
---

# IT - 314: Software Engineering

Group - 11

## Software Requirements Specification

---

Event Sphere

**Event Management and Participation Platform**

Prof - Dr. Saurabh Tiwari

Group Mentor - Priyanshu

**Group Members :**

| No | Student Id | Name |
|---|---|---|
| 1 | 202201104 | Rudraksh Asthana |
| 2 | 202201105 | Rishi Shah |
| 3 | 202201150 | Dhrudeep Sharma |
| 4 | 202201162 | Smit Godhani |
| 5 | 202201163 | Jay Goyani |
| 6 | 202201165 | Parshwa Modi |
| 7 | 202201167 | Het Gandhi |
| 8 | 202201171 | Sahil Chaudhari |
| 9 | 202201175 | Kathan Kadiya |

# INDEX

## 6. Technology Used

6.1 Frontend

5.2 Backend

5.3 Database

5.4 Development Tools

## 7. Bugs in our project

## 8. Not Implemented

## 9. References

# Chapter 1

# Introduction

## 1.1 Purpose of the document

This document presents the Software Requirements Specification (SRS) for the Event Sphere Platform. It aims to provide a comprehensive overview of the system's objectives, functionality, and target audience, ensuring effective communication and alignment among stakeholders, designers, and developers.

## 1.2 Overview of Event Sphere Platform

The Event Sphere Platform aims to connect colleges, clubs, committees, and participants in a seamless environment. Colleges can register and add their clubs and committees, which can then list events for students and other users to explore and participate in. Clubs and committees can track their event listing history, while students can keep track of their participation history. The platform ensures a smooth connection between event organizers and participants, making event management more accessible and organized.

## 1.3 Product Scope

This project involves developing a web-based event management system to connect colleges, clubs, committees, and participants. It will allow colleges to register, add clubs and committees, list events, and enable participants to register and track their event history. The platform will centralize event discovery and participation across various domains.

This document outlines the functional and non-functional requirements to guide the design, development, testing, and deployment of the platform. Key features include event registration, student verification, eligibility criteria, and history tracking for both organizers and participants.

# Chapter 2

# Specific Requirements

## 2.1 Requirement Elicitation Techniques

Effective requirement elicitation is crucial for developing a platform that meets user needs and expectations. To ensure that the Event Sphere Platform is well-aligned with the requirements of colleges, clubs, committees, and participants, we employed two primary techniques to gather and refine the system's features:

1. Brainstorming
2. Use case scenarios

These techniques helped us identify essential features and functionalities, ensuring that the platform delivers value to all users and provides a seamless event management experience.

### 1. Brainstorming

Brainstorming was used to generate ideas and solutions for addressing the needs of our event management platform. A group of participants, led by a leader, contributed their ideas freely, encouraging creativity and collaboration. This technique helped uncover innovative approaches for our system.

### Two main activities:

- ☐ **The Storm (Generating Ideas):** During this phase, we encouraged sharing all ideas, including practical and non-practical ones. The following ideas were discussed:
    - Login/Signup for users, colleges, clubs
    - See/ Update profile
    - Clubs/committees list, modify events
    - Event cataloging and organization
    - Advanced search and filter options for events

- Notifications and reminders for new event listings'
- User participation in events
- Track participation history
- Personalized recommendations
- Feedback and ratings for events
- Admin panel for managing events, users, and feedback
- Event registration and payment processing (if applicable)
- Blog page for news, updates, and community engagement

☐ **The Calm (Filtering Ideas)**: After refining and discussing the ideas, we finalized the key features for the initial development stages:

- Login/Signup for users, colleges, clubs
- See/ Update profile
- Clubs/committees list, modify events
- Event cataloging and organization
- Advanced search and filter options for events
- User participation in events
- Track participation history
- Admin panel for managing events, users, and feedback
- Event registration and payment processing (if applicable)
- Blog page for news, updates, and community engagement

### 2. Use Case Scenarios

A detailed explanation of the use case scenarios can be found in Section 4 of this SRS document. These scenarios outline the specific user interactions with the platform, ensuring a seamless experience for event organizers, participants, and admins.

