

**Gannan Normal University**



**《Doccumentation》**

****Project Title**: SmartCampusCompass**

****Team Name**:** HyperNova Studios  
****Submission Date**:** [April 12, 2025]

## ****Team Introduction****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Member** | | **Role** | **Responsibilities** | **Contact** | |
| [Afser Seddek Tasfe Dhali (贾威医)] | | Team Lead & Developer | Full-Stack Development, Plan and Software Management., Frontend Development | [tasfeafseramv@gmail.com] | |
| [S M SHAFI (邵飞)] | | Secqurity and Quality Assurancer & Developer | Project Management, Budge Management, planing and developing | [asshafi15@gmail.com] | |
| [ROQUNUZZAMAN (罗昆)] | UI/UX Designer & Developer | | Back end Development, Prototyping | | [Khanrokon571@gmail.com] |

# ABSTRACT

This document presents the comprehensive design, development, and deployment process of the "Smart Campus Compass" mobile application. The application is tailored to support university students by offering an all-in-one solution for navigating campus infrastructure, tracking academic events, accessing schedules, and improving student engagement with the university environment. It emphasizes real-time campus navigation, notification of academic and social events, user account management, and class reminders. This report systematically outlines the feasibility analysis, system requirements, design architecture, module-wise implementation, user interface development, testing strategies, and final observations. The project aims to address real-life campus challenges, optimize student productivity, and provide a scale-able digital solution for future academic institutions.

**Keywords:** Smart campus system, Mobile application, Student performance tracking, Real-time class navigation, Firebase backend, Flutter framework, Campus schedule optimization, Location-based services, Cross-platform app development, Student engagement system

Index

[Team Introduction 2](#_Toc537273610)

[ABSTRACT 3](#_Toc1936311282)

[1. Foreword 4](#_Toc616529936)

[2. Feasibility Analysis 5](#_Toc410037577)

[2.1 Technical Feasibility 5](#_Toc226533416)

[2.2 Economic Feasibility 5](#_Toc2006100228)

[2.3 Operational Feasibility 6](#_Toc1033274096)

[3. Requirement Analysis 6](#_Toc1684961830)

[3.1 Functional Requirements 6](#_Toc286623821)

[3.2 Non-Functional Requirements 7](#_Toc480739326)

[4. System Design 7](#_Toc952372068)

[4.1 General Design 7](#_Toc1321725785)

[4.2 Detailed Design 7](#_Toc674423927)

[5. Code Implementation 8](#_Toc624252223)

[6. UI Design 9](#_Toc1349496366)

[1.Login and Registration Page 9](#_Toc1410627395)

[2.Home Dashboard Page 9](#_Toc195164885)

[3.Campus Map Page with Search 9](#_Toc928693226)

[4.Lost & Found 9](#_Toc635902986)

[5.Carrier Path 9](#_Toc1742858230)

[7. System Testing 10](#_Toc541326530)

[8. Conclusion 11](#_Toc1334261018)

# Foreword

In modern academic institutions, student satisfaction and success heavily depend on access to real-time information, efficient time management, and well-connected digital services. Recognizing these needs, our team undertook the development of "Smart Campus Compass," a mobile application that seeks to improve the day-to-day experiences of students within a university setting. The idea was inspired by daily campus challenges: navigating large campuses, missing important notices, and struggling to remember class or event timings. By leveraging modern cross-platform development tools and cloud-based technologies, we aimed to develop an application that not only solves these problems but also offers a user-friendly and scalable solution that could be extended across institutions. This documentation chronicles the complete development cycle, from early brainstorming to implementation and final testing, and highlights the real-world impact such a tool can create in academic life.

# 2. Feasibility Analysis

## 2.1 Technical Feasibility

The app is built using Flutter, a UI toolkit developed by Google for crafting natively compiled applications for mobile from a single codebase. It supports reactive-style programming, which makes it easier to handle real-time updates like map tracking and event notifications. Firebase serves as our backend solution, providing services such as authentication, a real-time database, cloud storage, and messaging. The Google Maps API was incorporated to support navigation features. The tech stack is highly scalable and backed by strong community support and documentation, which ensured we could implement the necessary features with minimal risk.

## 2.2 Economic Feasibility

Budget constraints are always a key concern in student-led projects. We made cost-effective choices by selecting frameworks and platforms that offer free usage tiers. Flutter is open-source, while Firebase offers a flexible free tier that fully supports the scope of our app. Our primary investment was time and effort rather than financial costs. No external hardware or paid services were required, keeping the overall budget close to zero. This economic efficiency ensures that the app can be replicated by other student teams or institutions without significant financial barriers.

