

NAME: MUHAMMAD SOOMAR NOHRI

ROLL NUMBER: 2K24/IT/108

COURSE NAME: MOBILE APPLICATION DEVELOPMENT

ASSIGNMENT TITLE: StudentOS APP REPORT

SEMESTER: 3rd Year 5th Semester

DEPARTMENT: INFORMATION TECHNOLOGY

UNIVERSITY: UNIVERSITY OF SINDH

Contents

CHAPTER 1 – INTRODUCTION 5
1.1 Problem Background 5
1.2 Need of the App 5
1.3 Objectives 6
1.4 Scope 7
CHAPTER 2 – LITERATURE REVIEW 9
2.1 Similar Apps Analysis 9
2.1.1 My Study Life – School Planner 9
2.1.2 Todoist 10
2.1.3 Egenda – Homework Manager 11
2.2 Existing Problems 12
2.3 Opportunity for Improvement 13
CHAPTER 3 – SYSTEM ARCHITECTURE AND UI DESIGN 14
3.1 App Architecture Diagram 14
3.2 Database Design 15
3.3 Database Flow 17
3.4 Screen Navigation Diagram 17
3.5 Use Case Diagram 18
3.6 User Interface Design 19
CHAPTER 4 – AGILE METHODOLOGY 23
4.1 Introduction 23
4.2 Agile Process in StudentOS 23
CHAPTER 5 – PLANNING 24
5.1 Introduction 24
5.2 Requirement Analysis 24
5.3 Project Modules 25
5.4 Development Tools 25
5.5 Outcome of Planning Phase 25

CHAPTER 6 – UI/UX DESIGN 25
6.1 Introduction 25
6.2 User Interface Design 25
6.3 Navigation Design 26
6.4 Dashboard Design 26
6.5 Forms Design 26
6.6 List Design 26
6.7 Detail Pages 27
6.8 Outcome of UI/UX Design Phase 27
CHAPTER 7 – DEVELOPMENT SPRINT 1 27
7.1 Introduction 27
7.2 Database Implementation 27
7.3 Task Management Module 28
7.4 Notes Management Module 28
7.5 Outcome of Sprint 1 28
CHAPTER 8 – DEVELOPMENT SPRINT 2 28
8.1 Introduction 28
8.2 Attendance Management Module 28
8.3 GPA Calculator Module 29
8.4 Profile Management Module 29
8.5 Dashboard Integration 29
8.6 Outcome of Sprint 2 29
CHAPTER 9 – TESTING 29
9.1 Introduction 29
9.2 Testing Approach 30
9.3 Test Cases and Results 30
9.4 Issues Encountered 30
9.5 Results 30
9.6 Conclusion 31

CHAPTER 10 – CONCLUSION 31
10.1 Learning Outcomes 31
10.2 Challenges Faced 32
10.3 Future Improvements 32

CHAPTER 1 – INTRODUCTION

1.1 Problem Background

In the modern educational environment, students are required to manage multiple academic activities simultaneously. These activities include attending classes, completing assignments, maintaining notes, monitoring attendance, calculating GPA, and keeping track of important academic information. Managing all these activities manually can be difficult, time-consuming, and inefficient.

Many students still rely on traditional methods such as notebooks, paper records, separate mobile applications, or memory to manage their academic tasks. As a result, important information is often misplaced, forgotten, or poorly organized. For example, students may forget assignment deadlines, lose important notes, fail to monitor their attendance percentage, or make mistakes while calculating their GPA.

With the widespread use of smartphones, mobile applications have become an effective solution for managing daily activities. Educational mobile applications provide students with easy access to information anytime and anywhere. However, many available applications focus only on a single function, such as note-taking or attendance tracking, rather than providing a complete academic management solution.

To address these challenges, the StudentOS mobile application has been developed. StudentOS is an Android-based academic management application that combines multiple student-related functionalities into a single platform. It provides facilities for task management, note management, attendance tracking, GPA calculation, and profile management. The application aims to simplify academic organization and improve productivity among students.

By integrating these features into one application, StudentOS reduces the need to use multiple tools and helps students efficiently manage their academic responsibilities.

1.2 Need of the App

The rapid growth of mobile technology has transformed the way students interact with educational resources. Students increasingly depend on smartphones for communication, learning, scheduling, and information management. Therefore, there is a strong need for a dedicated application that can help students manage their academic activities in a structured and organized manner.

One of the major problems faced by students is the lack of a centralized academic management system. Important information such as assignments, notes, attendance records, GPA calculations, and personal academic details are often scattered across different platforms or stored manually. This creates confusion and increases the chances of losing important information.

Students frequently face challenges such as:

- Missing assignment deadlines.
- Forgetting important academic tasks.
- Losing lecture notes and study material.
- Difficulty in monitoring attendance percentages.
- Errors in manual GPA calculations.
- Lack of proper organization of academic information.

StudentOS addresses these challenges by providing an integrated solution that allows students to store, manage, and monitor all important academic information from a single mobile application.

The application improves convenience, saves time, reduces manual effort, and enhances academic productivity. Students can quickly access their information, monitor their progress, and stay organized throughout their academic journey.

1.3 Objectives

The primary objective of StudentOS is to provide a comprehensive mobile platform that assists students in managing their academic activities efficiently and effectively.

The specific objectives of the project are as follows:

General Objective

To develop an Android-based academic management application that enables students to organize, monitor, and manage their academic information through a single platform.

Specific Objectives

1. To develop a task management system that allows students to create, update, view, and delete academic tasks and assignments.
2. To provide a digital notes management system that enables students to store and organize subject-related notes within the application.
3. To develop an attendance tracking system that records attendance information and calculates attendance percentages automatically.
4. To implement a GPA calculator that helps students evaluate their academic performance based on grades and credit hours.
5. To provide a profile management system where students can save and update their personal and academic information.
6. To create a user-friendly interface that is easy to understand and navigate.

7. To improve student productivity through better academic organization and information management.
8. To reduce dependence on traditional manual methods of record keeping.
9. To provide quick access to academic information using mobile technology.
10. To integrate multiple academic management functions into a single application.

1.4 Scope

The scope of the StudentOS application focuses on providing essential academic management functionalities for students through an Android-based mobile platform.

The application is intended primarily for students who wish to organize their academic information efficiently. It allows users to maintain records of tasks, attendance, notes, GPA calculations, and profile information within a single application.

The major functional areas included in the project are:

Home Dashboard

The dashboard provides an overview of important academic information such as attendance percentage, GPA information, and pending tasks.

Task Management Module

This module enables students to create, view, update, and delete academic tasks and assignments. It helps students keep track of deadlines and pending work.

Notes Management Module

The notes module allows students to store subject-related notes digitally. Students can create, update, and manage their notes within the application.

Attendance Management Module

This module records attendance information for different subjects and automatically calculates attendance percentages. Students can monitor their attendance status and identify subjects requiring improvement.

GPA Calculator Module

The GPA calculator enables students to calculate their Grade Point Average based on credit hours and grades. This helps students evaluate their academic performance.

Profile Management Module

The profile module stores student information such as name, department, semester, university, email address, and CGPA.

Database Management

The application uses SQLite Database for local data storage and Shared Preferences for storing profile-related information. This allows data to remain available even after the application is closed.