## 2.2 Users and Stakeholders

**Users:** A user is an individual who interacts with the Event Sphere platform.

- Participants (Students, simple users)
- Event Organizers (Clubs and Committees)
- Admin

**stakeholders:** A stakeholder is anyone who impacts or is impacted by the platform's operations.

- Developers
- Designers
- Testers

## 2.3 User Stories

### 2.3.1 Participants Stories

1. As a visitor, I want to explore available events without registering, so that I can browse what the platform offers before committing to registration.
   - **Acceptance**: Visitor can view event titles, categories, dates, and locations.
   - **Success**: Visitor can browse events without registering.
   - **Failure**: Visitor is prompted to log in or register to access event details or register for an event.

2. As a student, I want to register on the platform using my student ID so that I can access events relevant to my preferences.
   - **Acceptance**: Registration requires name, email, student ID, and password.
   - **Success**: The student receives a confirmation email upon successful registration and can log in using their email and password.
   - **Failure**: If the registration details are incorrect or incomplete, the system prompts the student to correct them.

3. As an Event enthusiast, I want to register on the platform so that I can access events relevant to my preferences.
   - **Acceptance**: Registration requires name, email, and password.
   - **Success**: The user receives a confirmation email upon successful registration and can log in using their email and password.

- **Failure**: If the registration details are incorrect or incomplete, the system prompts the user to correct them.

4. As a student, I want to search for events by keywords, categories, dates, and locations so that I can easily find and participate in events that match my interests and schedule.
   - **Acceptance**: The search functionality allows searching by keywords, categories, dates, and locations.
   - **Success**: Accurate and relevant results are displayed based on the search criteria.
   - **Failure**: If no events match, a "No events found" message is displayed with suggestions to refine the search.

5. As a student, I want to register for an event and make payments if necessary so that I can secure my participation.
   - **Acceptance**: Students can register for events through the event detail page.
   - **Success**: If payment is required, the system prompts for payment, processes it securely, and updates the student's participation history.
   - **Failure**: If payment fails or registration is incomplete, the system notifies the student and prevents participation.

6. As a student, I want to filter events by categories, dates, and location so that I can quickly find events that match my interests.
   - **Acceptance**: The event browsing page allows filtering by categories, date range, and location.
   - **Success**: Only events that match the selected filters are displayed.
   - **Failure**: If no events match the filters, the system shows a message indicating no results and allows clearing filters to view all events.

7. As a student, I want to manage my profile and view my participation history so that I can track my event involvement.

- **Acceptance**: Users can update their personal information (e.g., email, password, profile picture).
- **Success**: The profile displays a history of all events the user has participated in, including event feedback.
- **Failure**: If the profile update fails or the participation history is unavailable, an error message is displayed.

8. As a user, I want two-factor authentication for added account security.
   - **Acceptance**: Users can enable two-factor authentication via SMS, email, or an app.
   - **Success**: The user is guided through a simple setup process.
   - **Failure**: If setup fails, an error message is shown.

### 2.3.2 Organizers Stories

1. As a college, I want to register on the website so I can add my clubs and committees.
   - **Acceptance**: The college can complete a registration form with necessary details.
   - **Success**: The college successfully registers and is able to add clubs and committees.
   - **Failure**: If registration details are incomplete or incorrect, the system prompts the college to correct them.

2. As a college representative, I want to register my college on the platform so that I can list events and manage participants.
   - **Acceptance**: College provides name, accreditation, contact info, and password.
   - **Success**: Registration confirmed by email, and the college can log in.
   - **Failure**: Invalid accreditation or incorrect details prompts for correction.

3. As a college representative, I want to list events on the platform so that students from various colleges can participate.
   - **Acceptance**: College can access a form to input event details.
   - **Success**: The event is successfully listed and visible to all registered users.
   - **Failure**: If the college is not registered, the event cannot be listed, and the system prompts the college to register first.

