interactive Features

- **Dynamic State Management:**
 - The calculation logic is implemented using real-time state management, ensuring accurate updates as users modify inputs.
- **User Feedback:**
 - Instant display of results in the output section enhances user experience.

User Experience

- The clean and organized layout ensures a smooth and efficient user experience.
- Flexible input options make it easy to calculate GPA for both semester and cumulative performance.

5.5.11 Attendance Tracker Screen code

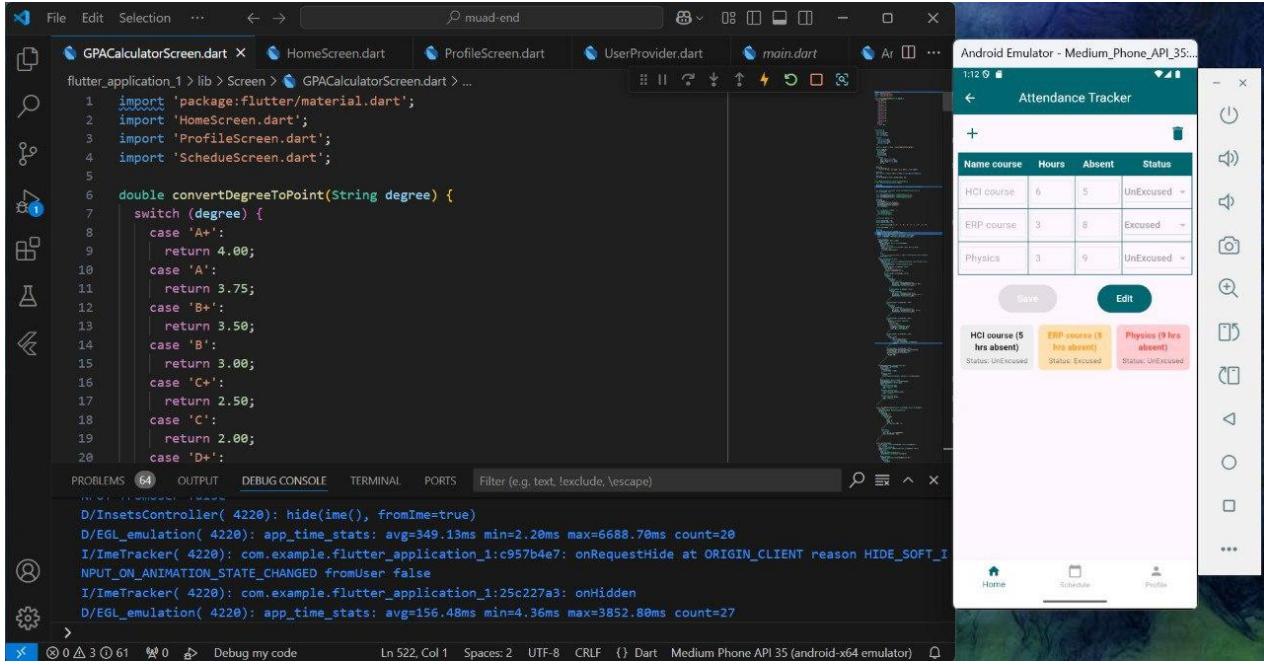


Figure 69 Attendance Tracker Screen code

The **Attendance Tracker Screen** helps users manage and monitor their attendance effectively. The design is intuitive and user-friendly, enabling users to track their attendance status and avoid warnings or deprivations due to absences. Below is a detailed description of the screen's functionality:

Main Features:

1. Input Fields:

- **Course Name:** A text field for entering the name of the course (e.g., HCI course, ERP course).
- **Total Hours:** A numeric input field to specify the total credit hours assigned to the course.
- **Hours Absent:** A numeric input field to record the number of hours the user has been absent for the course.

- Status:

- A dropdown menu to select the attendance status:
 - **Excused:** Absences with valid justification.
 - **Unexcused:** Absences without valid justification.

2. Attendance Summary Section:

- A card-based display summarizing attendance information for each course:
- **Course Name:** The name of the course is prominently displayed.
- **Absent Hours:** The total hours absent for the course.
- **Status:** The attendance status for the course.
- **Color-Coding:**
 - **Orange (First Warning):** Displayed if absences reach the first warning threshold.
 - **Red (Deprivation):** Displayed if absences exceed the allowable limit, resulting in deprivation.

3. Buttons:

- **Save:** A button that saves the entered attendance details.
- **Edit:** A button to modify existing attendance data for any course.

Bottom Navigation Bar:

The screen includes a bottom navigation bar for seamless access to key sections of the app:

- 1. Home:** Returns users to the main dashboard.
- 2. Schedule:** Directs users to manage and view their study timetables.
- 3. Profile:** Provides access to personal details and the option to log out.

Interactive Features

- **Dynamic State Management:** Real-time updates are applied as users modify attendance details, ensuring accuracy and efficiency.
- **User Feedback:** Visual cues, such as color-coded statuses and clear error messages, guide users effectively.

User Experience

- The organized and clean layout ensures an intuitive experience.
- Flexibility in input and real-time feedback enhances usability, making it easy to track attendance across multiple courses.

5.5.12 Profile Screen code

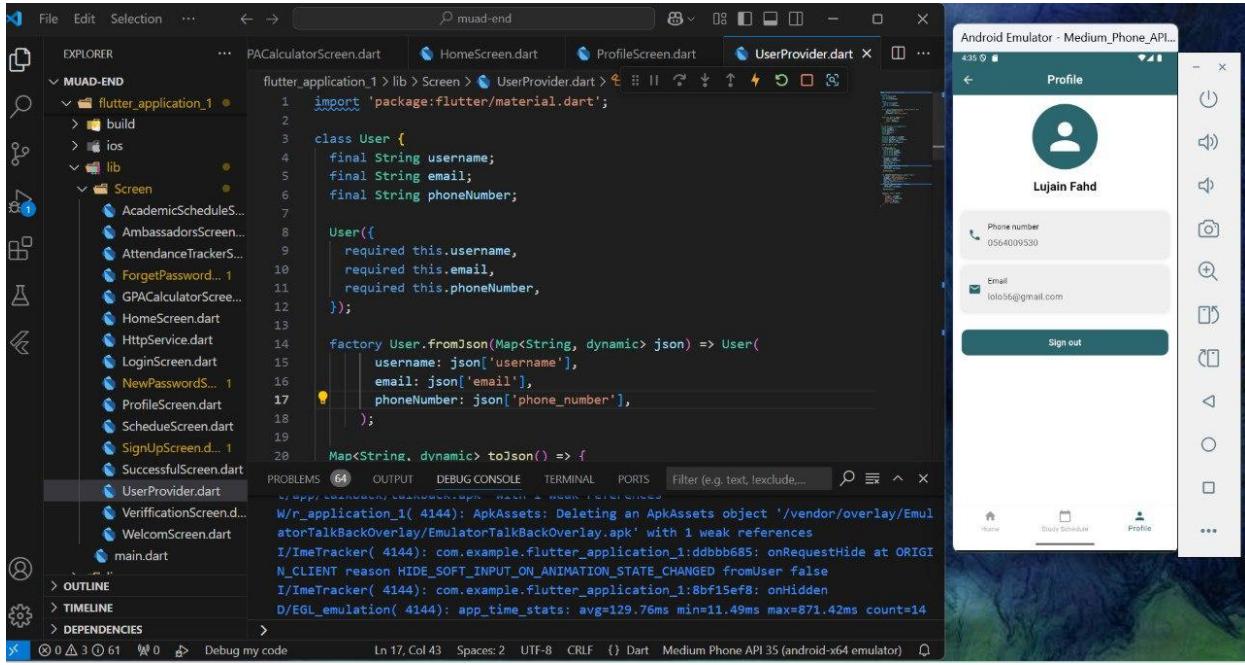


Figure 70 Profile Screen code

The **Profile Screen** provides users with a simple and clean interface to view their personal details and manage their account. The design ensures accessibility and focuses on presenting user information clearly while offering essential actions like signing out. Below is a detailed description of the screen's functionality:

Main Features:

1. Profile Header:

- Displays a circular avatar icon at the top of the screen.
- Shows the user's name beneath the avatar for a personalized experience.

2. User Information:

- **Phone Number:** The user's registered phone number is displayed with a phone icon for easy recognition.
- **Email:** The user's email address is shown with an email icon for clarity.

- **Sign-Out Button:** Positioned at the bottom of the screen, the "Sign Out" button allows users to log out of their account securely.

Bottom Navigation Bar:

The screen includes a bottom navigation bar for seamless access to key sections of the app:

- 1. Home:** Returns users to the main dashboard.
- 2. Schedule:** Directs users to manage and view their study timetables.
- 3. Profile:** Provides access to personal details and the option to log out.

Interactive Features:

- **Dynamic State Management:** Data displayed on the screen is retrieved dynamically from the UserProvider, ensuring it reflects the user's current account details.
- **Navigation and Transitions:** Seamless navigation between screens is implemented for a smooth user experience.

User Experience:

- The clean and minimal design ensures the screen is easy to navigate.
- Essential account details are prominently displayed, making the screen highly functional and user-friendly.

5.5.13 Machine Training Attempts

Problem 1: Image Processing and Data Extraction

Before updating the Umm Al-Qura University website, the data extracted from images containing course schedules suffered from inaccuracy and incompleteness. For instance, unrelated symbols (such as "@") appeared alongside course names like "Prophetic Biography," and data for classrooms and schedules included incomprehensible and irrelevant symbols. We attempted to use Optical Character Recognition (OCR) technology to train the system on the website's format. After extracting the data, it was cleaned of unwanted symbols and entered into a DataFrame. However, some fields in the final model remained empty.

Problem 2: Data Integration Between Frontend and Backend

After manually extracting data and applying filters, the team faced difficulties establishing a seamless connection between the frontend and backend when integrating filters with Google Cloud. Additionally, the idea of saving the final table in Excel format and storing it in a database for later display encountered challenges due to the lengthy time required to complete the process.