Limitations of Current Scope

The current version of StudentOS does not include cloud synchronization, online data backup, timetable management, push notifications, or multi-user support. These features may be incorporated in future versions of the application.

Future Scope

Future enhancements may include:

- Cloud database integration.
- User authentication and login system.
- Timetable management module.
- Assignment reminders and notifications.
- Online synchronization across devices.
- Data backup and recovery features.
- Integration with university management systems.
- Advanced analytics and performance tracking.

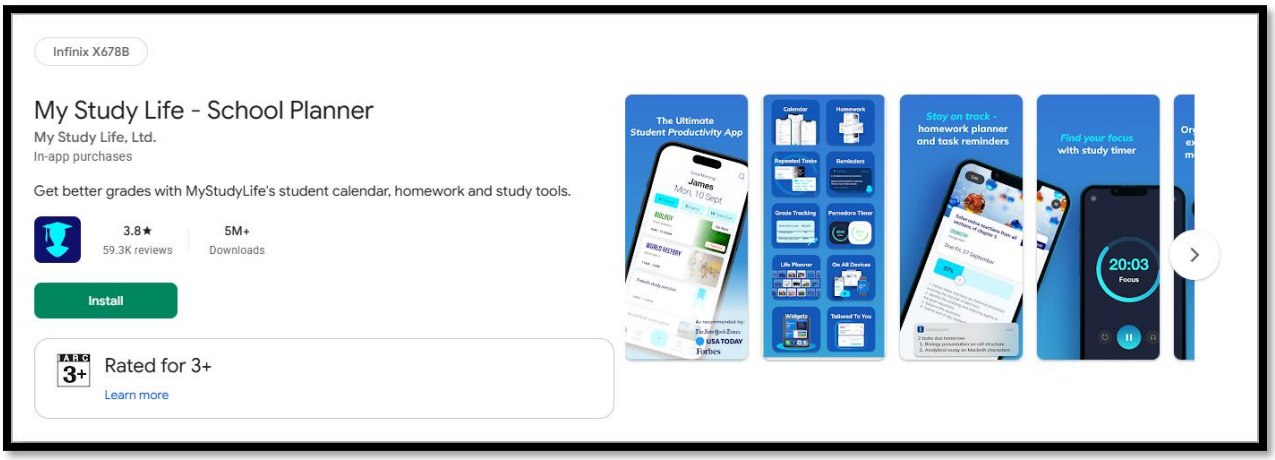
The future scope of StudentOS aims to transform the application into a complete academic management platform capable of supporting a wide range of student needs.

CHAPTER 2 – LITERATURE REVIEW

2.1 Similar Apps Analysis

Before developing StudentOS, several existing student productivity and academic management applications available on the Google Play Store were studied. The purpose of this analysis was to understand the features offered by existing applications, identify their limitations, and determine opportunities for improvement.

2.1.1 My Study Life – School Planner



My Study Life is a popular student planner application available on the Google Play Store. It helps students organize their class schedules, homework, assignments, exams, and study plans. The application provides reminders and scheduling tools that assist students in managing their academic workload efficiently. It also offers timetable management and grade tracking features. Millions of students use this application for academic organization and planning.

Key Features

- Homework and assignment management.
- Exam scheduling and reminders.
- Timetable management.
- Grade tracking.
- Multi-device synchronization.

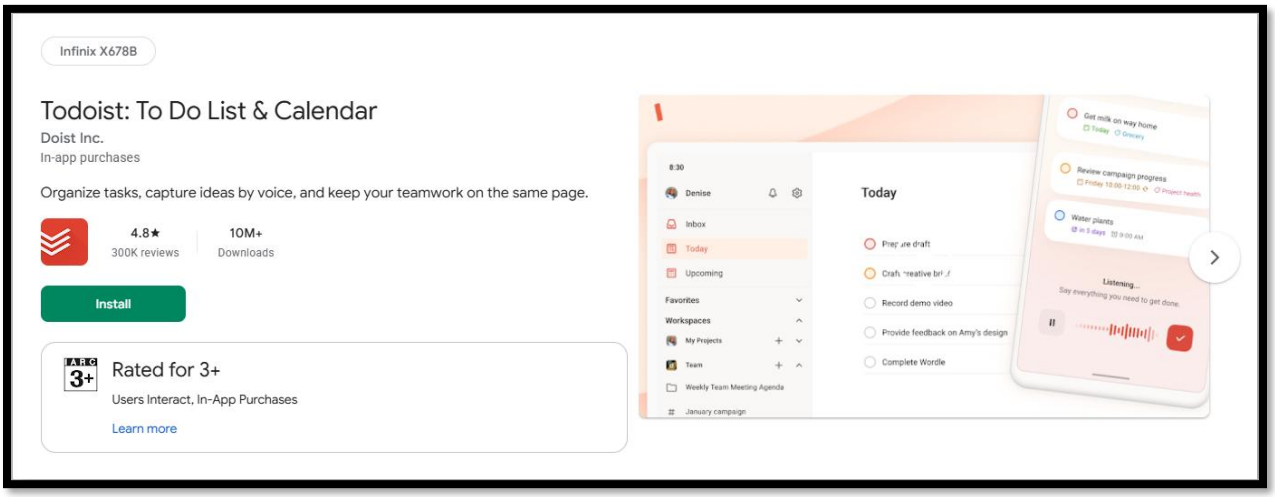
Advantages

- User-friendly interface.
- Effective timetable management.
- Academic planning tools.
- Cloud synchronization support.

Limitations

- Some advanced features require premium access.
- Limited customization options.
- Focuses mainly on planning rather than complete academic management.
- Does not provide integrated attendance tracking and local academic record management.

2.1.2 Todoist



Todoist is a task management and productivity application widely used by students and professionals. It allows users to create tasks, organize projects, set priorities, and receive reminders. The application is highly effective for personal productivity and assignment management. It supports collaboration and synchronization across multiple platforms.

Key Features

- Task management.
- Priority-based task organization.
- Deadline reminders.
- Project categorization.

- Cross-platform synchronization.