4. As a college representative, I want to view analytics on event participation so that I can assess the success of our events.
   - **Acceptance**: The college dashboard displays metrics like participant count, and payment summaries.
   - **Success**: Analytics are shown in graphical formats (e.g., charts, graphs) for easy understanding.
   - **Failure**: If the analytics data is unavailable, an error message is shown prompting the user to try again later.

5. As a college representative, I want to update event details (venue, timing, etc.) so that the participants are up to date with the schedule.
   - **Acceptance**: The event is already listed by the college.
   - **Success**: The event details are updated, and participants receive a notification about the changes.
   - **Failure**: If the event details cannot be updated, an error message is displayed.

6. As a college representative, I want to add a blog about an event so that I can provide useful information about the event.
   - **Acceptance**: The event is already listed by the college.
   - **Success**: The blog post is added and displayed on the event page, accessible to all interested users.
   - **Failure**: If the blog cannot be added, an error message is displayed.

### 2.3.3 Admin Stories

1. As an admin, I want to manage and moderate the platform content to maintain quality.
   - **Acceptance**: Admins can manage accounts, events, and feedback.
   - **Success**: Admins can flag/remove inappropriate content, and actions are logged.
   - **Failure**: An error message is displayed if an action cannot be performed.

2. As an admin, I want to be able to remove an event or blog from the platform.
   - **Acceptance**: Admin can remove an event or blog, deactivating it and removing all associated content.
   - **Success**: A confirmation message is displayed, and the action is logged for audit purposes.
   - **Failure**: If removal fails, an error message is displayed, and no changes are made.

3. As an admin, I want the platform to support modular updates to add features without disrupting services.
   - **Acceptance**: Platform supports modular updates with minimal dependencies.
   - **Success**: Updates are deployed without downtime.
   - **Failure**: If downtime occurs, the update is rolled back and the admin is notified.

## 2.4 Requirement Analysis

### 2.4.1 Functional requirements

1. User Registration and Authentication
   - Secure registration and login for users, colleges, clubs, and admin.
   - Role-based access control ensures users access only relevant features.

- Email verification is required, with an OTP sent to the user's email for authentication.

2. Event Listing by Colleges
   - Colleges can add and manage events with details such as title, description, date, time, venue and guidelines.
   - Events are visible to all registered users for participation.
   - Event details can be updated by the college representative.
3. Event Participation
   - Users can register for events and track their participation.
   - If required, payments for events are processed securely during registration.
   - Event registration updates the user's participation history.
4. Search and Filter Events
   - Users can search for events by keywords, categories and locations.
   - Advanced filtering options are available to narrow down event search results.
5. Profile Management
   A. User Profile Management
      - Users can create and update their profiles with personal details like name, email, password, and profile picture.
      - Profile pages show user participation history,

   B. College profile Management
      - Colleges can view their own details.
      - Colleges can add clubs and committees to the platform.
      - Colleges can view a list of their registered clubs.

   C. Club Profile Management
      - Clubs can view their registered events and blogs.
      - Clubs can manage events by editing or deleting them.
      - Clubs can delete blogs associated with their events.
      - Clubs can view participants for each event.

   D. Admin Panel
      - Admins can view all registered colleges, their associated clubs, and the events and blogs listed by those clubs.
      - Admins can manage and delete events and blogs that are irrelevant or violate platform guidelines.

6. Payment Integration
    - If required, event registrations include payment processing functionality.
    - Payments are securely processed using third-party payment gateways (Here, we have not implemented actual payment processing; instead, we have used a dummy interface where the payment is automatically completed upon clicking the payment option.)

