# CHAPTER FOUR

**SYSTEM IMPLEMENTATION**

## 4.1 INTRODUCTION

This chapter discusses the implementation of the student activities planner providing an overview of the choice of programming language and platform that was used in the implementation of the system. It also discusses the various features of the system.

## 4.2 PLATFORM AND TOOLS FOR IMPLEMENTATION

The tools discussed below were used in the implementation of the proposed student activities.

1. **Android SDK:** It is a software development kit that enables developers to create applications for the Android platform. It also separates the Android SDK manager into platforms, SDK tools, and other components into their different packages for easy management and use. The Android SDK includes sample projects with their source code, development tools, an emulator, and necessary libraries to build Android applications. Applications are written using the flutter programming language.
2. **Flutter:** Flutter is an open-source mobile application development framework created by Google. It allows developers to build high-performance, visually attractive, and feature-rich mobile apps for both Android and iOS platforms using a single codebase. Flutter uses a reactive programming model, which makes it easy to build complex user interfaces with animations, gestures, and more. Flutter comes with a rich set of pre-built widgets and tools that make it easy to create custom and visually appealing user interfaces. It also provides a hot reload feature that allows developers to see the changes they make to the code in real time without having to rebuild the entire app. Flutter is built on the Dart programming language, which is also created by Google. Dart is an object-oriented language that is easy to learn, and it offers a wide range of features like classes, mixings, interfaces, and more.
3. **Dart:** Dart is an open-source, object-oriented programming language created by Google. It is designed to be easy to learn and use, yet powerful enough to build complex applications for a variety of platforms, including web, mobile, desktop, and server. Dart is a statically typed language, which means that the type of a variable is known at compile time. This makes it easier to catch errors early in the development process and can improve the performance of the code. One of the key features of Dart is its ability to compile JavaScript, allowing developers to build web applications that can run in any modern web browser. Dart also includes a Just-In-Time (JIT) compiler, which can be used during development to provide faster iteration cycles, and an Ahead-Of-Time (AOT) compiler, which can be used to generate highly optimized native code for mobile and desktop applications.
4. **Firebase:** Firebase is a cloud-based platform developed by Google that provides developers with a variety of tools and services to build and manage web and mobile applications. It includes features such as a real-time database, authentication, cloud messaging, storage, hosting, and more. The real-time database is a NoSQL cloud-hosted database that allows developers to store and sync data in real-time across multiple clients. This means that any changes made to the data by one client will be immediately available to all other clients connected to the database.
5. **JSON:** JavaScript Object Notation is a readable format for structuring data that is used majorly to pass data between a server and a web application; it is used as an alternative to XML. It is also known as lightweight data because it is easy for humans to read and write and also easy for machines to parse and generate.
6. **Visual Studio Code (Vscode):** Visual Studio Code is a free and open-source code editor developed by Microsoft. It is available for Windows, macOS, and Linux, and supports many programming languages, including JavaScript, Python, C++, and Java, among others. Vscode is designed to be highly customizable, with a wide range of extensions and themes available to enhance the editor's functionality and appearance. It includes features such as code highlighting, auto-completion, debugging, Git integration, and IntelliSense, which provides smart code completion and suggestions based on the code being written. One of the strengths of Vscode is its extensibility. Developers can install extensions to add new functionality or customize the editor to their specific needs. Extensions can be created by anyone using the Vscode Extension API, and there are thousands of extensions available in the Visual Studio Marketplace, covering a wide range of use cases.

## 

## 4.3 PROGRAM MODULES AND INTERFACES

This section contains screenshots that show the different interfaces and their functionalities.

**4.3.1 The splash screen page**

The splash screen is a screen that appears for a brief moment while the application is loading. The splash screen is typically used to provide a seamless transition from the application launch to the main screen by displaying branding, logos, or other visually appealing images.

****

Figure 4.1 splash screen page

**4.3.2 The sign-up page**

The sign-up page is a screen page that allows a user to create a new account by providing personal information such as their name, email address, username, and password. The sign-up page typically includes input fields for the user to enter their personal information, as well as a button to submit the form and create their account.