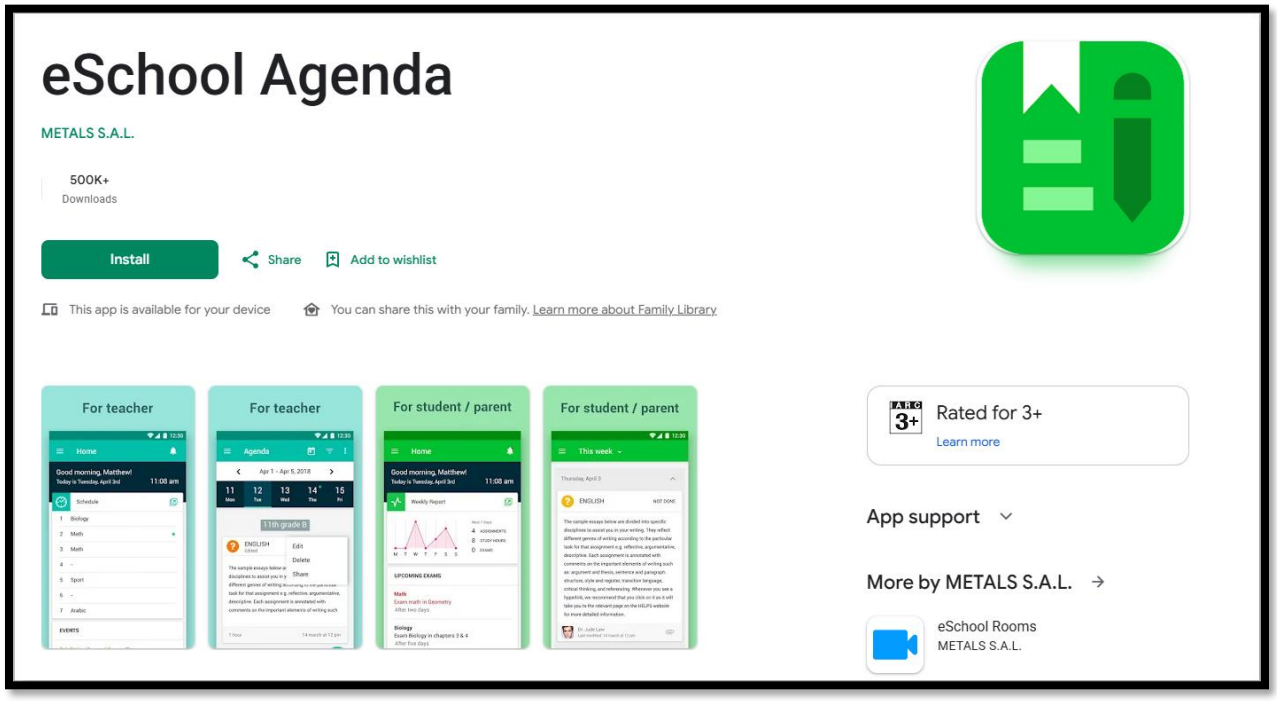
Advantages

- Clean and modern interface.
- Excellent task organization.
- Productivity enhancement tools.
- Easy task scheduling.

Limitations

- Not specifically designed for students.
- No attendance tracking system.
- No GPA calculator.
- No academic note management module.
- Some advanced features require paid subscription.

2.1.3 Egenda – Homework Manager



Egenda is a homework management application designed specifically for students. The application helps students organize assignments, homework tasks, and class-related activities. It provides a structured way

to manage academic work and deadlines. The application was developed with a strong focus on simplifying homework management for students.

Key Features

- Homework management.
- Assignment tracking.
- Class organization.
- Academic scheduling.
- Task reminders.

Advantages

- Student-focused design.
- Easy assignment tracking.
- Simple and intuitive interface.
- Useful homework organization tools.

Limitations

- Limited attendance management functionality.
- No GPA calculator.
- Limited profile management features.
- Lack of integrated academic dashboard.
- Fewer academic management modules compared to comprehensive student systems.

2.2 Existing Problems

The analysis of existing applications revealed several common problems and limitations.

Most applications focus on only one or two academic functions rather than providing a complete academic management solution. For example, some applications specialize in task management while others focus on scheduling or homework tracking. As a result, students often need to install multiple applications to manage different academic activities.

Another issue is that many applications require internet connectivity or cloud synchronization for full functionality. This can be inconvenient for students who need offline access to their academic information.

Several applications also restrict important features behind premium subscriptions, limiting accessibility for students who prefer free solutions. Furthermore, many applications lack attendance tracking, GPA calculation, and profile management features that are important for academic monitoring.

The major problems identified are:

- Lack of an all-in-one academic management solution.
- Limited attendance tracking functionality.
- Absence of GPA calculation features.
- Dependence on multiple applications.
- Premium feature restrictions.
- Limited offline functionality.
- Incomplete academic progress monitoring.

2.3 Opportunity for Improvement

The identified limitations create an opportunity to develop a more comprehensive and student-centered academic management application.

StudentOS aims to address these shortcomings by integrating multiple academic management functions into a single Android application. Unlike existing applications that focus on individual features, StudentOS provides a centralized platform where students can manage tasks, attendance, notes, GPA calculations, and personal academic information.

The proposed system offers several improvements:

- Integrated academic dashboard.
- Task management with CRUD operations.
- Attendance tracking with percentage calculation.
- GPA calculation module.
- Digital note management system.
- Student profile management.
- Offline data storage using SQLite Database.
- Simple and user-friendly interface.
- Free access to all available features.

By combining these features into one platform, StudentOS reduces complexity, improves productivity, and provides students with a complete academic management solution that supports their daily educational activities.

CHAPTER 3 – SYSTEM ARCHITECTURE AND UI DESIGN

3.1 App Architecture Diagram

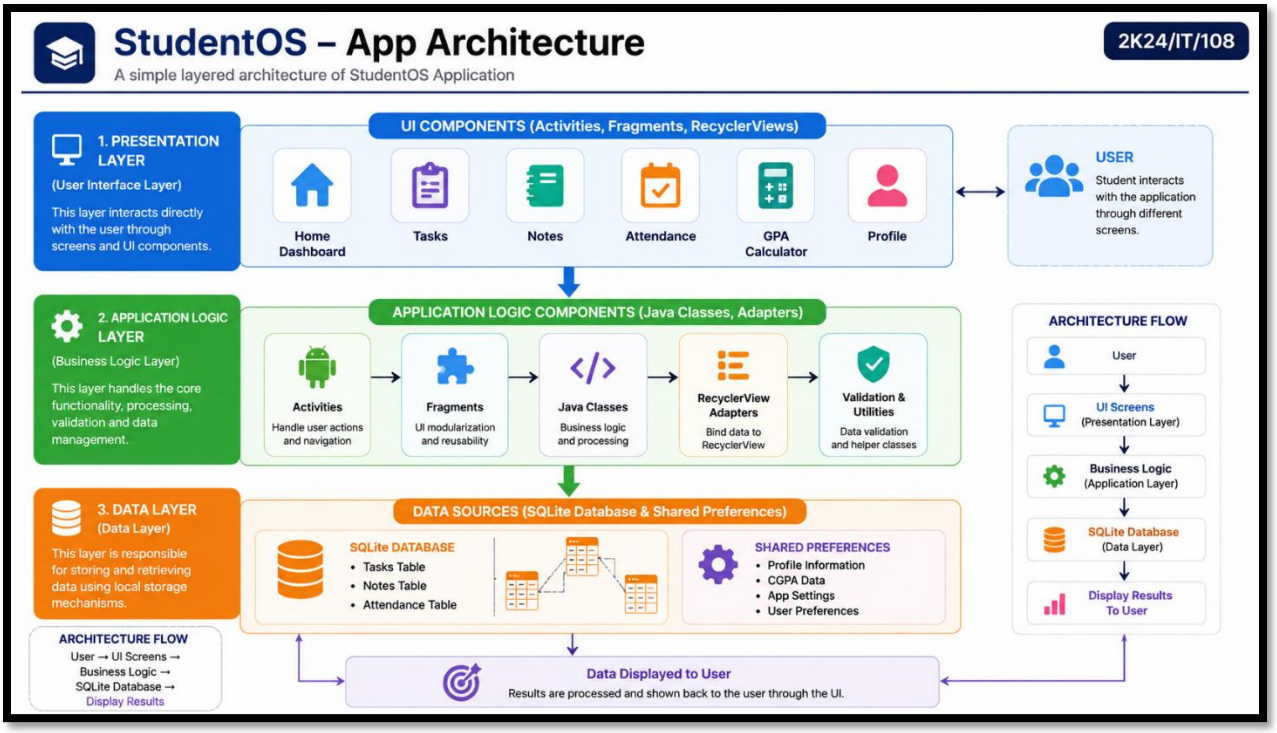
StudentOS follows a simple layered architecture consisting of User Interface, Application Logic, and Database Layer.