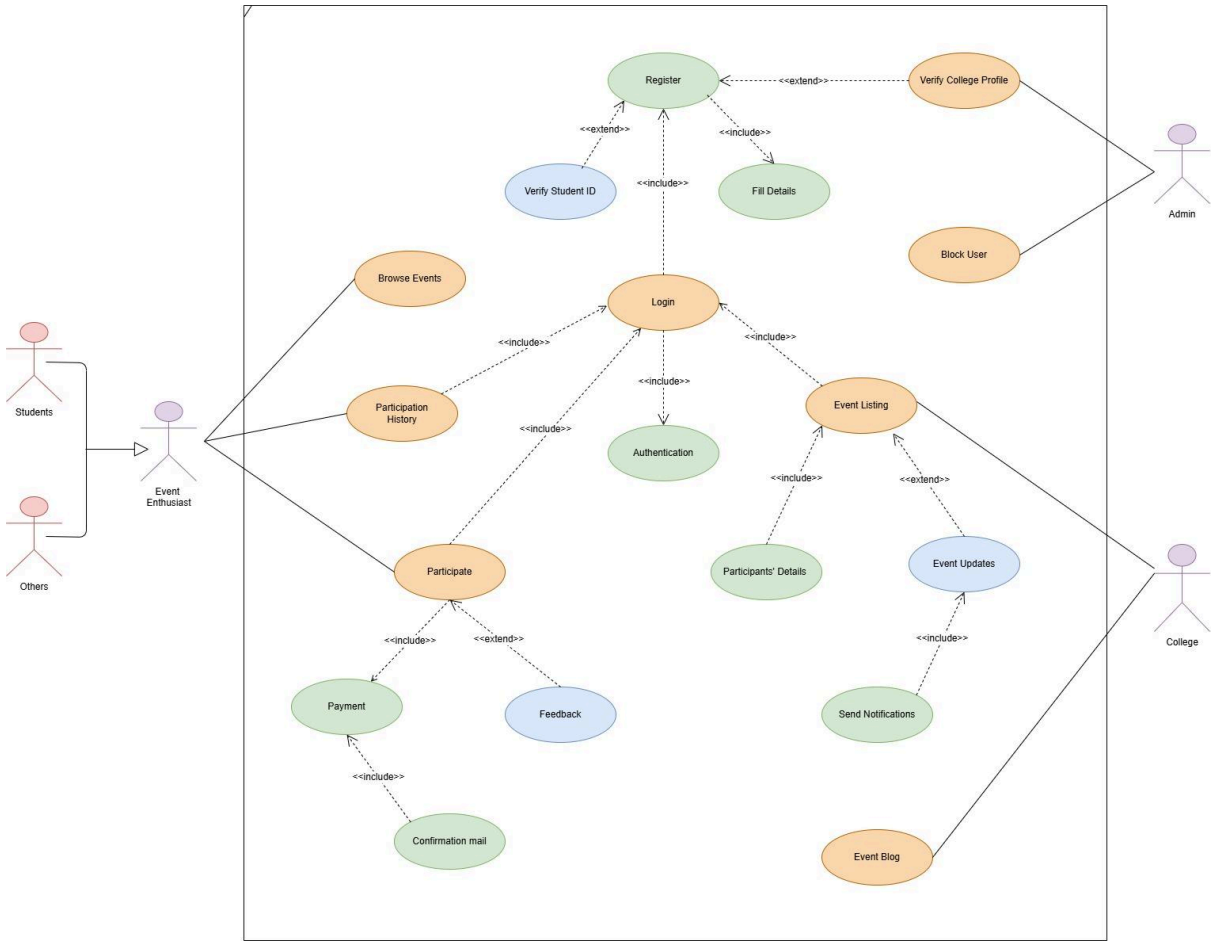
### 2.4.2 Non-Functional requirements

1. **Performance**: The system should provide fast response times and handle high traffic during events with minimal delays.
2. **Reliability**: Ensure consistent uptime and data accuracy with minimal downtime for maintenance.
3. **Scalability**:
    - The system must handle a rising volume of event submissions, ensuring smooth processes for submission, approval, and display of events.
    - The platform should efficiently manage an increasing number of user requests for event registration and attendance tracking, ensuring seamless user experience.
4. **Load Handling**: The system should maintain stability under increased loads, such as high numbers of simultaneous users and large transaction volumes.
5. **Usability**: Design an intuitive and user-friendly interface for seamless navigation across all user roles.
6. **Security**:
    - The system should implement secure user authentication mechanisms.
    - The system should enforce role-based access control.
7. **Data Privacy**: Ensure compliance with data privacy regulations and protect personal information of users.