## 2.3 Operational Feasibility

Usability was prioritized throughout the development process. Surveys and feedback from fellow students revealed their expectations—simplicity, speed, and clarity. Based on this input, the user interface was designed to be intuitive and visually clean. All features were organized with minimal navigation steps. The home screen provides direct access to core features: maps, events, and schedules. During testing, students from different departments easily adapted to using the app within a few minutes. Operationally, the app is viable for real-world deployment in a university environment and requires no training or technical background from users.

# 3. Requirement Analysis

## 3.1 Functional Requirements

1. User registration and secure login/logout system
2. Password reset and profile management options
3. Real-time interactive map with search and routing to campus buildings
4. Timely alerts and notifications for classes, exams.
5. Event feed showing all upcoming university activities
6. Dashboard

## 3.2 Non-Functional Requirements

It should be responsive and adapt to various screen sizes

Authentication and user data must be securely stored and transmitted

The app should support real-time updates and push notifications

Future-proofing: scalable architecture that supports new modules like chatbot integration

Low-latency performance and minimal loading times even under large data volume.

# System Design

## 4.1 General Design

The system architecture is based on the MVC pattern. The Model layer interacts with Firebase Firestore to manage user and event data. The Controller layer processes inputs, interacts with APIs, and updates the View accordingly. The View layer, built with Flutter widgets, offers a seamless and reactive UI. Each feature is divided into modules that follow SOLID principles to maintain code readability and facilitate future expansion.

## 4.2 Detailed Design

Authentication Module: Allows users to register, sign in, and log out; linked to Firebase Auth

Navigation Module: Google Maps API is used to display interactive maps and live location updates

Reminder Module: Allows setting up reminders for specific events or class schedules

Notification Module: Integrated with Firebase Cloud Messaging for real-time push alerts

Settings & Profile Module: Lets users update profile data and app preferences

Admin Panel (future plan): Role-based access for teachers or admins to post updates

# 5. Code Implementation

The application codebase is organized into multiple Dart files, grouped by feature. The primary tools and packages used include:

firebase\_core, firebase\_auth, cloud\_firestore, firebase\_messaging, provider, google\_maps\_flutter

**main.dart:** Initializes Firebase, routing, and global providers

**app\_localization.dart**: File manages localization in a Flutter app by providing translated strings and handling locale-specific content for multiple languages.

**weather\_data.dart:** file in a Flutter app typically handles fetching, storing, and processing weather-related data, such as temperature, humidity, and forecast information, from an API or local source for display in the app.

**ar\_navigation\_screen.dart:** Initializes the AR navigation screen by setting up necessary resources. Builds the UI for the AR navigation screen with real-time location data.

### ****canteen\_screen.dart:**** Builds the UI for the Canteen screen with search functionality, grid items, and button interactions. Creates individual grid items for the canteen categories with icons, labels, and taps.

### ****career\_path\_screen.dart :****Displays the career path options and information for the user. Renders various career options available in the career path.

### ****claim\_found\_screen.dart :****Displays the form to claim a found item with necessary fields for user input Handles the submission of the claim for a found item.

### ****help\_desk\_screen.dart :**** Builds the UI for the help desk page with buttons and support options Submits a request to the help desk system for assistance.

### ****history\_screen.dart :**** Renders the user's history of actions or activities within the app. Loads historical data related to the user's activity.

### ****login\_screen.dart :**** Builds the UI for the login screen with fields for username, password, and login button. Handles the user login process by authenticating credentials.

### ****lost\_found\_screen.dart :**** Displays options for lost and found items with respective UI components. Navigates to the screen to claim a lost item.

### ****menu\_screen.dart :**** Builds the main menu UI with navigational options for various app features. Handles navigation based on the selected menu item.

### ****notifications\_screen.dart :**** Displays the notifications screen with a list of notifications. Marks a notification as read when selected by the user.

### ****placeholder\_form\_screen.dart :**** Displays a placeholder form with input fields and a submit button. Handles form submission by collecting input data and processing it.

### ****profile\_screen.dart :**** Builds the UI for the profile screen with the user's details.Updates the user profile data after changes are made.

### ****report\_item\_screen.dart :****Displays the screen for reporting a lost or found item with necessary input fields.Handles the item report submission with appropriate validation.

### ****setting\_screen.dart :**** Renders the settings screen with options to change app preferences.Saves the changes made to the app settings.

### ****signup\_screen.dart :**** Builds the UI for the signup screen with fields for registration Handles user registration process by saving the new account details.

