Software Requirements Specification (SRS)

“Smart Compass Campus”

Authors : Md. Bilayet Hossain (221-15-5821)

Al Razi Talukder (221-15-5356)

Client :

Instructor : Md.Umaid Hasan (Lecturer,Dept of CSE,DIU)

1. Introduction:

1.1. Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the development of the Smart Compass Campus application. This document serves as a reference for stakeholders, including developers, testers, and project managers, to ensure the successful implementation of the application.

1.2. Scope

Smart Compass Campus is a mobile and web application designed to provide navigation and location-based services within a university campus. The application aims to improve the overall campus experience for students, faculty, and visitors by offering features such as navigation, points of interest (POI) discovery, event notifications, and access to campus information.

1.3. Definitions, Acronyms, and Abbreviations

GPS: Global Positioning System

POI: Points of Interest

UI: User Interface

API: Application Programming Interface

1.4. References

1.DIU Student

2. DIU 1Card

3.Compas Application

2. Overall Description:

2.1. Product Perspective

Smart Compass Campus is a standalone mobile application that operates independently. It interfaces with campus maps and databases to provide navigation and location-based services. While it is independent, it may include integrations with external services like weather forecasts and calendar applications.

2.2. Product Features

The core features of Smart Compass Campus include:

***Outdoor Navigation:** Providing real-time navigation guidance for outdoor areas using GPS.

***Indoor Navigation:** Utilizing Bluetooth beacons and Wi-Fi for accurate indoor navigation.

***POI Discovery:** Displaying information and images of campus POIs.

***Event Notifications:** Notifying users of campus events, lectures, and activities.

***Campus News:** Delivering news and updates relevant to the campus community.

***User Profile Management:** Allowing users to create and manage their profiles.

2.3. User Classes and Characteristics

***Students:** Primary users who navigate the campus, discover events, and access academic information.

***Faculty:** Users who use the app for navigation and event notifications.

***Visitors:** Individuals who require assistance in exploring the campus.

2.4. Operating Environment

Smart Compass Campus will be available on both **Android** and **iOS** platforms. It requires a GPS-enabled device with internet connectivity.

2.5. Design and Implementation Constraints

The application will be developed using modern mobile and web app development frameworks and technologies. It should prioritize user-friendliness, accessibility, and compatibility with assistive technologies.

2.6. Assumptions and Dependencies

*Assumptions include the availability of reliable GPS and Wi-Fi coverage across the campus.

*Dependencies may include third-party APIs for weather forecasts or campus event schedules.

3. System Overview:

3.1 System Description

The Smart Compass Campus system consists of the following key components:

***User Interface:** Includes web and mobile interfaces for users to interact with the system.

***Navigation Engine:** The core component responsible for calculating and providing navigation directions.

***Database:** Stores campus maps, POI data, user profiles, and event information.

***User Profile Management:** Allows users to create, update, and manage their profiles.

4. Functional Requirements:

4.1 User Registration and Authentication

*****Users can register for an account.

*****Users can log in using email and password.

*****Users can log in using social media accounts.

*****User profiles should be securely stored.

4.2 User Profile Management

*****Users can create and update their profiles.

*****Users can set preferences for navigation (e.g., walking speed, accessibility options).

*****Users can save favorite locations.

4.3 Campus Navigation

*****Users can search for locations on campus.

*****Users can view indoor and outdoor maps.

*****Users can receive turn-by-turn directions to their destination.

*****The system provides alternative routes based on user preferences.

*****The system can provide accessible routes for users with disabilities.

4.4 Points of Interest (POI)

*****Users can search for POIs **(e.g., classrooms, restrooms, cafeterias).**

*****Users can view detailed information about a POI, including hours of operation.

4.5 Personalized Recommendations

*****The system can recommend POIs based on user interests and location.

*****The system can suggest campus events and activities based on user preferences.

4.6 Event Notifications

*****Users can subscribe to receive notifications about campus events.

*****Event notifications are sent in real-time and can be personalized.

4.7 Campus News:

*****Display a news feed with campus news and updates.

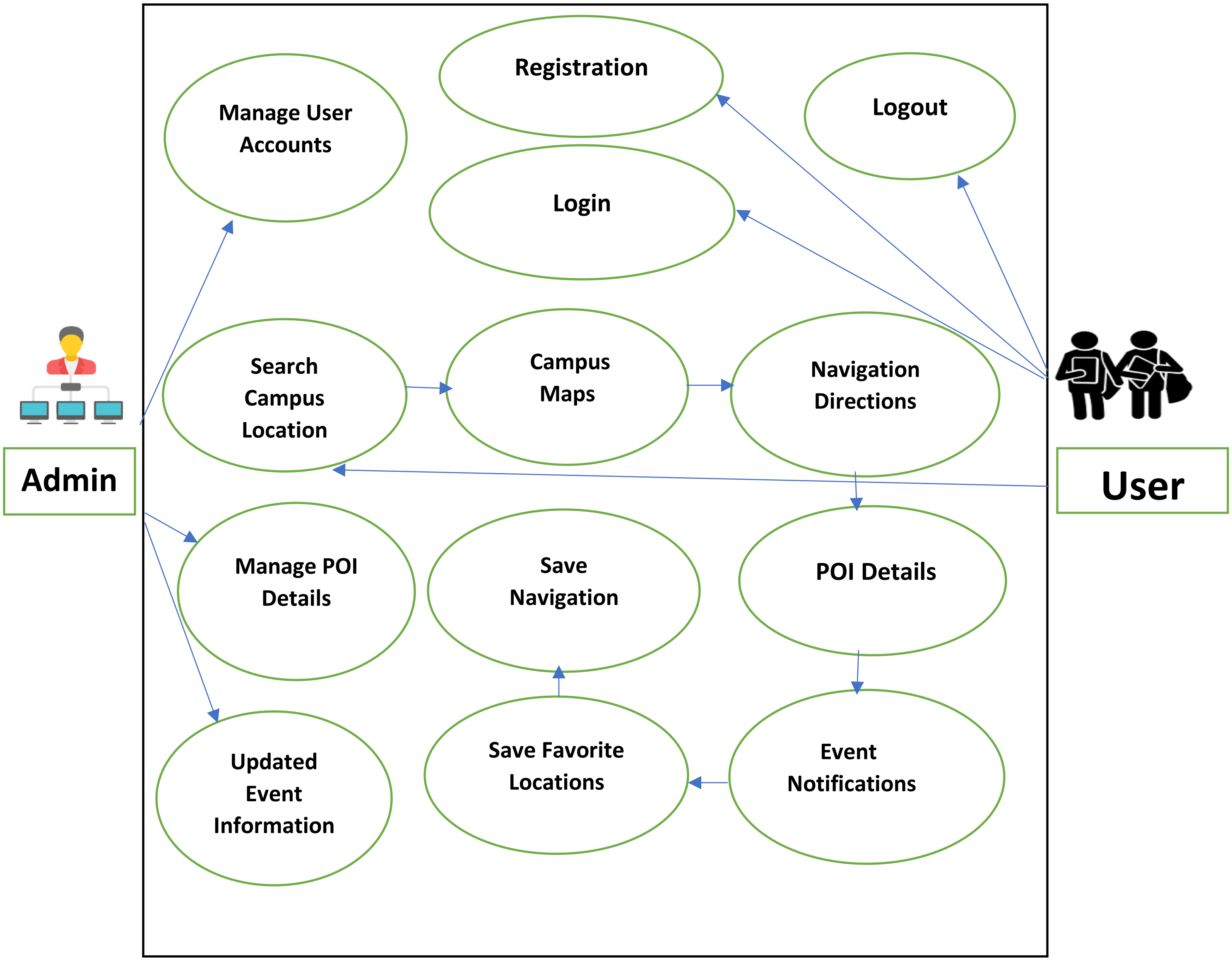
***** Allow users to filter news by categories **(e.g., academic, sports, events).**

4.8 Accessibility:

*Ensure the app complies with accessibility standards (e.g., WCAG) for users with disabilities.