Architecture Components

- User Interface Layer (Activities, Fragments, RecyclerViews)
- Application Logic Layer (Java Classes, Adapters)
- Data Layer (SQLite Database and Shared Preferences)

Architecture Flow

User → UI Screens → Business Logic → SQLite Database → Display Results



3.2 Database Design

StudentOS uses **SQLite Database** to store and manage student-related information locally on the device. The database consists of three main tables: **Tasks**, **Notes**, and **Attendance**. These tables are designed to efficiently organize academic data and support CRUD (Create, Read, Update, Delete) operations within the application.

Tasks Table

The Tasks table stores information related to academic tasks and assignments. Each task contains a unique ID, task title, description, priority level, and deadline.

Field Name	Data Type	Description
id	Integer	Unique Task ID
title	Text	Task Title
description	Text	Task Description
priority	Text	Task Priority

Field Name	Data Type	Description
deadline	Text	Task Deadline

Notes Table

The Notes table is used to store academic notes created by the student. Each note contains a title, content, and category for better organization.

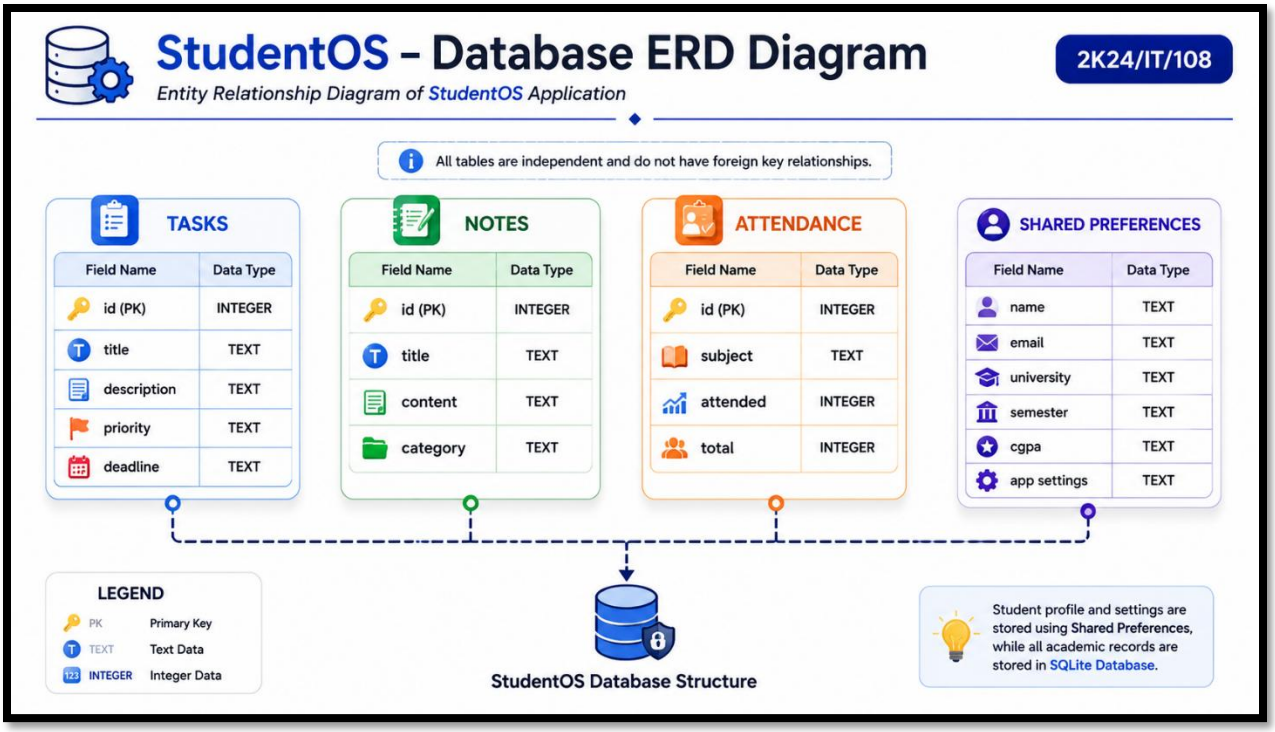
Field Name	Data Type	Description
id	Integer	Unique Note ID
title	Text	Note Title
content	Text	Note Content
category	Text	Note Category

Attendance Table

The Attendance table maintains attendance records for different subjects. It stores the number of attended classes and total classes, which are later used to calculate attendance percentage.

Field Name	Data Type	Description
id	Integer	Unique Record ID
subject	Text	Subject Name
attended	Integer	Classes Attended
total	Integer	Total Classes

Database Relationship:
The three tables operate independently and are connected through the application logic. Student profile information and CGPA are stored separately using Shared Preferences instead of SQLite Database.



3.3 Database Flow

The application stores user data locally using SQLite Database.

Flow:

User Input → Activity/Fragment → DatabaseHelper → SQLite Database → Display Data

3.4 Screen Navigation Diagram

The navigation structure of StudentOS is based on Bottom Navigation.