Solutions for Problem 2:

An alternative plan was devised to divide the options into two main categories:

1. "**Your Way**": Allows users to manually input data such as course names, credit hours, and classroom numbers.
2. "**Smart Organizer (AI Planner)**": Utilizes artificial intelligence to process data and automatically generate the schedule based on advanced algorithms.

We have documented our attempts to enhance system performance using the specified data and algorithms. The experiments have been recorded and organized as videos and related files, which can be accessed through the following link:

- **Google Drive Link:** [Click here to access the files](#)
- **Report Link:** [Click here to access the files](#)

5.5.14 ways to create a schedule Screen code

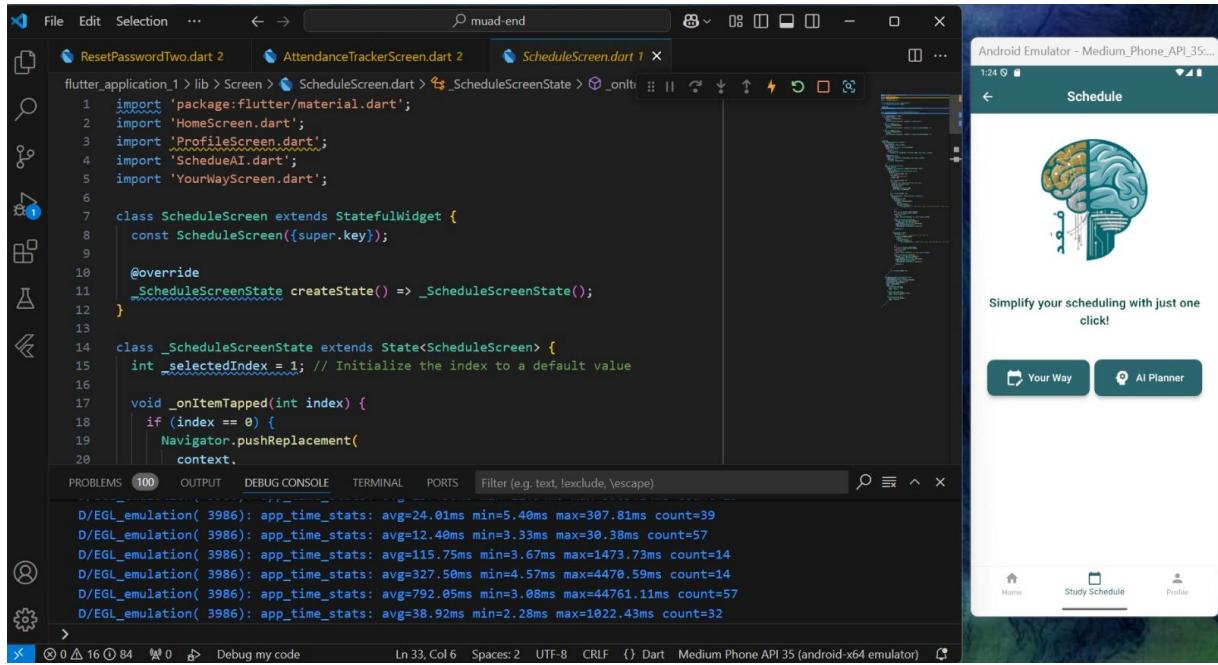


Figure 71 ways to create a schedule Screen code

ways to create a schedule Screen helps users effectively organize and manage their study schedules. The design is visually appealing and user-friendly, offering a seamless experience for creating and customizing schedules.

Main Features:

1. Visual Representation:

-The screen features an illustration of a brain divided into two sections: one representing personalized planning and the other symbolizing AI-powered intelligence.

-A motivational text beneath the illustration reads: "Simplify your scheduling with just one click!" encouraging users to explore the options provided.

2. Main Buttons:

- **Your Way:** A button that allows users to manually customize their study schedules based on personal preferences and needs.
- **AI Planner:** A button offering the option to generate a study schedule using artificial intelligence for greater efficiency.

Bottom Navigation Bar:

The screen includes a bottom navigation bar for seamless access to key sections of the app:

- 1. Home:** Returns users to the main dashboard.
- 2. Schedule:** Directs users to manage and view their study timetables.
- 3. Profile:** Provides access to personal details and the option to log out.

Interactive Features:

- **Simplicity:** The design ensures a visually comfortable and easily navigable interface.
- **Clarity:** Clear instructions and prominent buttons guide users effectively.
- **Dynamic Functionality:** Real-time updates occur as users make choices or navigate back from other screens.

5.5.15 Your Way Screen code

The screenshot shows a code editor with three tabs: 'ResetPasswordTwo.dart 2', 'AttendanceTrackerScreen.dart 2', and 'YourWayScreen.dart'. The 'YourWayScreen.dart' tab is active, displaying the following Dart code:

```
import 'package:flutter/material.dart';
import 'package:flutter_application_1/Screen/course_provider.dart';
import 'package:provider/provider.dart';
import 'AcademicScheduleScreen.dart';
import 'Course.dart';

class YourWayScreen extends StatefulWidget {
  const YourWayScreen({super.key});

  @override
  _YourWayScreenState createState() => _YourWayScreenState();
}

class _YourWayScreenState extends State<YourWayScreen> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  String courseName = "";
  int courseHours = 2;
  String sectionNumber = "";
  String lectureHallNumber = "";
  List<String> selectedDays = [ ];
}
```

The code defines a stateful widget 'YourWayScreen' that extends 'StatefulWidget'. It includes imports for 'flutter/material.dart', 'flutter_application_1/Screen/course_provider.dart', 'provider/provider.dart', 'AcademicScheduleScreen.dart', and 'Course.dart'. The class contains a constructor, an overridden 'createState' method returning '_YourWayScreenState', and a state class '_YourWayScreenState' that extends 'State<YourWayScreen>'. This state class includes a global key for the form, variables for course name, hours, section number, and lecture hall number, and a list for selected days.

Below the code editor is a terminal window showing log output from the Android emulator:

```
D/EGL_emulation( 3986): app_time_stats: avg=12.40ms min=3.33ms max=30.38ms count=57
D/EGL_emulation( 3986): app_time_stats: avg=115.75ms min=3.67ms max=1473.73ms count=14
D/EGL_emulation( 3986): app_time_stats: avg=327.50ms min=4.57ms max=4470.59ms count=14
D/EGL_emulation( 3986): app_time_stats: avg=92.05ms min=3.08ms max=44761.11ms count=57
D/EGL_emulation( 3986): app_time_stats: avg=38.92ms min=2.28ms max=1022.43ms count=32
D/EGL_emulation( 3986): app_time_stats: avg=2264.88ms min=3.25ms max=38388.58ms count=17
```

To the right of the code editor is a preview of the mobile application running on an Android emulator. The app has a dark-themed header with the title 'Your Way' and a brain icon. The main screen displays a placeholder text 'Enter the course details to create your schedule:' and three input fields: 'Enter Course Name:' (with a placeholder 'Course Name'), 'Enter Number of Hours:' (with a dropdown menu showing '2'), and 'Select Course Days:' (with a dropdown menu showing 'Select days'). Below these fields is a 'Course Details' section with two buttons: 'Section Number' and 'Lecture Hall Nu...'.

Figure 72 Your Way Screen code

Your Way Screen code helps users effectively organize and manage their study schedules.

The design is visually appealing and user-friendly, offering a seamless experience for creating and customizing schedules.

Main Features:

1. Course Input Form:

Course Name:

Input field with a placeholder that prompts users to enter the name of the course they want to add.

Number of Hours:

A dropdown menu for selecting the number of hours for the course, ensuring precise input.

Course Days:

A dropdown menu where users can select the days on which the course takes place.

Course Details:

Includes fields to enter additional information:

Section Number: Text input for specifying the section number of the course.

Lecture Hall Number: Text input for adding the lecture hall location, enhancing clarity.

2. Actions

-Add Course Button: Positioned prominently, allowing users to save the entered course details to their schedule.

-View Schedule Button: Provides quick access to view the compiled lecture schedule.

Interactive Features:

Dynamic State Management:

- All fields are connected to a state management system, ensuring user input is accurately captured and dynamically displayed.

Error Handling:

- Validation mechanisms (not shown but assumed based on common UI practices) ensure users input valid data before proceeding.

Smooth Navigation:

- Transition between this screen and the schedule view is seamless, maintaining user focus.

5.5.16 Popup Box Screen code

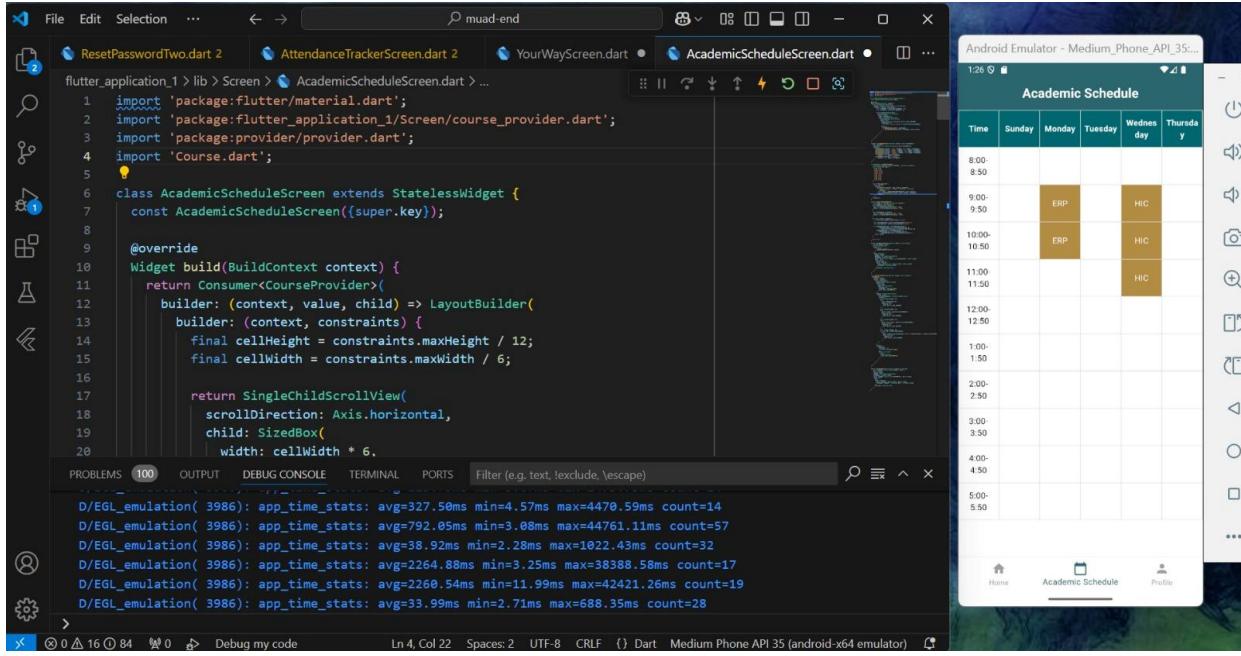


Figure 73 Popup Box Screen code

Popup Box Screen Code The popup box displays detailed information about a selected course from the schedule.