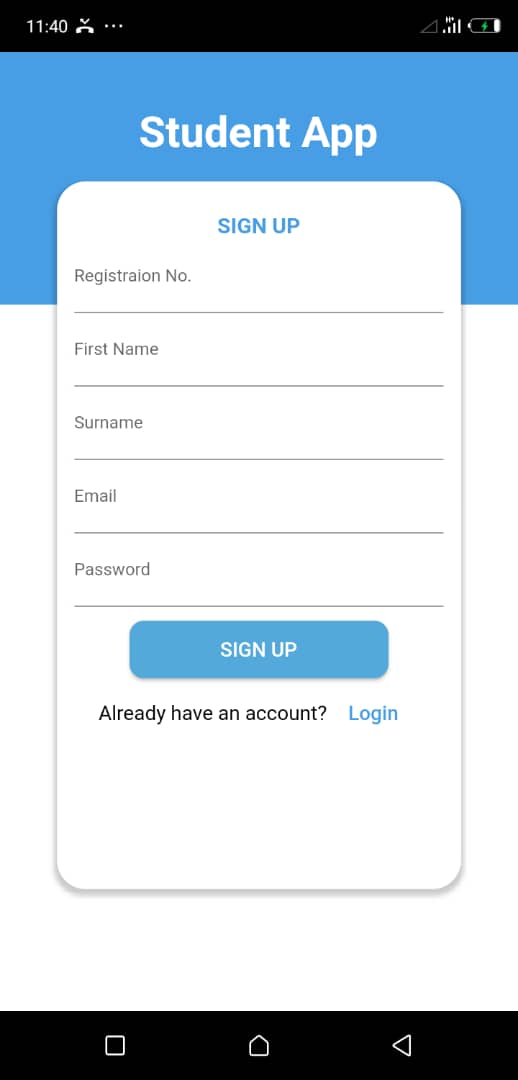


Figure 4.2 sign-up page

**4.3.3 The sign-in page**

The sign-in page is the page that allows users to enter their login credentials, such as username and password, to access the application. It is typically the first page a user sees when they attempt to access the application. The sign-in page typically includes fields for the user to enter their login information and also includes options for creating a new account. Once the user has entered their login credentials correctly, they are redirected to the main dashboard of the application.

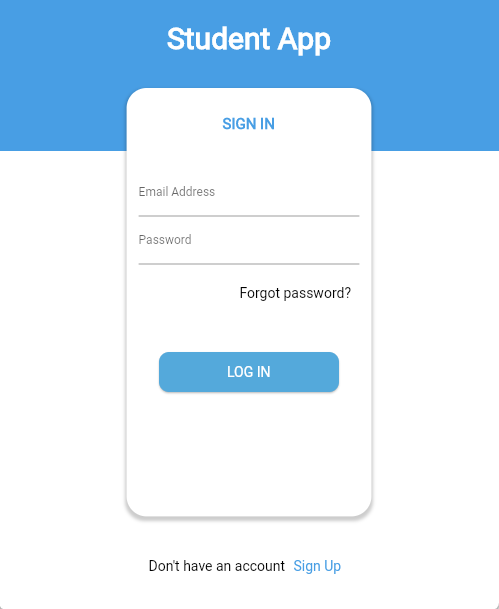


Figure 4.3 sign-in page

**4.3.4 The dashboard page**

The dashboard page refers to the home page of the application that provides a summary of important information or actions to the user. It also includes a sidebar drawer that allows the user to perform various functionalities within the application.

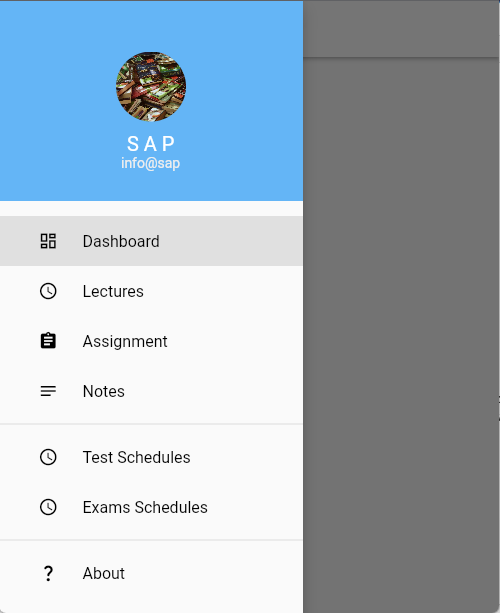


Figure 4.4 Dashboard page

**4.3.5 The lectures page**

The lecture page is an interface page within the application that allows the user to create and add a new lecture schedule to a collection, and also view the list of lectures added. To add a new lecture schedule, a user must input the course code, venue, time, and day for the lecture.