### ****view\_found\_items\_screen.dart :**** Displays a list of found items with relevant details for the user to view. Filters the found items based on selected criteria or search parameters.

### ****view\_lost\_items\_screen.dart :**** Displays a list of lost items with relevant information for the user to view. Filters the lost items based on selected criteria or search parameters.

### ****weather\_screen.dart :****Displays the current weather and forecast details. Fetches weather data from an external API and updates the UI.

### ****weather\_service.dart** :** Retrieves weather data from a weather API based on location or city.Parses the weather data and extracts useful information for display.

### ****app\_theme.dart** :** Retrieves the current app theme, including colors and text styles.Sets the theme for the application based on user preferences or default settings.

### ****app\_shell.dart :**** Constructs the basic app layout with app bar, body, and navigation options.Configures the app bar for the screen, including title and actions.

### ****firebase\_option.dart :**** Initializes Firebase services for the app (authentication, Firestore, etc.). Configures Firebase authentication options like login and signup.

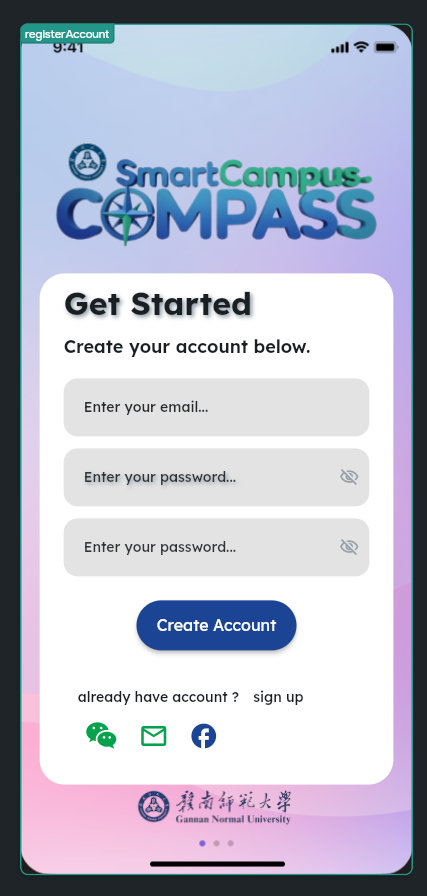
We used Git for version control and hosted the code repository on GitHub for collaboration among the team. Code was tested on real Android devices and emulators during each development stage.

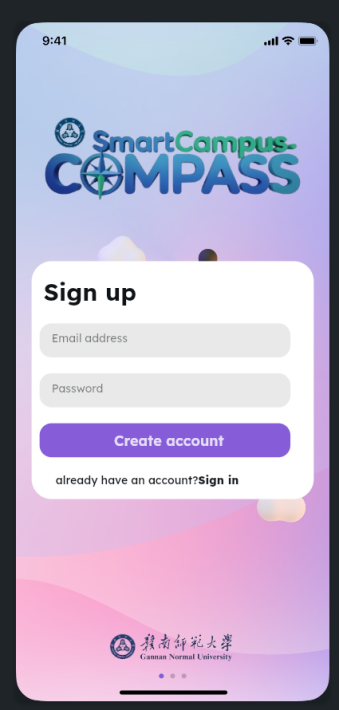
# 6. UI Design

## 1.Lsign up and Create Account Page

Description : The **Sign Up** page enables new users to create an account and access the platform by providing basic personal information and selecting login credentials (e.g., username and password). The page also often includes an email or phone verification process to ensure the user's identity and protect against spam or fraudulent accounts.

UI:



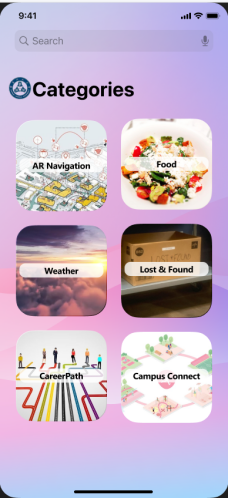


## 

## 2.Categories Page

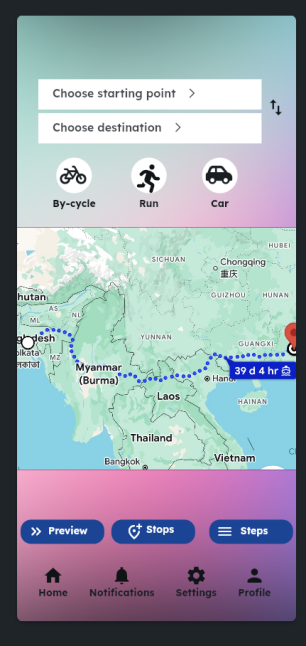
Description :This page displays a grid of categorized options (e.g., North Canteen, South Canteen, Stores, Others) that the user can tap to view more specific content under each category.