Main Features:

1. Course Input Form:

- **Course Name:** Input field with a placeholder that prompts users to enter the name of the course they want to add.
- **Number of Hours:** A dropdown menu for selecting the number of hours for the course, ensuring precise input.
- **Course Days:** A dropdown menu where users can select the days on which the course takes place.
- **Course Details:** Includes fields to enter additional information

Section Number: Text input for specifying the section number of the course.

Lecture Hall Number: Text input for adding the lecture hall location, enhancing clarity.

2. Actions

- **Credit Hours:**
Shows the total number of credit hours for the course.
- **Section Number:**
Displays the section number of the course.
- **Lecture Hall Number:**
Indicates the room number where the lecture takes place.
- **Lecture Timing:**
Displays the day and time in the format: "Day: Start Time - End Time".
Example: "Wednesday: 9:00 AM - 11:50 AM."

3. Close Button:

- Positioned at the bottom of the popup.
- Labeled as "**Close**", allowing users to dismiss the popup and return to the schedule.

Interactive Features:

- The popup box appears immediately when the user taps on a scheduled course cell.
- The design ensures the displayed information is clear and accessible.
- The close button provides a seamless user experience, allowing quick navigation back to the table.

5.5.17 AI Planner Screen code

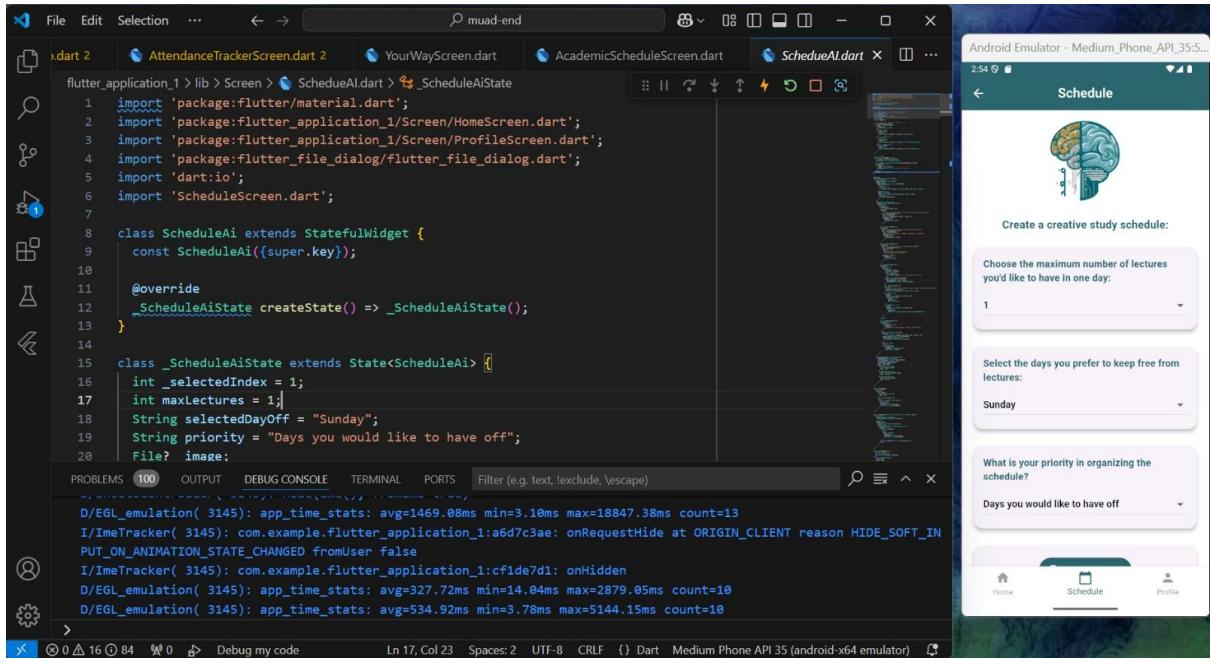


Figure 74 AI Planner Screen code

The **AI Planner Screen code** is a user-friendly tool designed to help users create a personalized study schedule that suits their preferences. The interface offers a clear and organized layout to simplify the process of making scheduling decisions.

Main Features:

1. Input Fields:

- **Select the maximum number of lectures per day:** A dropdown menu allowing users to specify the maximum number of lectures they would like to attend in a single day (e.g., 1 lecture).

- Choose preferred days off:

- A dropdown menu enabling users to select a day (Sunday..) they prefer to keep free from lectures.

- Set scheduling priority:

- A dropdown menu for defining scheduling priorities ("Days you would like to have off").

2. Image Upload Section:

- **Upload Image" Button:** Allows users to upload an image of their schedule to make organizing their lectures easier.

3. Create Button:

- A primary button at the bottom that saves the preferences and generates the study schedule accordingly.

Bottom Navigation Bar:

The screen includes a bottom navigation bar for seamless access to key sections of the app:

- 1. Home:** Returns users to the main dashboard.
- 2. Schedule:** Directs users to manage and view their study timetables.
- 3. Profile:** Provides access to personal details and the option to log out.

Interactive Features

- The image upload feature adds a visual aspect to enhance scheduling.
- Clear instructional text ensures smooth usage for users of all experience levels.

User Experience

- The screen's simple and intuitive design ensures that creating a study schedule is easy and enjoyable.
- Interactive dropdown menus make it effortless to select options.
- The "Create" button offers instant feedback to confirm actions.

6 CHAPTER 6 Testing and Evaluation

6.1 Introduction

Chapter 6 provides a comprehensive overview of the testing process, survey analysis, and future enhancements for the Muad application. This chapter highlights the importance of testing to ensure that the application meets both functional and non-functional requirements, delivering a seamless user experience. It also explores the feedback gathered through a survey, analysing responses to both open and closed-ended questions, which reflect user opinions on various aspects of the application, including design, usability, and features. Additionally, this chapter discusses the future enhancements planned for the application based on testing outcomes and user feedback, emphasizing continuous improvement to meet evolving user needs. Through rigorous testing, in-depth survey analysis, and a clear vision for future upgrades, this chapter lays the foundation for the sustainable improvement and development of the application.

6.2 Testing

The testing table provides a detailed and structured overview of the testing performed on key components of the Muad application. Each component, such as registration, Email Verification, Attendance and Calculate GPA, was thoroughly evaluated to ensure the system functions smoothly and delivers seamless user experience. The testing focuses on both functional aspects and usability, helping to identify and address any potential issues during the development phase.

Component	Test Description	Test Steps	Component Test Result	Pass/Fail
Registration	Verify that the registration form is displayed and functional.	<ol style="list-style-type: none"> 1. Open the registration page. 2. Enter valid user details (name, email, password). 3. Submit the form. 	Registration forms functional.	Pass
Email Verification	Ensure the system verifies the email address.	<ol style="list-style-type: none"> 1. Register with a valid email. 2. Check for a verification email. 3. Click the verification link. 	Email verified successfully.	Pass
Save User Data	Verify that user data is securely stored in the database.	<ol style="list-style-type: none"> 1. Register a new user. 2. Verify data is saved in the database. 3. Attempt to retrieve user information. 	Data stored securely.	Pass
Redirect to Home	Ensure successful redirection after registration.	<ol style="list-style-type: none"> 1. Register a new user. 2. Check if the system redirects to the home page upon completion. 	Redirected to home page.	Pass
User Login Form	Verify the login form is functional.	<ol style="list-style-type: none"> 1. Navigate to the login page. 2. Enter valid credentials. 3. Submit the form and verify login success. 	Login form functional.	Pass
Track Attendance	Ensure users can specify attendance for courses.	<ol style="list-style-type: none"> 1. Navigate to the attendance section. 2. Enter attendance for a specific course. 3. Verify the data is saved successfully. 	Attendance tracked.	Pass

Table 11 Software Testing

Component	Test Description	Test Steps	Component Test Result	Pass/Fail
Display Attendance	Verify the system displays attendance records.	1. Enter attendance data. 2. Check the summary view to confirm correct display of attendance records.	Attendance displayed	Pass
Input Grades	Ensure users can enter grades and credit hours.	1. Navigate to the grades section. 2. Input grades and corresponding credit hours. 3. Confirm data is saved and displayed correctly.	Grades entered.	Pass
Calculate GPA	Verify GPA calculation based on entered grades.	1. Input grades and credit hours. 2. Verify the calculated GPA matches the expected result.	GPA calculated.	Pass
Ambassadors Interface	Verify the display and selection of ambassadors	1. Select the major and gender filters. 2. Verify that ambassador cards are displayed. 3. Select an ambassador and click their Twitter or LinkedIn link. 4. Confirm redirection to the selected ambassador's profile.	Ambassador display and selection work successfully	Pass
Your Way	Verify the ability to add and view courses	1. Enter the course name, credit hours, section name, and room number. 2. Select the lecture day and time (start & end). 3. Click "Add Course" to verify the course is saved. 4. Click "View Schedule" to check if the schedule is displayed correctly.	Course addition and schedule display work successfully	Pass

Table 12 Software Testing 2

The Testing Table (Table 11-12 – Software Testing Table) documents the purpose, process, and results of each test. It begins with the test description, which outlines the specific functionality being assessed, such as verifying the smooth operation of the registration process or ensuring the accuracy of GPA calculations. Following this, the test steps provide a detailed sequence of actions performed during testing to maintain consistency across evaluations.