Home Dashboard

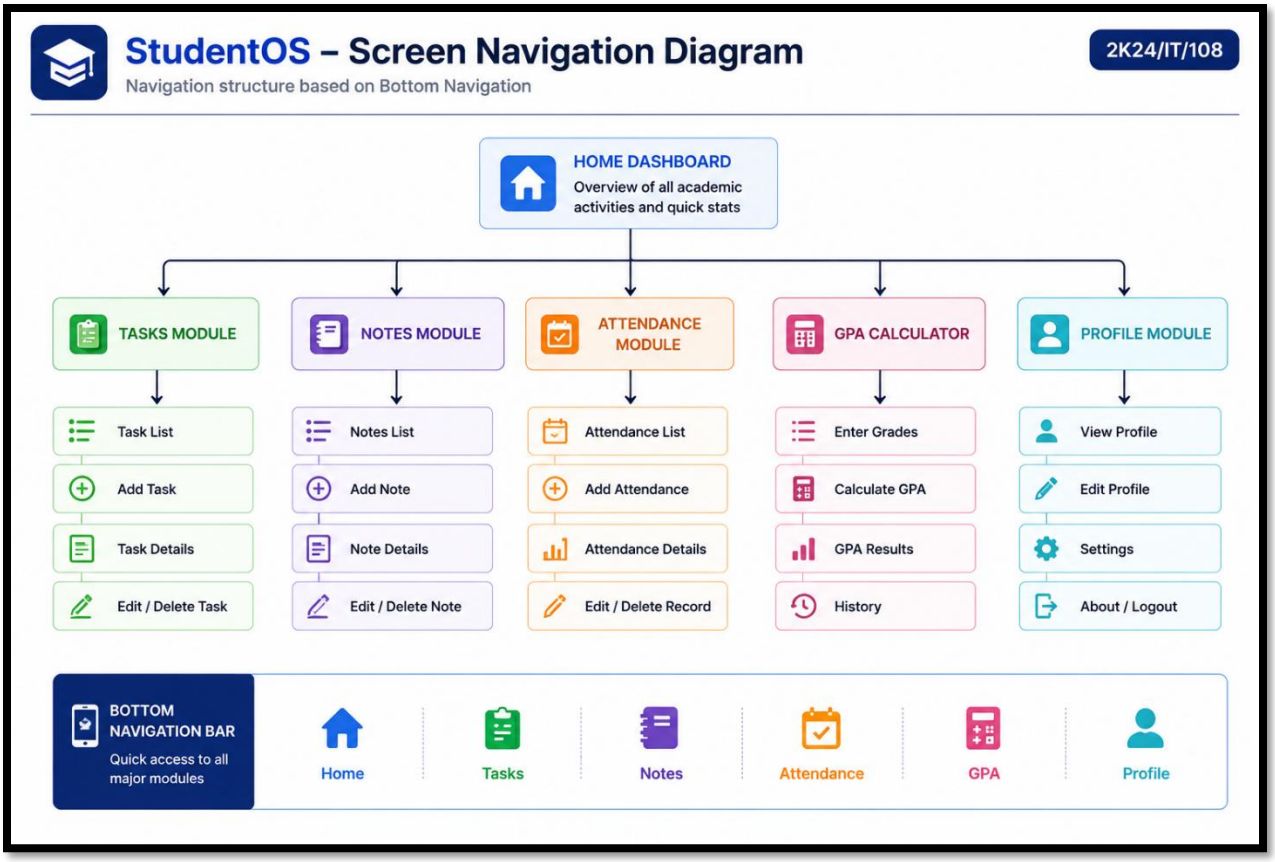
├── Tasks Module

├── Notes Module

├── Attendance Module

├── Profile Module

└── GPA Calculator

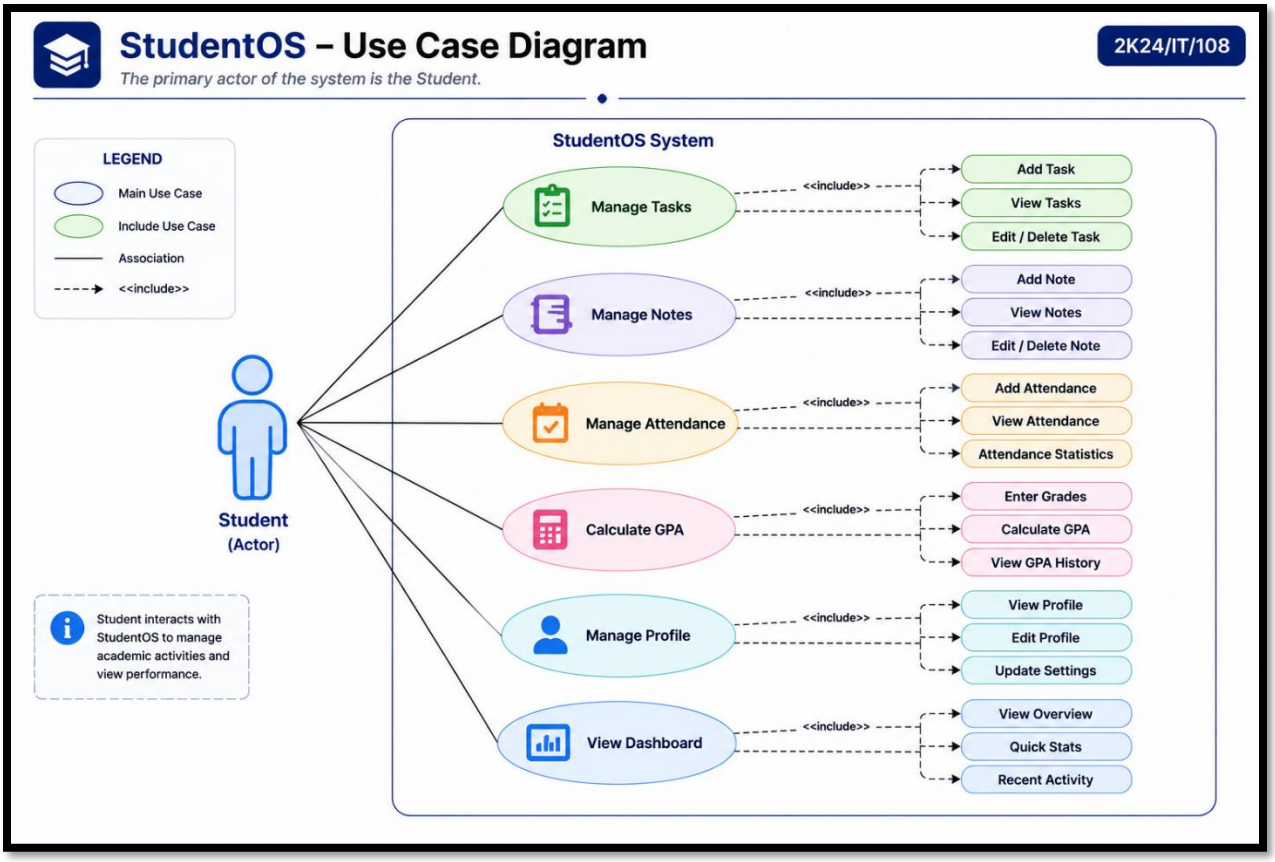


3.5 Use Case Diagram

The primary actor of the system is the Student.

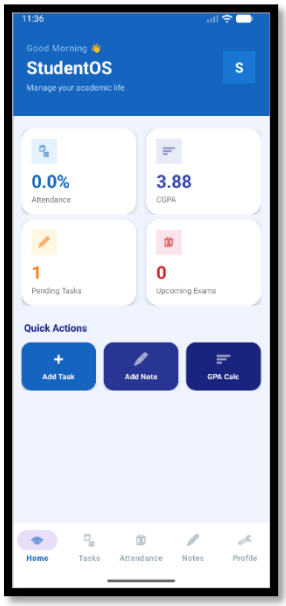
Main Use Cases

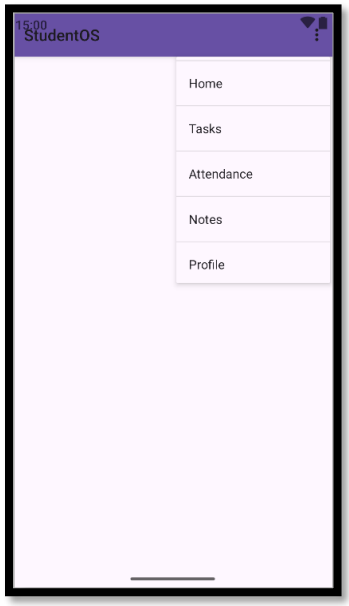
- Manage Tasks
- Manage Notes
- Manage Attendance
- Calculate GPA
- Manage Profile
- View Dashboard



3.6 User Interface Design

Home Screen





Purpose

The Home Screen serves as the dashboard of StudentOS. It displays attendance percentage, task count, GPA information, and provides quick access to different modules.