UI:

:

## Campus Map Page with Search

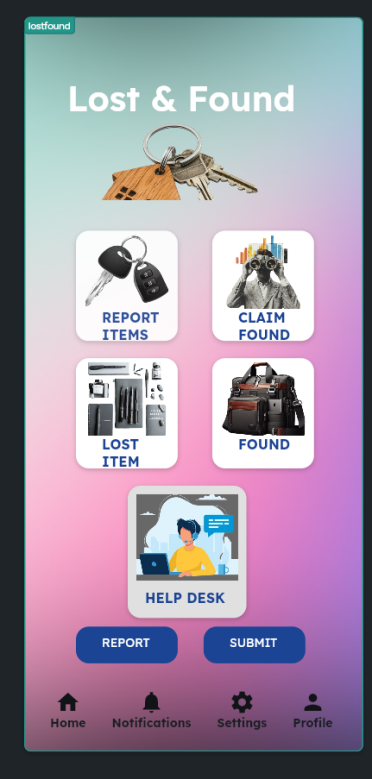
## Description :The ****Campus Map**** page helps users visualize and explore the layout of the campus



## 4.Lost & Found

Description :The **Lost and Found** page allows students, staff, and visitors to report **lost** or **found** items within the campus.

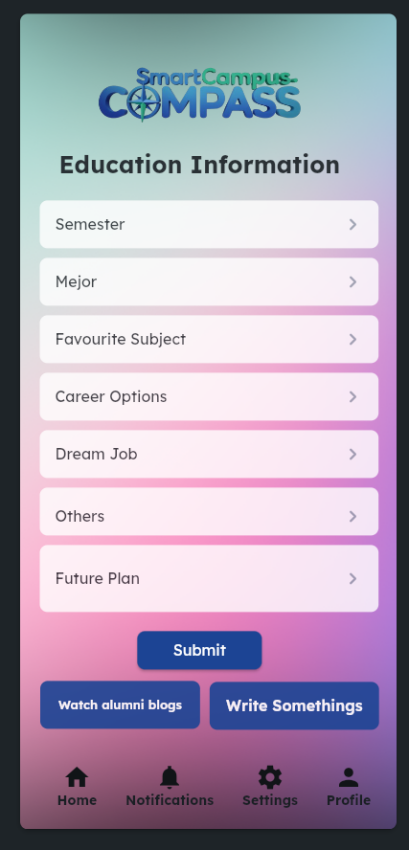
UI:



## 5.Career Path

Description :The **Career Path** page is designed to help students explore, understand, and choose different career options based on their interests, academic background, and future goals

UI:



## 6.Weather

Description :The **Weather** page provides users with up-to-date information about the current weather and forecasts for a specific location. This page helps users plan their day by providing insights into temperature, humidity, wind speed, air quality, and more. It could be tied to their current location or allow users to search for different cities.

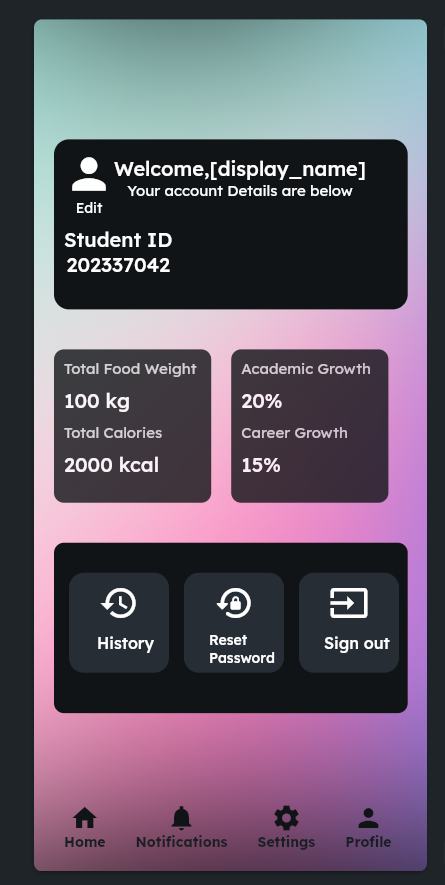
UI:



1. **Profile page**

Description:The **Profile** page serves as a centralized hub for users to view and update their personal information, preferences, and account settings. It helps users customize their experience, manage privacy settings, and view key details about their account (such as username, profile picture, and activity). This page enhances user interaction by providing personalization options.