The component test result records the outcome of each test, indicating whether the expected behavior was achieved. Lastly, the pass/fail status reflects the success of each component in meeting the criteria. For the Muad application, all components passed their respective tests, demonstrating the system's readiness for use and ensuring a seamless and effective user experience.

6.3 Survey Analysis

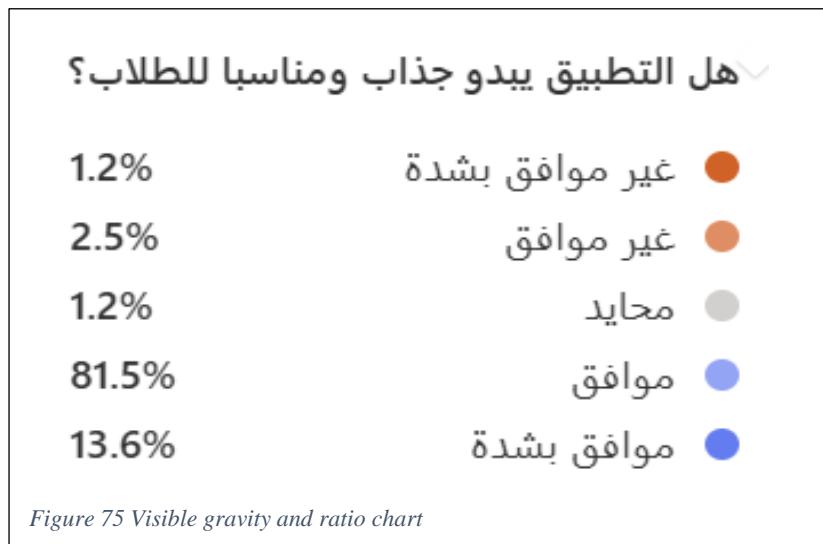
The survey analysis section provides insights into user feedback collected through a structured questionnaire aimed at evaluating various aspects of the Muad application. With 81 responses gathered, the survey included both open and closed-ended questions, offering users the opportunity to share their opinions on the application's functionality, design, and overall experience.

This section examines key topics, such as the visual appeal of the application, the relevance of features like the login process, the effectiveness of its functionalities, and users' suggestions for improvements or missing sections.

Analyzing this feedback helps identify areas where the application performs well and highlights opportunities for future enhancements.

The responses also reflect how users perceive the value of the application in improving their academic or organizational experiences and whether they would recommend it to others. This survey analysis provides valuable direction for refining the application to better meet user expectations and enhance overall satisfaction.

6.3.1 Visual Appeal and Appropriateness for Educational App

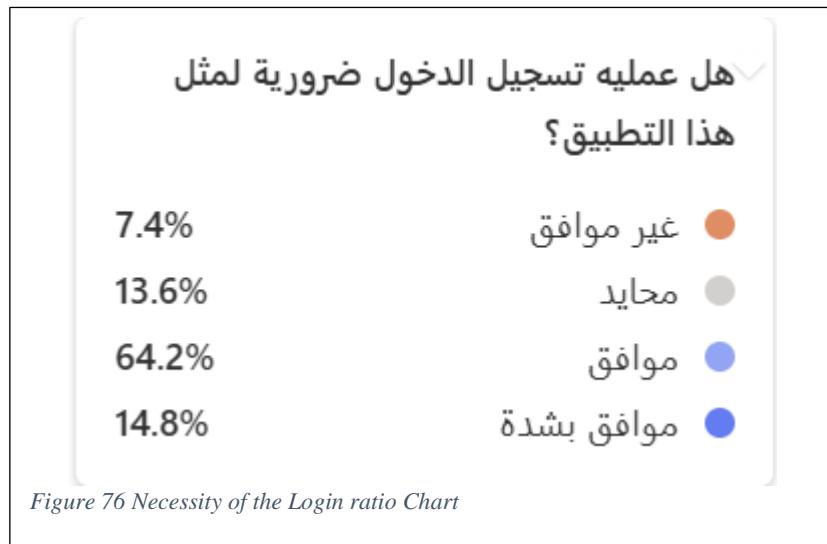


The Ratio Chart in Figure 75 Presents the results of a survey question evaluating whether users found the Muad application visually appealing and suitable as an educational app. Out of 81 respondents, the majority had a positive response.

A significant 13.6% of users strongly agreed that the application meets visual and design expectations for an educational platform, while 81.5% agreed with the statement. This shows that the application's design resonates well with most users, making it engaging and appropriate for educational purposes.

Overall, the data demonstrates that the majority of users (over 95.1%) were satisfied with the application's design, affirming that the visual elements align with the expectations for educational related application.

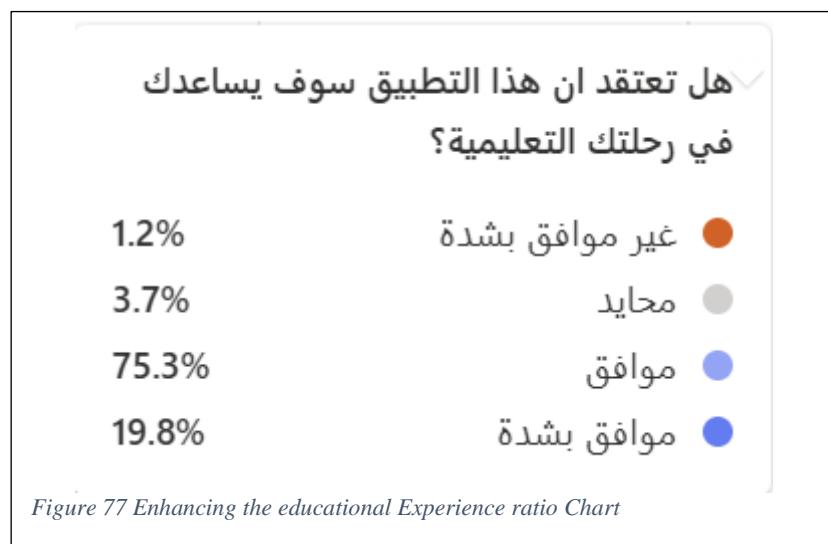
6.3.2 Necessity of the Login Process



The Ratio Chart in Figure 76 illustrates the responses to a survey question asking whether users found the login process necessary for this type of application. The results show that 14.8% of respondents strongly agreed and 64.2% agreed that the login process was essential, suggesting that the majority of users appreciate the value of having secure and personalized experience in the application.

Overall, the results suggest that most users (over 79%) support the inclusion of a login feature, appreciating the potential for security, personalized content, or saved preferences. However, a smaller percentage of users feel that the login process might not be required, hinting at an opportunity to explore easier access options or guest modes in future updates.

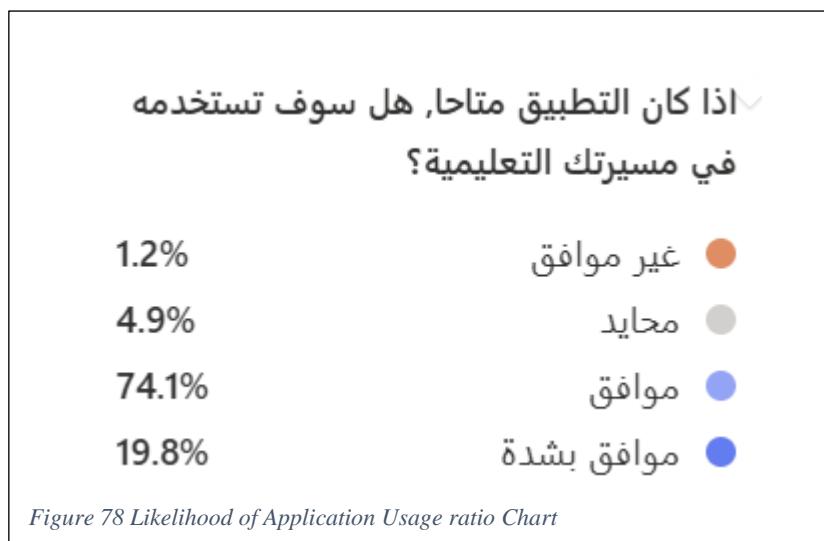
6.3.3 Enhancing the educational Experience



The Ratio Chart Figure 77 Enhancing the educational Experience ratio Chart showcases the responses to a question assessing whether users believe the application adds value to their educational experience. The majority of respondents provided positive feedback, with 19.8% strongly agreeing and 75.3% agreeing that the application enhances their learning experience. This indicates that most users found the app useful and relevant in enriching their educational activities.

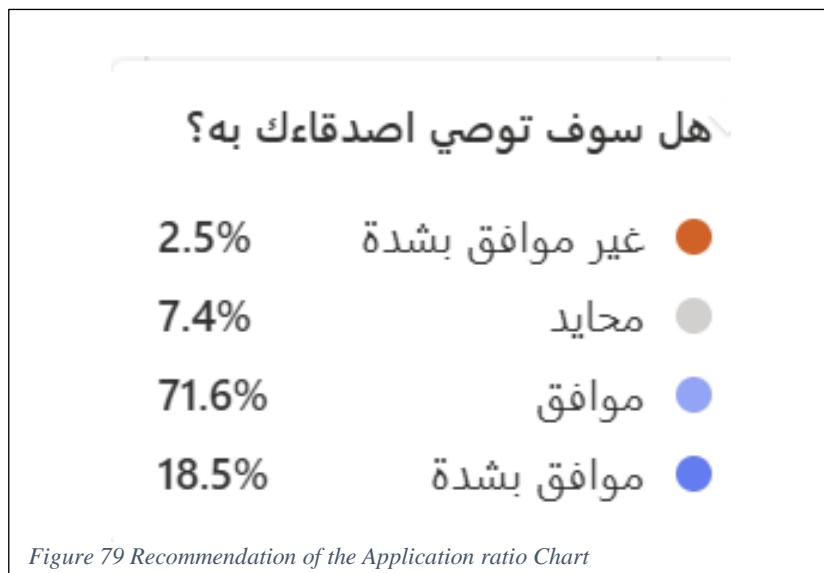
Overall, the feedback highlights that most users appreciate the value provided by the application, reinforcing its role in enhancing educational engagement and suggesting that the app effectively meets user expectations for learning-related experiences.