Tasks Screen

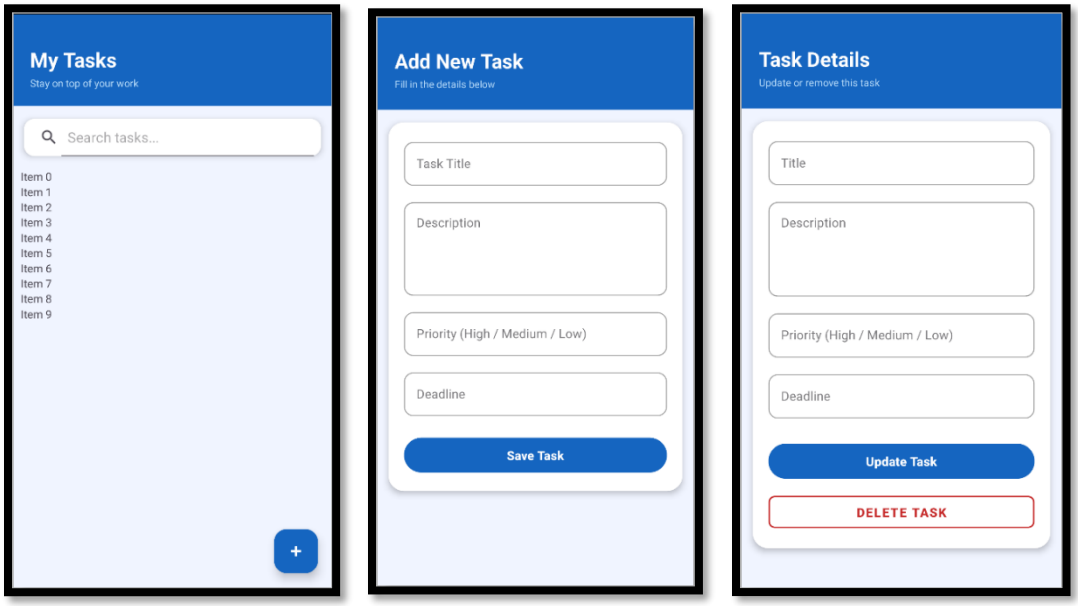
The Tasks module allows students to add, update, view, and delete academic tasks and assignments.

Add Task Screen

This screen allows users to enter task details such as title, description, priority, and deadline.

Task Detail Screen

This screen displays task information and provides update and delete functionality.



Notes Screen

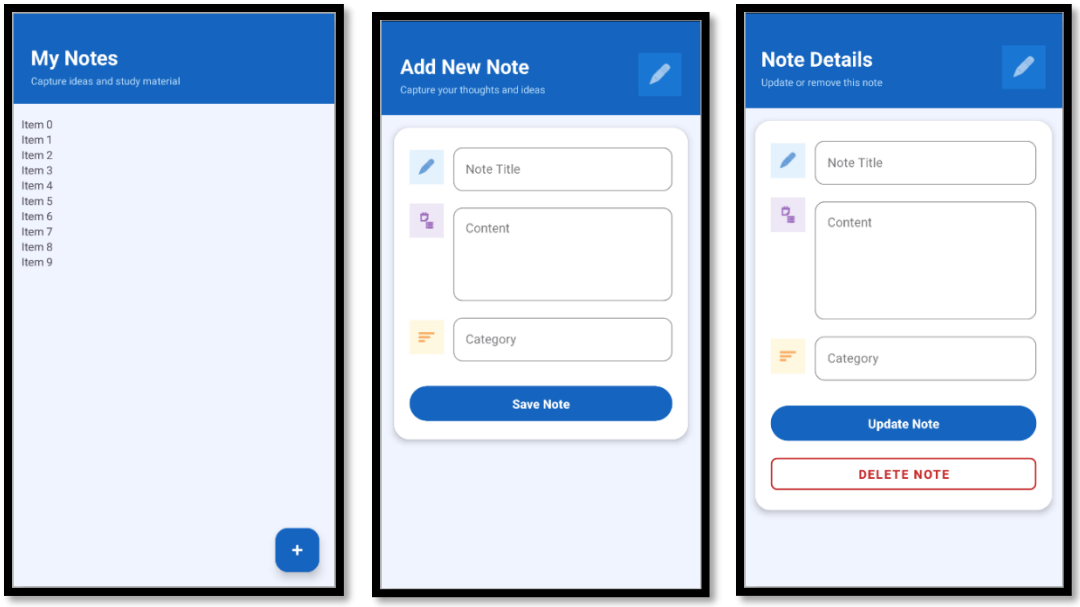
The Notes module helps students store and manage academic notes.

Add Note Screen

Allows students to create and save new notes.

Note Detail Screen

Displays complete note information and allows editing or deletion.



Attendance Screen

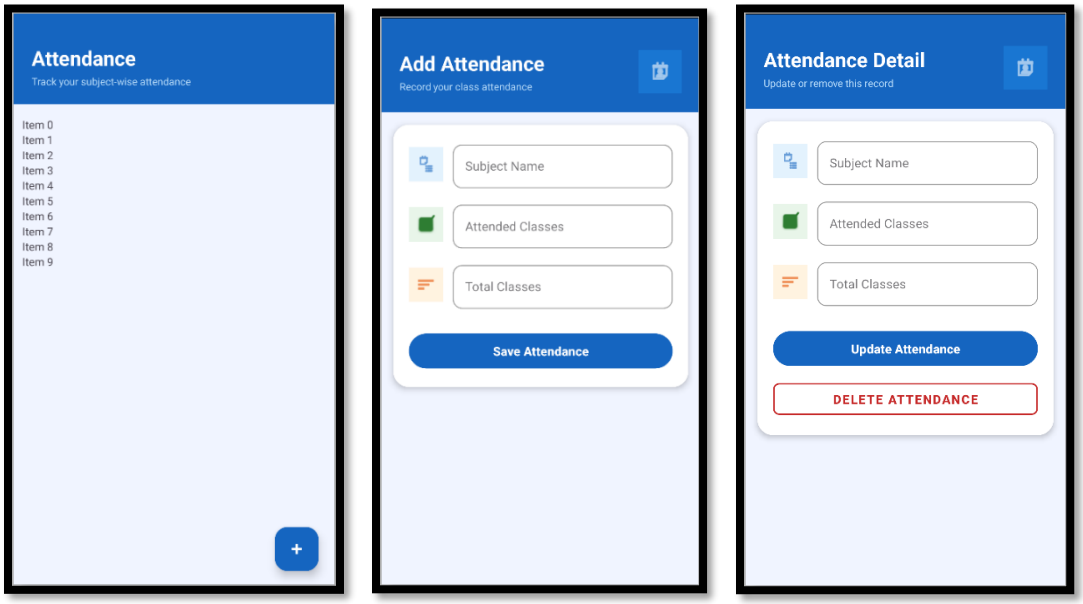
Allows students to track attendance records and attendance percentage.

Add Attendance Screen

Used for entering subject attendance information.