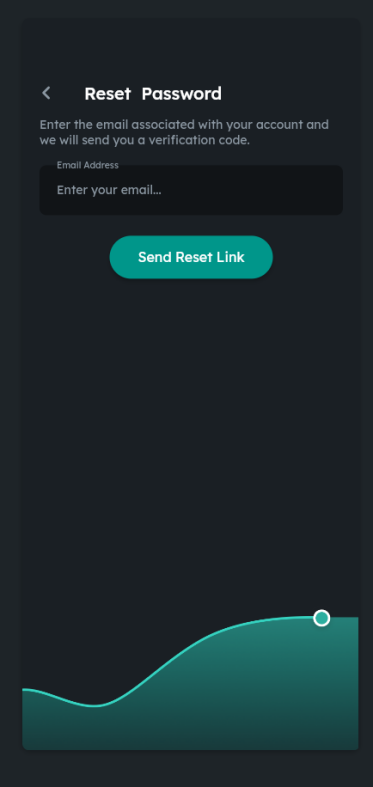
UI:



**8.Reset Password**

Description**:**The **Reset Password** page allows users to recover or change their password securely if they have forgotten it or need to update it for security reasons

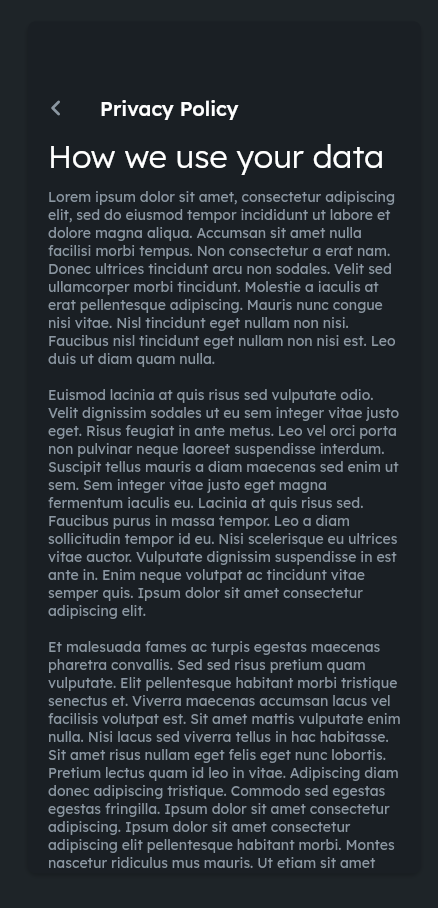
UI:



1. **Privacy and Policies**

Description:The **Privacy and Policies** page ensures transparency by outlining the practices, rights, and responsibilities related to user data, and sets the legal terms users must agree to when using a service or platform. This page is crucial for complying with data protection laws and building trust with users

UI:



# 7. System Testing

We conducted rigorous testing to ensure a high-quality, bug-free user experience. Both functional and non-functional tests were planned and executed. Key testing activities included:

**Unit Testing:** Verifying each function such as login, map loading, and data fetch from Firebase

**Integration Testing:** Ensuring different modules (e.g., authentication + navigation) worked together smoothly

**UI Testing:** Validating visual consistency across devices with different screen sizes

**Usability Testing:** Gathering student feedback on ease-of-use and design clarity

**Performance Testing:** Measuring map rendering speed, notification delivery time, and app responsiveness under load

**Security Testing:** Basic validation of user input and safe storage of credentials using Firebase rules

**Test Environment:**

Devices: Xiaomi, OnePlus, Huawei smartphones, iOS emulator

Tools: Android Studio Emulator, Flutter DevTools

Bug reports were tracked, prioritized, and resolved using GitHub Issues.

# 8. Conclusion

The development of Smart Campus Compass allowed our team to practically apply software engineering principles and tools to solve a real-world academic problem. From idealization and prototyping to actual deployment and testing, every phase was handled with careful planning and teamwork. The project emphasized the power of user-focused design, agile development, and efficient tool usage. Through the use of Flutter and Firebase, we built a scale-able, responsive, and feature-rich application that can be expanded to serve universities beyond our own.

Moving forward, we plan to enhance the app by integrating offline support, real-time chat, AI-driven event suggestions, and multilingual support. The project has been an enriching learning experience and sets a strong foundation for future innovations in smart campus systems.