6.3.4 The Possibility of Using the App during Education Planning or During Educational Experiments



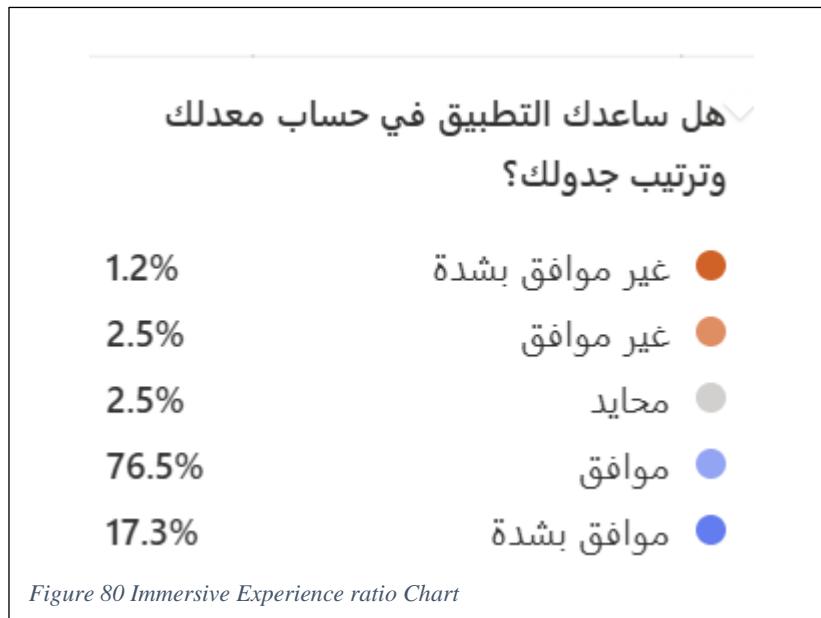
The Ratio Chart Figure 78 illustrates the responses to a question asking users if they would use the application while planning or during their studies. The majority of users provided positive feedback, with 19.8% strongly agreeing and 74.1% agreeing that they would find the application helpful for academic planning. This suggests that most respondents view the application as a valuable tool to enhance their study planning and educational experiences. Overall, the data reflects a strong interest in using the application during their academic journey, highlighting its potential to support users in both the planning phase and throughout their studies. The overwhelmingly positive responses emphasize the app's value in providing useful features and information that align with students' expectations.

6.3.5 Recommendation of the Application to Others



The Ratio Chart **Figure 79** Recommendation of the Application Pie Chart presents the responses to a survey question asking whether users would recommend the application to others. The results show a strong positive response, with 18.5% of respondents strongly agreeing and 71.6% agreeing that they would recommend the application. This highlights the users' satisfaction with the app and their confidence in sharing it with others. Overall, the results indicate high user satisfaction, with the majority willing to recommend the application, suggesting it meets their expectations and offers valuable features for educational related experiences.

6.3.6 Experiment with calculating the rate and organizing the Schedule



The Ratio Chart Figure 81 Immersive Experience ratio Chart illustrates the responses to a survey question asking whether the GPA calculation and schedule management features helped users effectively plan their academic activities. The majority of respondents gave positive feedback, with 17.3% strongly agreeing and 76.5% agreeing that these features enhanced their academic planning experience. This indicates that the GPA calculator and schedule management tools successfully engaged users, providing them with practical and efficient solutions for managing their studies.

Overall, the feedback reflects a strong appreciation for these features, with most users acknowledging their effectiveness in improving academic organization and performance. This suggests that the GPA calculation and schedule management tools are valuable components of the application, contributing positively to the user experience.

6.3.7 Potential Improvements for the Schedule Management Experience

The feedback from participants highlights several areas for Schedule Management Experience. One of the key suggestions emphasized the importance of integrating AI capabilities to enhance ease of use and save students' time. Additionally, respondents recommended adding the name of the instructor to the schedule for better organization. Other significant recommendations focused on improving the user experience by refining icon designs, such as introducing a night mode, supporting multiple languages to cater to a broader audience, allowing users to customize the table color to suit their preferences, and providing the ability to share schedules with others for collaboration and coordination. These suggestions highlight potential enhancements that could make the schedule management feature more efficient and user-friendly.

6.3.8 Identifying Unnecessary or Missing Sections in the Application

Users offered insights into features they found unnecessary or areas where additional functionality could improve the application. A few respondents felt that the login process was not essential unless it allowed users to save favorite points or provided other personalized features, making the effort of logging in more worthwhile. Additionally, it was suggested to add the course type (practical/theoretical) to make it easier for students to organize their schedules and understand the nature of their courses. Some users also recommended providing a dashboard to display the daily schedule clearly and linking the schedule with the university's registration system for automatic updates. Additionally, it was recommended to include a dedicated interface for writing notes, enhancing the user experience and helping students manage their tasks more efficiently.

6.4 Usability Test

In this section, the results of the Usability Test conducted on the "Muad" app, designed for university students, will be reviewed. The goal of the test was to assess the ease of use of the app and its effectiveness in helping users complete various tasks such as logging in, calculating GPA, creating study schedules, tracking absences, and communicating with ambassadors.

The test was conducted with a group of participants from diverse academic and experiential backgrounds, ranging from beginners to those with experience using apps. Data was collected by observing the participants' performance while completing the tasks, as well as their feedback on the challenges they faced and suggestions for improving the experience using think-aloud usability testing method.

This section aims to provide a comprehensive overview of the app's usability, identify strengths and weaknesses, and offer recommendations for enhancing the user experience in the future.

6.4.1 Participant Details

	Participant	Field of Study	Experience Level	Notes
1	Ebtisam Al Hilali	Software Engineering	Advanced	A student interested in new technologies
2	Eman Aali	History and Archaeology	Beginner	No experience with apps.
3	Lama Al-Harbi	Artificial Intelligence	Advanced	Experienced in using apps.
4	Jana Al-shanbry	Software Engineering	Intermediate	Uses educational apps daily.
5	Azhar Al-Hilali	Islamic Law	Beginner	Prefers simple interfaces.
6	Thuraya Omar	Software Engineering	Intermediate	Has a background in apps.

Table 13 Participant Details

6.4.2 Task Performance Details

Task	Participant 1	Participant 2	Participant 3
Login and Account Creation	No issues and liked the error messages that helped them register quickly.	Had difficulty due to the password length and strict requirements.	Completed the registration easily after reading the password requirements carefully.
GPA Calculation	Completed the task easily within two minutes.	Took 4 minutes and struggled to add courses due to registering 16 credit hours.	Completed the task in two minutes with no issues.
Schedule Creation	Completed easily in 4 minutes.	Took 6 minutes due to difficulty coordinating time between 3-4 hours and was unsure about how to input the times.	Created the schedule in 3 minutes.
Absence Calculation	Completed easily and liked the color alerts.	Successfully completed but noted that the "Edit" button was unclear.	Completed easily in under 2 minutes.
Ambassadors	Completed easily within one minute.	Easily completed and tried searching for the right ambassador.	Took one minute, helped by searching and filtering by specialty.

Table 14 Task Performance Details 1

Task	Participant 4	Participant 5	Participant 6
Login and Account Creation	no issues and created an account in under two minutes.	Took longer due to repeated attempts with the password.	Registered and logged in easily.
GPA Calculation	Forgot to add course names but was alerted by the error messages.	Took an additional 3 minutes due to confusion with entering credit hours.	Completed the task easily within two minutes.
Schedule Creation	Entered courses correctly but made an error with the time.	Took 7 minutes due to time coordination and not understanding how to input it correctly.	Completed in under 4 minutes.
Absence Calculation	Needed help to understand the color alerts.	Took 4 minutes due to trial and error, experimenting with the first alert and penalty.	Completed easily within 2 minutes.
Ambassadors	Completed easily, recommended future searches for a professor and contacting via email.	Had some difficulty understanding how to communicate via Twitter.	Completed easily in under one minute.

Table 15 Task Performance Details 2

In the Task Performance Details Table 1 and 2 for the Login and Account Creation task, the error messages were clear and helped participants quickly understand the mistakes, improving their experience. In the Cumulative GPA Calculation task, participants noted that adding course names for multiple courses takes longer depending on the registered credit hours. In the Create Study Schedule task, participants expressed a desire for automatic suggestions for days and times, which could assist new users, along with clearer instructions for coordinating time between hours. In the Track Absences task, participants suggested improving the clarity of the "Edit" button and adding explanations for the color-coded alerts. Finally, in the Communicate with Ambassadors task, customization was effective in helping participants select specialties and search efficiently.

General Usability Test Results

Interface Design:

- **Positives:** The user interface is simple and clear.
- **Common Issues:** Some buttons were unclear to certain users, such as the "Edit" button in the Absence Tracking section.
- **Suggestions:** Improve button clarity and provide better explanations of their functions within the app.

Color-coded Alerts:

- **Positives:** The alerts were visually appealing and helped users understand the status.
- **Common Issues:** Some users needed clarification on the meaning of the alerts.
- **Suggestions:** Add descriptive lines explaining the meanings of the colors, such as (Warning/Disallowed).

Ease of Use:

- **Positives:** Most tasks were clear and easy for the majority of participants.
- **Common Issues:** Tasks requiring the input of complex data were confusing for some, such as coordinating the different time slots (3-4 hours).
- **Suggestions:** Add guidance tips for users while entering data.

6.5 Future Enhancements

Based on user feedback and observations, several potential enhancements have been identified to improve the overall performance, usability, and value of the application. These suggestions focus on optimizing the technical aspects, improving the user experience, and expanding the application's reach through strategic partnerships and platform availability. Implementing these enhancements would help ensure the application continues to meet the evolving needs of users and maintains relevance within the educational sector.