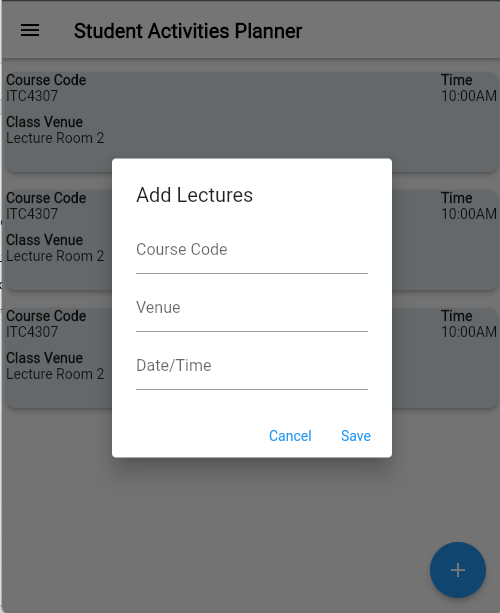


Figure 4.5 lecture page

**4.3.6 The note page**

The note page allows the user to create or add a new note to a list or collection. To add a new note, a user must enter the course code for the note and the note itself.

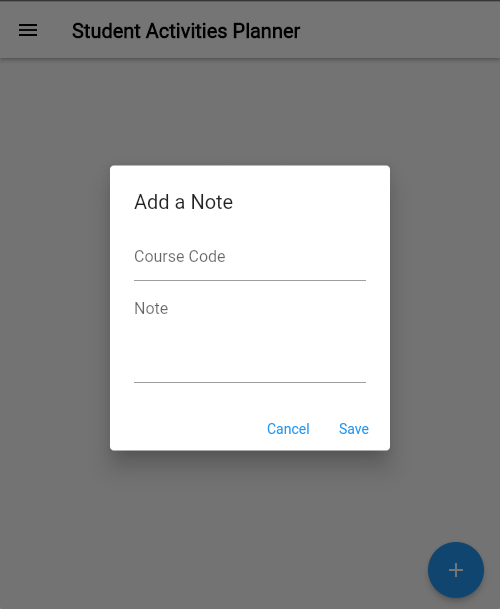


Figure 4.6 note page

**4.3.7 The exam page**

The exam page is an interface page within the application that allows the user to create and add a new exam schedule, and also view the list of exam schedules added. To add a new exam schedule, a user must input the course code, venue, time, and day for the exam.

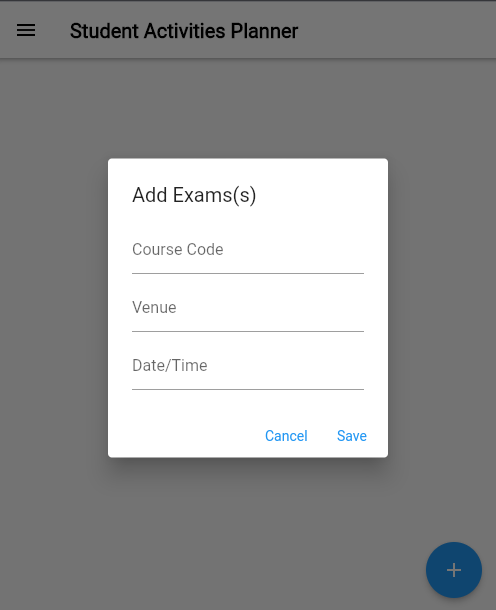


Figure 4.7 exam page

**4.4 CONCLUSION**

This chapter highlights some of the technical details of the application implementation, including the programming languages used, the data storage methods, and the application modules (interfaces) and their functionalities.