Attendance Detail Screen

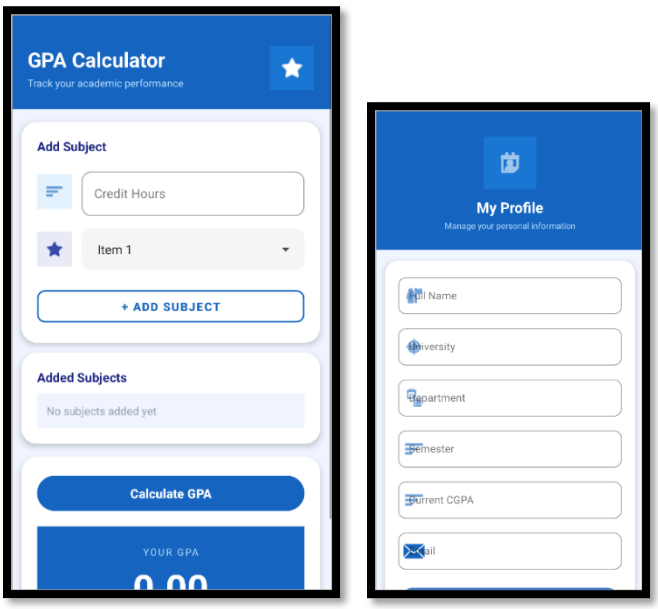
Allows updating and deleting attendance records.



GPA Calculator Screen and Profile

Calculates GPA based on grades and credit hours entered by the user.

Stores student personal and academic information such as name, university, semester, CGPA, and email.



Navigation Structure

StudentOS uses Bottom Navigation for easy access to all modules. The navigation provides a simple and user-friendly way to switch between Home, Tasks, Notes, Attendance, and Profile sections.

CHAPTER 4 – AGILE METHODOLOGY

4.1 Introduction

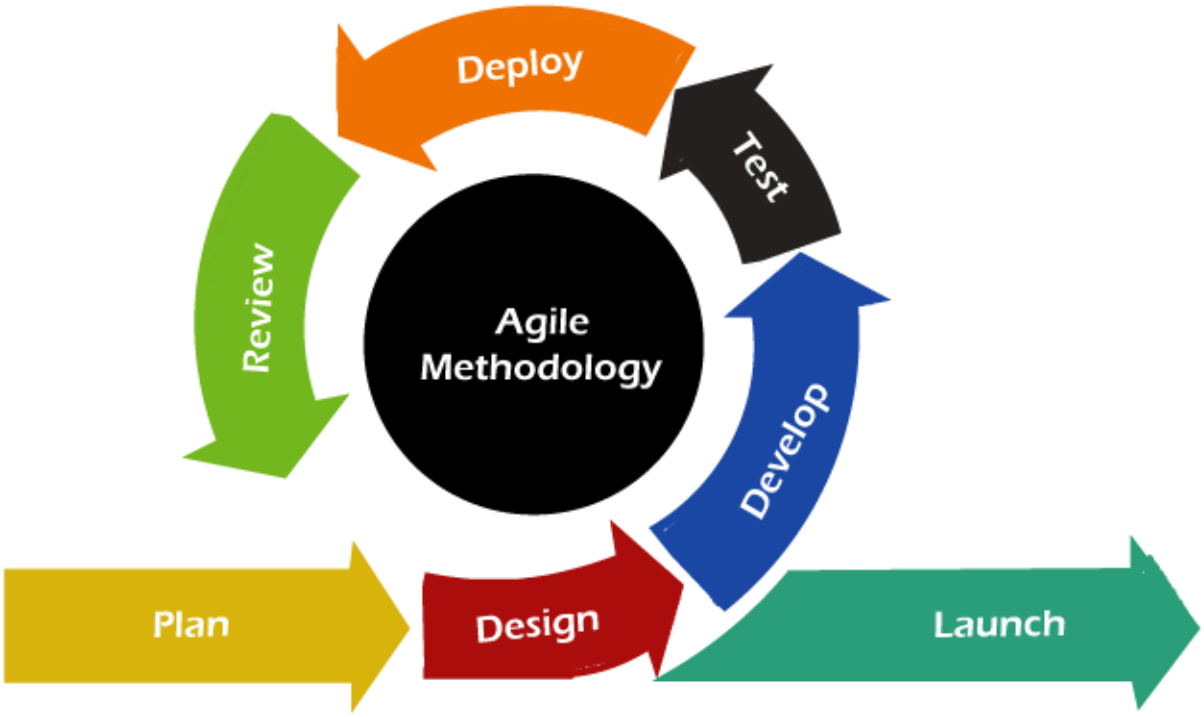
Agile is a software development methodology that focuses on developing applications in small and manageable phases. It allows developers to continuously improve the system through planning, development, testing, and feedback. Agile helps in delivering a functional product step by step rather than building the entire system at once.

For the development of StudentOS, Agile methodology was adopted to ensure systematic development and continuous improvement throughout the project. The project was divided into multiple phases including Planning, UI/UX Design, Development Sprint 1, Development Sprint 2, and Testing.

4.2 Agile Process in StudentOS

The Agile approach enabled the development of StudentOS in an organized manner. Each phase focused on specific objectives and produced measurable outcomes. Requirements were first identified, followed by interface design, feature implementation, database integration, and testing.

This methodology provided flexibility during development and allowed new improvements to be added whenever required. As a result, StudentOS was developed efficiently while maintaining quality and usability.



CHAPTER 5 – PLANNING

5.1 Introduction

Planning is the first phase of the Agile development process. In this phase, the project objectives, requirements, features, and technologies were identified before implementation. Proper planning helped establish a clear roadmap for the development of StudentOS.

5.2 Requirement Analysis

The main purpose of StudentOS is to help students manage their academic activities through a single mobile application. The system was planned to include task management, notes management, attendance tracking, GPA calculation, and profile management features.

The application was designed to provide a simple user interface, offline functionality, and easy access to academic information.

5.3 Project Modules

StudentOS was divided into several modules including Home Dashboard, Tasks, Notes, Attendance, GPA Calculator, and Profile Management. Each module was planned independently and later integrated into the complete system.

5.4 Development Tools

Android Studio was selected as the development environment, Java was used as the programming language, XML was used for interface design, SQLite was used for local database management, and Shared Preferences were used for storing profile information.

5.5 Outcome of Planning Phase

At the end of the planning phase, all system requirements were finalized, project modules were identified, and the development strategy was prepared. This phase provided a strong foundation for the design and implementation of StudentOS.

CHAPTER 6 – UI/UX DESIGN

6.1 Introduction

The UI/UX Design phase focused on creating a simple, user-friendly, and visually appealing interface for StudentOS. The application was designed to ensure easy navigation and quick access to academic information. The primary goal was to provide a clean interface that allows students to manage their tasks, notes, attendance, GPA, and profile efficiently.

6.2 User Interface Design

The user interface was developed using XML layouts in Android Studio. A consistent design theme was maintained throughout the application to improve usability and user experience.

The design includes cards, forms, buttons, lists, and navigation components that help users interact with different modules easily.

Design Features

- Simple and clean interface.
- Easy-to-read typography.
- Consistent color scheme.
- Responsive layouts.
- User-friendly forms and lists.
- Bottom Navigation for quick access.