Code Optimization: Improving the efficiency of the application's code can enhance performance, ensuring faster load times, smoother transitions, and reduced resource consumption. This will provide users with a seamless and responsive experience.

Chatbot Integration: Adding a chatbot feature can provide students with instant support, addressing their inquiries in real-time. The chatbot can assist students with academic planning, answering questions related to courses, schedules, and GPA calculations, and offering personalized recommendations, further enhancing the overall user experience.

Collaborations with Universities: Partnering with universities in Makkah or beyond will create opportunities to enrich the content and services offered by the application. Collaborations with educational institutions can attract more students and provide added value, enhancing the overall academic experience for users.

Multilingual Support: Offering the application in multiple languages will cater to a diverse audience, ensuring students from different backgrounds can easily use the app. Expanding language support will improve accessibility and make the application more inclusive for international students.

iOS App Store Availability: Releasing the application on the iOS App Store will broaden its availability, allowing users with Apple devices to benefit from its features. This will increase the user base and ensure compatibility with both Android and iOS platforms.

6.6 Conclusion

In conclusion, the feedback gathered from users highlights the potential of the application to significantly enhance students' academic experiences. By integrating AI-powered features such as GPA calculation and schedule management, the app provides users with efficient tools for academic planning and organization. The suggestions for improvements, such as incorporating a chatbot, supporting multiple languages, and adding useful features like course type indicators and the ability to share schedules, further emphasize the app's potential to meet the evolving needs of its users. Collaborations with universities and enhancing accessibility through multilingual support will ensure that the application remains a valuable resource for students globally. Implementing these improvements will not only optimize the app's performance and user satisfaction but also strengthen its relevance in the educational sector.

7 CHAPTER 7 Conclusion and Future Work

7.1 Introduction

The "Muad" application is an innovative tool designed to simplify students' academic lives by providing a range of features that help them efficiently manage their schedules and academic responsibilities. The app offers functionalities such as attendance tracking with color-coded alerts to notify users when they approach the maximum allowed absence limit. It also enables users to calculate their cumulative and semester GPAs with ease. Additionally, the app facilitates communication with academic ambassadors via social media platforms like Twitter and LinkedIn. Furthermore, students can add and manage their class schedules using smart tools, enhancing their overall academic experience.

7.2 Future Work

Following implementation, user suggestions, and ongoing research, several future enhancements have been identified to improve the functionality and scope of the application:

1. AI Training and Full Integration: Enhancing AI models and seamlessly integrating them with both the front-end and back-end of the application to provide a cohesive user experience.
2. Code Optimization: Enhancing the efficiency of the application's code will improve performance, reduce load times, and increase responsiveness.
3. Student Clubs: To encourage student involvement and development, we are adding a section for student clubs, showcasing activities, events, and available courses.
4. Academic Consultations: Offering academic consultations from experts to guide students in making informed academic decisions.

5. To-Do List: Including a to-do list feature to help students organize their daily tasks, both academic and personal.
6. Multilingual Support: Expanding the app's language options to cater to a diverse audience of students from various backgrounds.
7. iOS Availability: Launching the app on the iOS App Store to reach a broader audience and ensure compatibility with both Android and iOS devices.
8. Partnerships with Educational Institutions: Collaborating with universities and schools to expand the app's content and services, creating additional value for users.

These improvements will significantly enhance the app's performance, user experience, and functionality, ensuring "Muad" remains an essential tool for students seeking to optimize their academic lives.

8 Conclusion

The conclusion of this report encapsulates the key insights, accomplishments, and findings from each chapter, demonstrating the development and evaluation process of the Muad application.

Chapter 1 provided a comprehensive introduction to the project, identifying the problem domain and defining the proposed system's features. It outlined the aims, objectives, and methodologies adopted to deliver a user-focused academic management application. The core features such as attendance tracking, communication with ambassadors, and GPA calculation were designed to streamline academic processes and enhance the user experience.

Chapter 2 discussed the background and existing works relevant to the project. It examined similar applications and identified their limitations, establishing the basis for the innovations integrated into Muad. This chapter justified the need for the application by addressing gaps in current academic tools and proposing novel features.

Chapter 3 focused on the software requirements specifications, detailing the functional and non-functional requirements. It defined the project scope, user scenarios, and product perspective, offering an in-depth look into system interfaces, operating environments, and user classes. The design constraints ensured that the application was optimized for usability, flexibility, and scalability.

Chapter 4 explored the system's architecture and design. It provided visual representations through use case diagrams, context diagrams, activity diagrams, and sequence diagrams, illustrating the user interactions with the system. This chapter also described the structural design, interfaces, and data stores used to implement the application, emphasizing the technological components supporting the user experience.

Chapter 5 detailed the implementation and validation processes, discussing the technology stack, collaboration tools, and coding practices employed to develop the system. This chapter reflected on the practical steps undertaken to transform the design into a functional application, ensuring seamless integration of features and system reliability.

Chapter 6 focused on testing and evaluation, highlighting the application's performance and user feedback. Through structured surveys and analysis, the chapter explored the functionality, usability, and effectiveness of the features. User feedback provided insights into potential enhancements and identified areas for improvement, guiding future development efforts.

In summary, the "Muad" project successfully addressed user needs by offering a comprehensive set of features, including absence tracking with color-coded alerts, cumulative and semester GPA calculation, and communication with academic ambassadors. The project prioritized user-centered design and delivered enhanced academic experience for students through tailored academic functionalities.

The report provided a detailed overview of the project lifecycle, from conceptualization and design to implementation and evaluation. Feedback and results highlighted the importance and effectiveness of the application while identifying opportunities for future improvements, laying a solid foundation for further development in upcoming versions

9 References

MyStudyLife A free app designed to organize students' academic lives, providing tools for managing schedules, tasks, and exams. Retrieved from <https://mystudylife.com/>

GradeCalculator.ai An AI-powered tool for tracking academic performance, offering precise and user-friendly features for calculating and understanding grades across various educational systems. Retrieved from <https://gradecalculator.ai/about>

Integration of Artificial Intelligence Performance Prediction and Learning Analytics A scholarly article discussing the use of AI to enhance student learning in online engineering courses. Retrieved from <https://www.ratemyprofessors.com/>

UML Use Case Diagrams An Introduction: A detailed guide on Use Case Diagrams in UML. Retrieved from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-use-case-diagram/>

Use Case and Scenario Diagrams: A guide on creating Scenario Diagrams using UML. Retrieved from <https://www.uml-diagrams.org/use-case-scenario-diagram.html>

UML Sequence Diagram Tutorial: Learn how to create Sequence Diagrams in UML. Retrieved from <https://creately.com/diagram/example/inxmrzy5/uml-sequence-diagram-for-creating-a-user-account>

UML Activity Diagrams A Complete Guide: A comprehensive guide to Activity Diagrams in UML. Retrieved from <https://www.lucidchart.com/pages/uml-activity-diagram>

UML Class Diagram A Complete Guide: A full explanation of UML Class Diagrams and how to use them. Retrieved from <https://www.lucidchart.com/pages/uml-class-diagram>

What is System Architecture : A guide on system architecture and how to represent it using diagrams. Retrieved from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/system-architecture-diagram/>

Flutter Documentation Official documentation to learn Flutter from basics to advanced topics. Retrieved from <https://flutter.dev/docs>

Dart Documentation: Learn the fundamentals of Dart programming language, which is used for Flutter. Retrieved from <https://dart.dev/guides>

Flutter by Example: Practical examples to learn Flutter step by step. Retrieved from <https://flutterbyexample.com/>

freeCodeCamp - Flutter Tutorial: A comprehensive course to learn Flutter from beginner to advanced. Retrieved from <https://www.freecodecamp.org/news/learn-flutter-beginner-to-advanced/>

The Net Ninja Flutter Tutorial: Free video course on YouTube to learn Flutter. Retrieved from https://www.youtube.com/playlist?list=PL4cUxeGkcC9gQ7xu9ByiJ5Q8hCH9f0y_3

Udemy - Flutter Bootcamp: Complete course on Udemy to learn Flutter and Dart. Retrieved from <https://www.udemy.com/course/flutter-bootcamp-with-dart/>

Flutter Community: A community for knowledge sharing and problem-solving in Flutter. Retrieved from <https://fluttercommunity.dev/>

Flutter Awesome: A collection of libraries and tools for Flutter projects. Retrieved from <https://flutterawesome.com/>

YouTube - Video 1: Flutter tutorial on YouTube. Retrieved from <https://youtu.be/4emxjxonNRI?si=VM8Mm1EOJVA9nUVR>

YouTube - Video 2: Flutter tutorial on YouTube. Retrieved from https://youtu.be/pCK6prSq8aw?si=eL87-bN-WkZ7Fe_T

YouTube - Video 3: Flutter development tutorial. Retrieved from <https://www.youtube.com/watch?v=xQ2qiM0Zpr8>

YouTube - Video 4: Flutter advanced concepts. Retrieved from
<https://www.youtube.com/watch?v=6XrL5jXmTwM>

pub.dev: Official Flutter package repository. Retrieved from <https://pub.dev/>

Flutter Documentation: Official Flutter documentation. Retrieved from
<https://docs.flutter.dev/>

GitHub: A platform for code sharing and collaboration. Retrieved from <https://github.com/>

CSS Gradient: CSS gradient generator for designing gradients. Retrieved from
<https://cssgradient.io/>

Flutter Awesome: A curated list of libraries and tools for Flutter. Retrieved from
<https://flutterawesome.com/>

YouTube - Playlist 1: Flutter tutorial playlist. Retrieved from
<https://youtube.com/playlist?list=PLeWmXrh004782w91CeXQ4bK8ivREDau-x&si=Ummn4aPCjgzFAivS>

YouTube - Playlist 2: Another Flutter tutorial playlist. Retrieved from
https://youtube.com/playlist?list=PLw6Y5u47CYq47oDw63bMqkq06fjuoK_GJ&si=0ViVT-Dq1CwUNAwjC