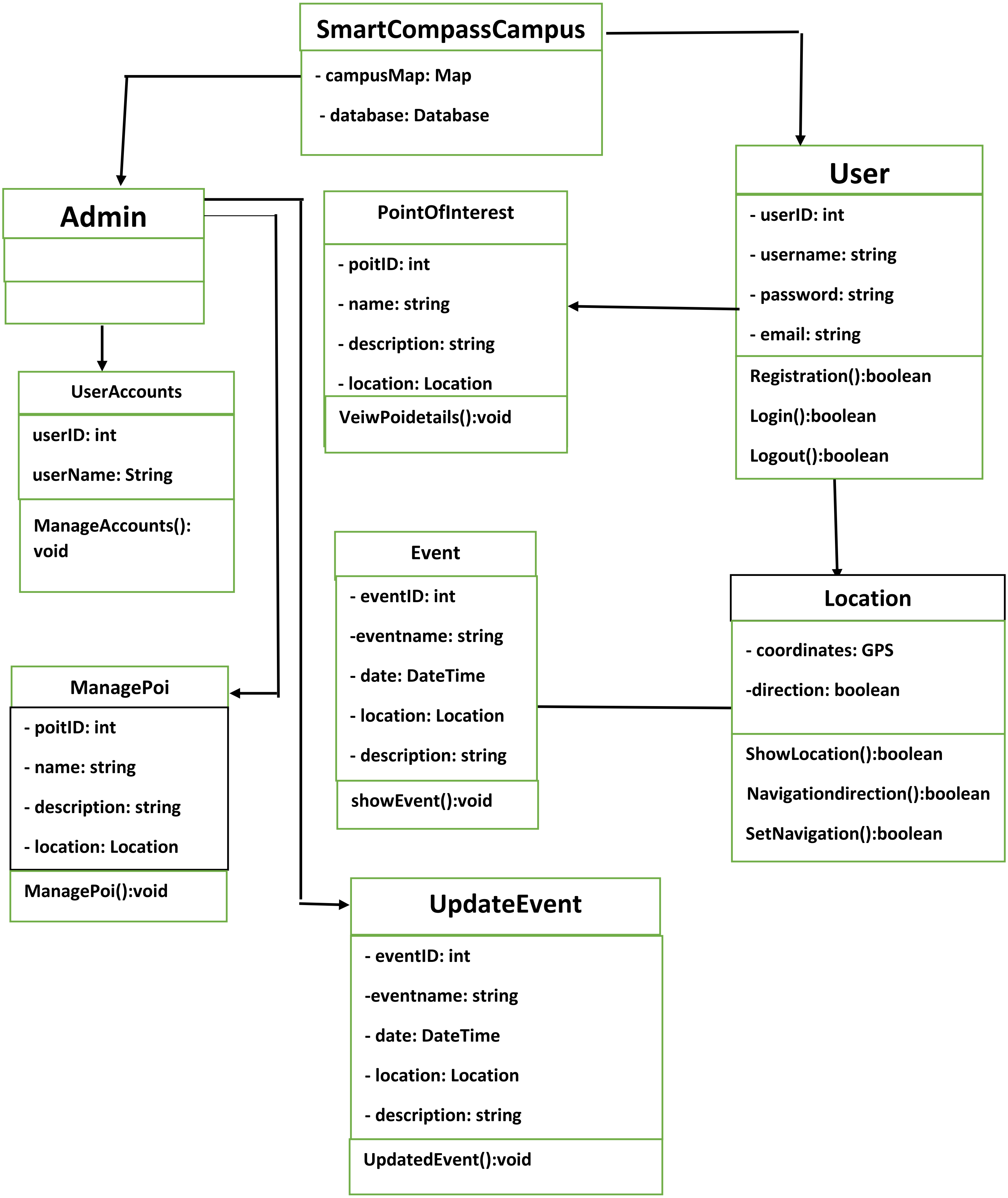
5. Modeling Requirements

*Use Case Diagram



In this simplified User Case Diagram, we have **two primary actors**: **"User"** and **"Admin." The "User"** interacts with various use cases related to **navigation, POI**, **preferences,** and **notifications.** The "**Admin"** is responsible for managing **user accounts, POI data,** and **event information** within the system.

*Class Diagram



SmartCompassCampus
- campusMap: Map
- database: Database

Admin

User
- userID: int
- username: string
- password: string
- email: string

Registration():boolean
Login():boolean
Logout():boolean

PointOfInterest
- poitID: int
- name: string
- description: string
- location: Location

VeiwPoidetails():void

UserAccounts
userID: int
userName: String

ManageAccounts(): void

Event
- eventID: int
-eventname: string
- date: DateTime
- location: Location
- description: string

showEvent():void

Location
- coordinates: GPS
-direction: boolean

ShowLocation():boolean
Navigationdirection():boolean
SetNavigation():boolean

ManagePoi
- poitID: int
- name: string
- description: string
- location: Location

ManagePoi():void

UpdateEvent
- eventID: int
-eventname: string
- date: DateTime
- location: Location
- description: string

UpdatedEvent():void

In this simplified Class Diagram:

***User:** Represents users of the system, with attributes such as user ID, username, password and email.

***UserAccounts:** Admin manages user accounts details with atrributes such as ID, and **username.**

***PointOfInterest:** Represents points of interest on campus, with attributes like POI ID, name, description, and location.

***Event:** Represents campus events, with attributes including event ID, name, date, location, and description and Admin can also Update event information.

***ManagePOI:** Admin Manage POI details with atrributes poi id,name ,description and location.

***Location:** Represents geographical locations on the campus, including coordinates (e.g., GPS coordinates).

6. Non-Functional Requirements:

6.1 Performance Requirements

* Real-time navigation should provide accurate and timely instructions.

*The system should respond to user requests within **1 seconds**.

*The system should be able to handle a minimum of **1,000 concurrent** users during peak times.

6.2 Usability

*The user interface should be intuitive and user-friendly.

*The system should be accessible to users with disabilities.

6.3 Security Requirements

*User data and privacy must be protected, complying with relevant data protection regulations.

*User authentication and authorization should be robust.

*Ensure secure data transmission between the app and servers.

6.4 Compatibility

*****The system should work seamlessly on popular web browsers **(e.g., Chrome, Firefox**) and mobile platforms **(iOS, Android).**

6.5 Reliability

*****The system should have a backup and recovery mechanism to prevent data loss.

7. Other Requirements

7.1 Support multiple languages to accommodate international users.

7.2 Provide comprehensive user manuals and developer documentation.

7.3 Implement thorough testing procedures to identify and resolve bugs and issues.

8.Conclusion:

In conclusion, the Smart Campus Project has ushered in a new era of efficiency, sustainability, and connectivity within our educational institution. Over the course of this project, we have successfully implemented a range of cutting-edge technologies and solutions designed to enhance the overall campus experience for students, faculty, and staff.

Our smart campus infrastructure, comprising sensor networks, data analytics systems, and automation tools, has significantly improved campus operations.