6.3 Navigation Design

StudentOS uses Bottom Navigation as the primary navigation method. This allows users to switch between different modules without difficulty.

Navigation Structure:

Home → Tasks → Notes → Attendance → Profile

The GPA Calculator can be accessed directly from the Home Dashboard.

6.4 Dashboard Design

The Home Dashboard provides a quick overview of important academic information including attendance percentage, GPA information, and pending tasks.

The dashboard acts as the central screen of the application and helps users monitor their academic progress efficiently.

6.5 Forms Design

Forms were designed for adding and updating information within the application. Input fields, buttons, and validation mechanisms were used to ensure proper data entry.

Examples include:

- Add Task Form
- Add Note Form
- Add Attendance Form
- Profile Form

6.6 List Design

RecyclerView was used to display lists of tasks, notes, and attendance records. This provides efficient data presentation and improves performance when displaying multiple records.

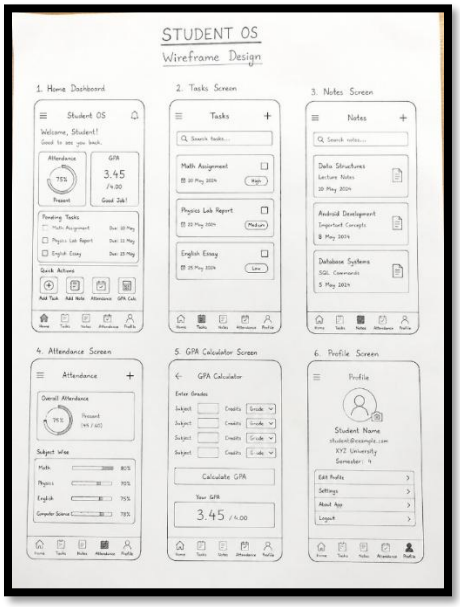
6.7 Detail Pages

Detail pages allow users to view, update, and delete existing records. Separate detail screens were developed for Tasks, Notes, and Attendance records.

These screens provide complete information about a selected item and support CRUD operations.

6.8 Outcome of UI/UX Design Phase

At the completion of this phase, all application screens were designed and connected through a structured navigation system. The resulting interface provides a smooth user experience and enables efficient interaction with all StudentOS modules.



CHAPTER 7 – DEVELOPMENT SPRINT 1

7.1 Introduction

Development Sprint 1 focused on implementing the core functionalities of StudentOS. During this sprint, the database structure was created and the fundamental modules required for academic management were developed.

7.2 Database Implementation

SQLite Database was integrated into the application to store and manage data locally. Separate tables were created for Tasks, Notes, and Attendance records. CRUD (Create, Read, Update, Delete) operations were implemented through the DatabaseHelper class.

7.3 Task Management Module

The Task Management module was developed to allow students to create, view, update, and delete academic tasks. Users can enter task details such as title, description, priority, and deadline.

7.4 Notes Management Module

The Notes module was implemented to help students store and organize study notes. Functionality for adding, viewing, updating, and deleting notes was provided.

7.5 Outcome of Sprint 1

At the end of Sprint 1, the database layer was completed and the Task and Notes modules were fully functional. Users were able to store and manage academic information successfully.

CHAPTER 8 – DEVELOPMENT SPRINT 2

8.1 Introduction

Development Sprint 2 focused on implementing the remaining modules and integrating all components of StudentOS into a complete application.

8.2 Attendance Management Module

The Attendance module was developed to record subject attendance information. The system calculates attendance percentages based on attended and total classes.

8.3 GPA Calculator Module

A GPA Calculator was implemented to help students calculate their GPA using grades and credit hours. This module provides quick academic performance evaluation.

8.4 Profile Management Module

The Profile module was developed using Shared Preferences to store user information such as name, email, university, semester, and CGPA.

8.5 Dashboard Integration

The Home Dashboard was developed to display important academic information including attendance percentage, task count, and GPA-related data. Navigation between modules was also completed during this sprint.

8.6 Outcome of Sprint 2

At the completion of Sprint 2, all major modules were successfully integrated into StudentOS. The application was fully functional and ready for testing and evaluation.

CHAPTER 9 – TESTING

9.1 Introduction

Testing is an important phase of software development that ensures the application functions correctly and meets user requirements. The StudentOS application was tested after the completion of all modules to identify and remove errors before final deployment.

9.2 Testing Approach

Functional testing was performed on each module individually as well as on the complete integrated system. The purpose was to verify that all features worked according to the specified requirements.

9.3 Test Cases and Results

Module	Test Performed	Result
Tasks	Add Task	Passed
Tasks	Update Task	Passed
Tasks	Delete Task	Passed
Notes	Add Note	Passed
Notes	Update Note	Passed
Notes	Delete Note	Passed
Attendance	Add Attendance	Passed
Attendance	Update Attendance	Passed
Attendance	Delete Attendance	Passed
GPA Calculator	Calculate GPA	Passed
Profile	Save Profile Data	Passed
Dashboard	Display Dynamic Data	Passed
Navigation	Screen Navigation	Passed

9.4 Issues Encountered

During development, a few issues were identified including navigation errors, database integration problems, and user interface adjustments. These issues were resolved through debugging and repeated testing.

9.5 Results

The testing process confirmed that all modules of StudentOS function correctly. CRUD operations, attendance calculations, GPA calculations, profile management, and dashboard functionality worked successfully without major errors.

9.6 Conclusion

The StudentOS application was successfully tested and all major functionalities operated as expected. The application met its objectives of providing an integrated academic management solution for students. The final system was stable, user-friendly, and ready for deployment.

CHAPTER 10 – CONCLUSION

10.1 Learning Outcomes

The development of StudentOS provided valuable practical experience in Android application development. Through this project, knowledge of Java programming, XML-based user interface design, SQLite database management, Shared Preferences, Activities, Fragments, RecyclerView, and CRUD operations was strengthened.

The project also improved problem-solving skills, debugging techniques, database integration knowledge, and understanding of the Agile software development methodology. Furthermore, it provided hands-on experience in designing and developing a complete mobile application from planning to deployment.

10.2 Challenges Faced

Several challenges were encountered during the development of StudentOS. One of the major challenges was integrating multiple modules while maintaining smooth navigation between screens. Database implementation and CRUD operations required careful testing to ensure accurate data storage and retrieval.

Another challenge involved designing a user-friendly interface that was both functional and visually appealing. Debugging runtime errors, handling data updates, and ensuring proper synchronization between different modules also required significant effort during development.

These challenges were successfully resolved through continuous testing, debugging, and iterative improvements throughout the Agile development process.

10.3 Future Improvements

Although StudentOS successfully meets its current objectives, several enhancements can be added in future versions to increase functionality and usability.

Potential future improvements include:

- Cloud database integration.
- User authentication and login system.
- Timetable management module.
- Push notifications and reminders.
- Online backup and synchronization.
- Dark mode support.
- Assignment deadline alerts.
- Advanced academic analytics and reports.
- Multi-device data synchronization.
- Integration with university management systems.

These improvements would further enhance the application and transform StudentOS into a more comprehensive academic management platform for students.