YouTube - Playlist 3: Flutter tutorial series. Retrieved from
https://youtube.com/playlist?list=PLjOjILrTIBG1wlcm65P7lGZsNe4a2q4fq&si=i7p_1W-zPyp4oQ5I

The PHP Framework for Web Artisans: Official documentation for Laravel.
Retrieved from <https://laravel.com/docs/11>

How to Build a Real-Time Chat App with Laravel Reverb: A guide on creating real-time chat applications. Retrieved from <https://www.freecodecamp.org/news/laravel-reverb-realtime-chat-app/>

Understanding Laravel — Build a Simple TODO App: Step-by-step tutorial to understand Laravel. Retrieved from <https://ibrajix.medium.com/understanding-laravel-build-a-simple-todo-app-f15a8a677f0b>

Table Plus User Guide: Official guide to using Table Plus.
Retrieved from <https://docs.tableplus.com/getting-started>

Getting Started with Table Plus: A beginner's guide to Table Plus.
Retrieved from <https://medium.com/tableplus/getting-started-with-tableplus-1ad69e3a635d>

Postman User Guide: Comprehensive guide to Postman features.
Retrieved from <https://learning.postman.com/docs/introduction/overview/>

Postman Beginner's Course - API Testing: Video course on YouTube.
Retrieved from <https://www.youtube.com/watch?v=VyxIQ2ZXw4>

MySQL User Guide: Official MySQL documentation.
Retrieved from <https://dev.mysql.com/doc/>

SQL Course for Beginners [Full Course]: Beginner's SQL course on YouTube.
Retrieved from https://www.youtube.com/watch?v=7S_tz1z_5bA

Ali Majrashi Course: A complete Arabic MySQL course.
Retrieved from <https://3alam.pro/ali-majrashi/series/mysql>

PHP MySQL Full Course in Arabic: An Arabic course for PHP and MySQL.
Retrieved from <https://wrshaah.com/video/KO87boBerDk>

25 VS Code Productivity Tips and Speed Hacks: YouTube video on improving VS Code productivity. Retrieved from <https://www.youtube.com/watch?v=ifTF3ags0XI>

GitHub User Guide: Official GitHub documentation.
Retrieved from <https://docs.github.com/en/get-started/start-your-journey>

Git and GitHub for Beginners - Crash Course: YouTube course for Git and GitHub.

Retrieved from <https://www.youtube.com/watch?v=RGOj5yH7evk>

Git Documentation: Official documentation for Git.

Retrieved from <https://git-scm.com/doc>

Learn Git in 15 Minutes: Quick Git tutorial on YouTube.

Retrieved from <https://www.youtube.com/watch?v=USjZcfj8yxE>

Git and GitHub Documentation: Blog tutorial on Git and GitHub.

Retrieved from <https://blog.codershiyar.com/2023/07/git-github.html>

Termius Blogs: Tutorials for Termius usage.

Retrieved from <https://support.termius.com/hc/en-us>

What is XAMPP: Overview of XAMPP.

Retrieved from <https://www.apachefriends.org/index.html>

Introduction to XAMPP: Video introduction to XAMPP on YouTube.

Retrieved from <https://www.youtube.com/watch?v=h6DEDm7C37A>

Trello Support: Official Trello support documentation.

Retrieved from <https://support.atlassian.com/trello/>

Getting Started with Trello (Demo): Trello tutorial on YouTube.

Retrieved from <https://www.youtube.com/watch?v=xky48zyL9iA>

VS Code and Git Guide: GitHub repository for using VS Code with Git.

Retrieved from <https://github.com/MjeedBakr/vscode-and-git-guide>

Get Started with Jira: Official Jira signup page.

Retrieved from <https://www.atlassian.com/try/cloud/signup?bundle=jira-software>

Laravel OTP Login Tutorial: Tutorial on implementing OTP login in Laravel.

Retrieved from <https://dev.to/techtoolindia/laravel-otp-login-tutorial-5ek1>

DB_CONNECTION: Step-by-step guide to connecting MySQL in Laravel.

Retrieved from <https://igotocode.com/ar/connecting-mysql-in-laravel-step-by-step-guide>

Laravel RESTful API: Comprehensive guide to building RESTful APIs in Laravel.

Retrieved from <https://igotocode.com/ar/building-laravel-restful-api-application-comprehensive-guide>

Laravel WebSocket: Tutorial for integrating Laravel WebSocket.

Retrieved from <https://igotocode.com/ar/integrating-laravel-websocket-with-database-real-time-data-management>

Email Verification in Laravel: Guide for creating an email verification system in Laravel.

Retrieved from <https://codeteklab.com/articles/laravel-11-create-login-register-email-verification-system>

Routing in Laravel: Documentation on Laravel routing.

Retrieved from <https://wiki.hsoub.com/Laravel/routing>

Creating a Model in Laravel: Guide on how to create a model in Laravel.

Retrieved from

<https://academy.hsoub.com/programming/php/laravel/%D9%83%D9%8A%D9%81-%D8%AA%D9%86%D8%B4%D8%A6-%D9%86%D9%85%D9%88%D8%B0%D8%AC%D8%A7-model-%D9%81%D9%8A-laravel-r297/>

10 Appendix

Appendix 1 :

The screenshot shows a Google Sheets document titled 'plan' with multiple tabs. The active tab, 'Sheet1', displays a complex project timeline from January 1st to April 1st. The timeline is divided into several phases: 'Initial setup', 'Database creation', 'Data entry', 'Analysis', 'Testing', 'Deployment', and 'Maintenance'. Each phase contains numerous tasks, many of which are color-coded (green, yellow, red) and some have checkmarks indicating completion. The 'Analysis' phase, spanning from January 15th to February 15th, includes tasks like 'Identify key metrics', 'Analyze user behavior', 'Create reports', and 'Present findings'. The 'Testing' phase, from February 15th to March 15th, includes tasks like 'Unit testing', 'Integration testing', 'System testing', and 'User acceptance testing'. The 'Deployment' phase, from March 15th to April 1st, includes tasks like 'Deploy to test environment', 'Monitor system performance', and 'Collect user feedback'. The 'Maintenance' phase, from April 1st to April 15th, includes tasks like 'Handle support requests', 'Update documentation', and 'Plan for future releases'.

Figure 81 Screenshot of Google Sheets

Appendix 2:

The screenshot shows a Microsoft Excel spreadsheet titled 'plan' with the formula bar showing '1/6/1900'. The active sheet, 'Sheet1', contains a table titled 'Minutes of meeting' with columns for 'Arrange tasks', 'Status', 'Member', 'Week', 'Date from', 'Date to', 'Features to do', 'Details', 'Arrange tasks', and 'Status'. The table has 11 rows, with rows 2 and 3 having 'Done' status and rows 4 and 5 having 'In Progress' status. Row 6 has 'Reham' in the Member column. Rows 7 through 11 all have 'Done' in the Status column. The 'Features to do' column contains tasks like 'Search for a way to train the machine.', 'We started using Google Colab.', 'Way to create a schedule', and 'Interface to choose how to arrange the table'. The 'Details' column contains URLs and notes. The 'Arrange tasks' and 'Status' columns are empty. The 'File' tab is selected at the top. A sidebar on the left shows 'Document Recovery' with a link to a shared file.

Figure 82 Screenshot of Microsoft Excel

Appendix 3:

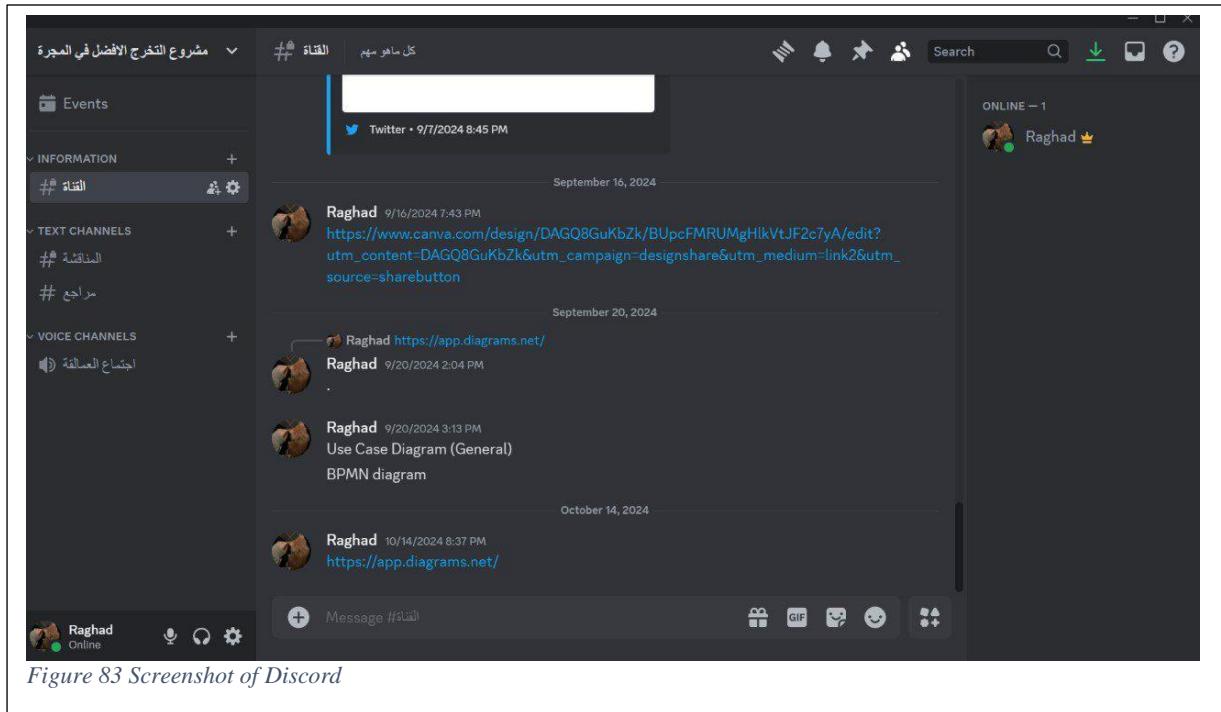


Figure 83 Screenshot of Discord

Appendix 4:

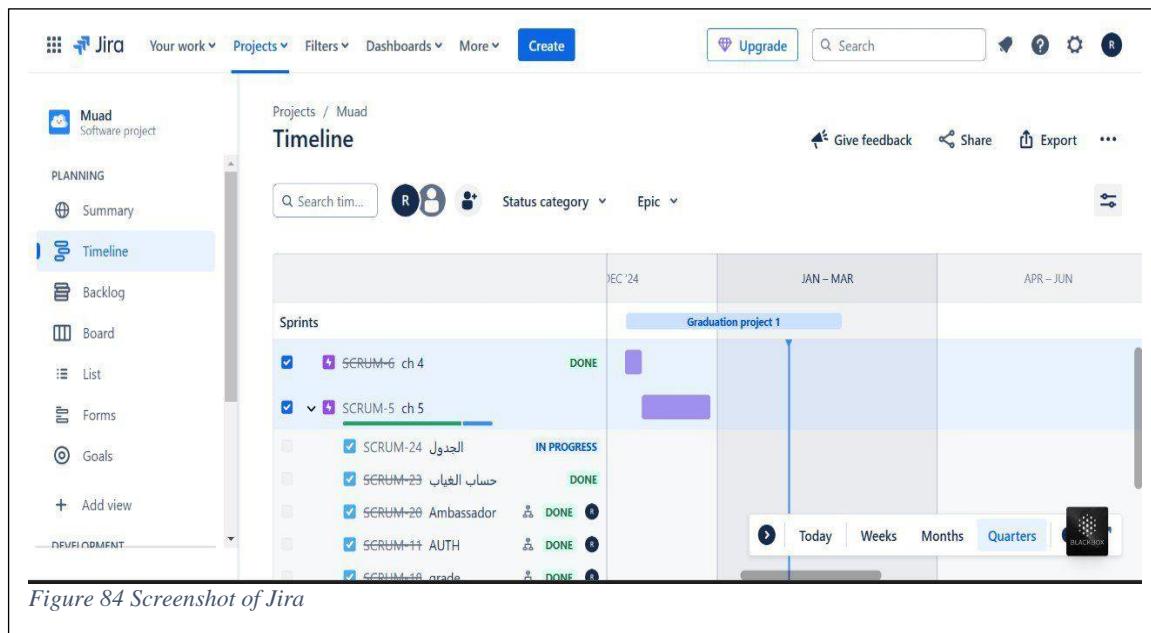


Figure 84 Screenshot of Jira

Appendix 5:

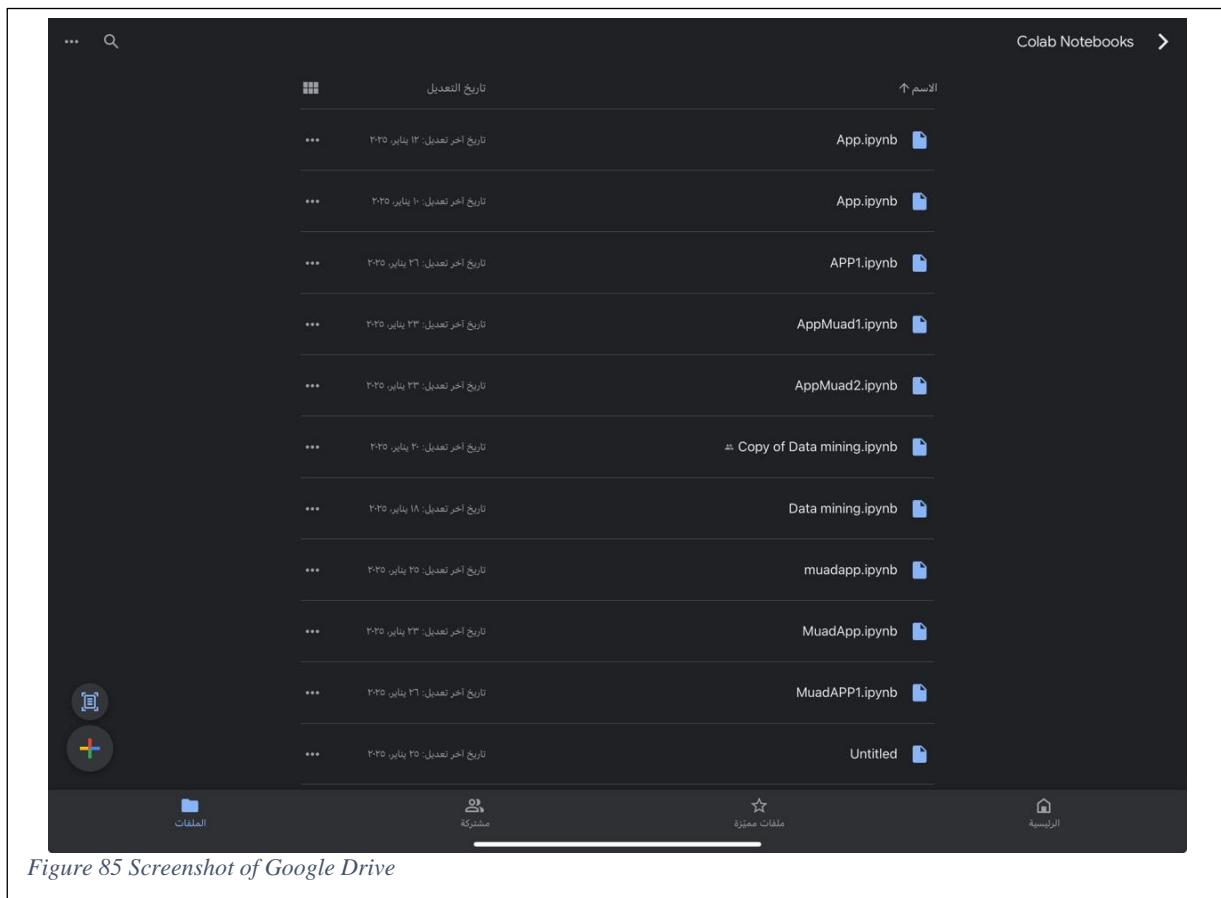


Figure 85 Screenshot of Google Drive

Appendix 5:

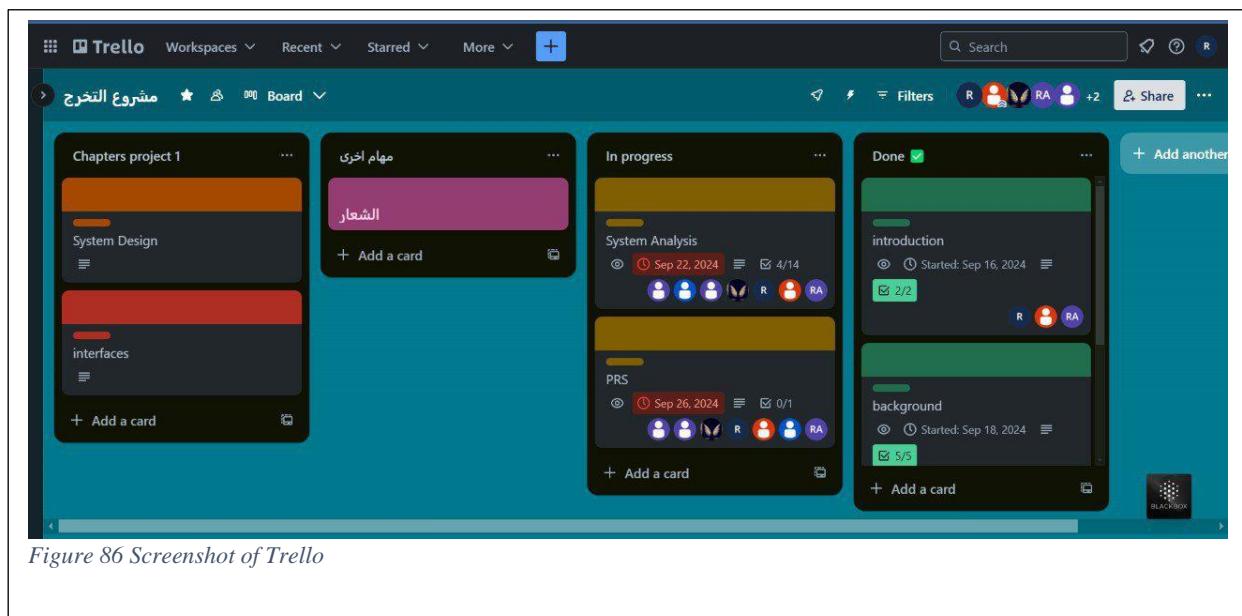


Figure 86 Screenshot of Trello

Appendix 7:

The screenshot shows the GitHub repository page for 'Muad-App'. The repository is public and has 1 branch and 0 tags. The main commit is by 'JooooJo565' titled 'first commit' at 4497e9d · 43 minutes ago. The commit message is '1 Commit'. The commit history includes entries for 'android', 'assets', 'ios', 'lib', 'linux', 'macos', 'test', 'web', and 'windows', all labeled as 'first commit' at 43 minutes ago. A file named '.gitignore' is also listed as having a 'first commit' at 43 minutes ago. The repository has 0 stars, 1 watching, and 0 forks. It has no releases or packages published.

Figure 87 Screenshot of GitHub 1

The screenshot shows the GitHub repository page for 'muad-backend'. The repository is private and has 1 branch and 0 tags. The main commit is by 'reem0l' titled 'Inconsistency in Github action names (#6478)' at b98a66d · 5 days ago. The commit message is '7,146 Commits'. The commit history includes entries for '.github/workflows', 'app', 'bootstrap', 'config', 'database', 'muad-backend', 'public', and 'resources', all with various commit messages and dates ranging from 5 days ago to last year. The repository has 1 star, 1 watching, and 0 forks. It has no releases or packages published.

Figure 88 Screenshot of GitHub 2