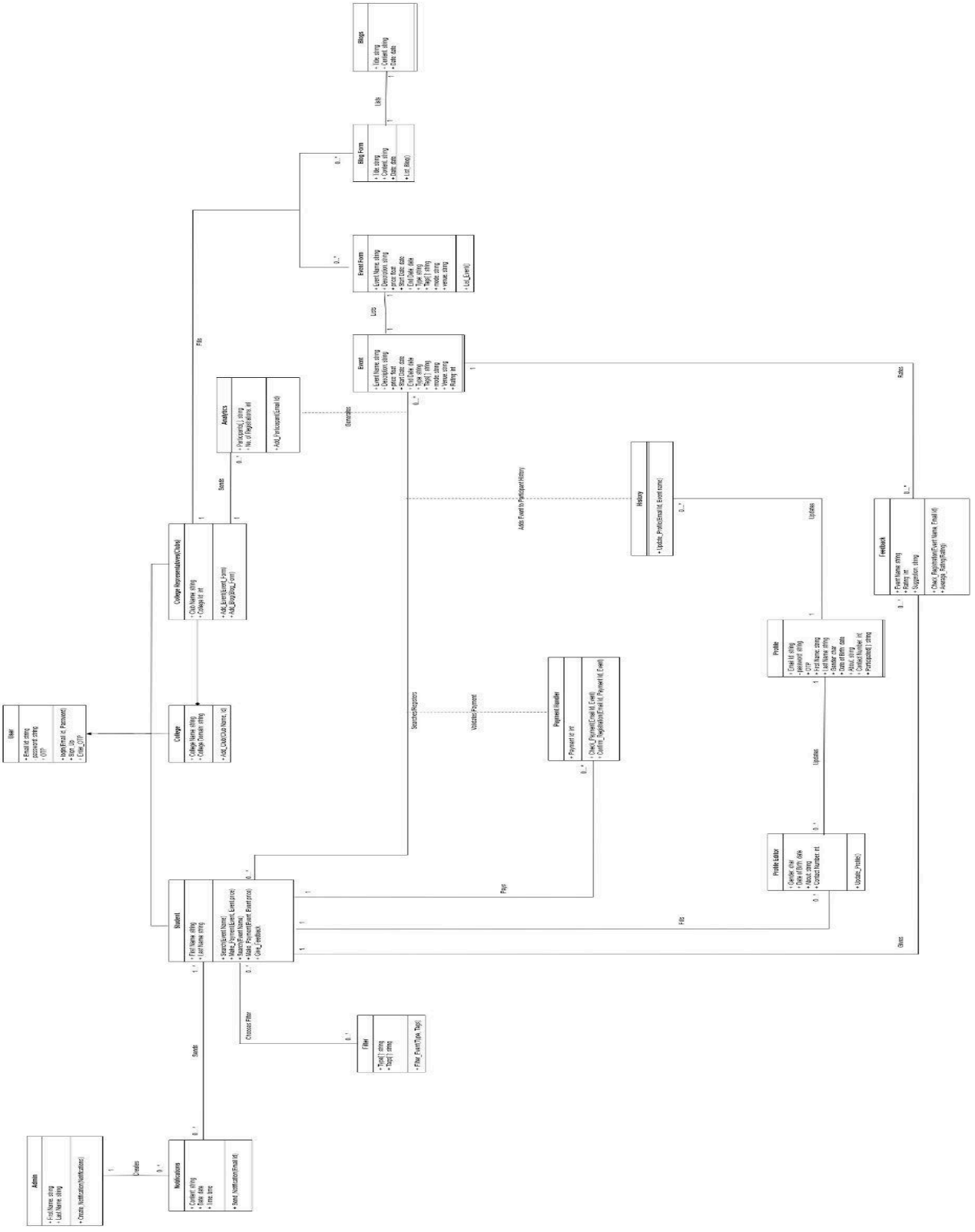
# Chapter 3

# System Behaviour

## 3.1 Use Case Diagram



# 3.2 Class Diagram



# Chapter 4

# Product Backlog

## 4.1 Sprint 1: Platform Registration (Duration - 2 weeks)

**Stories:**

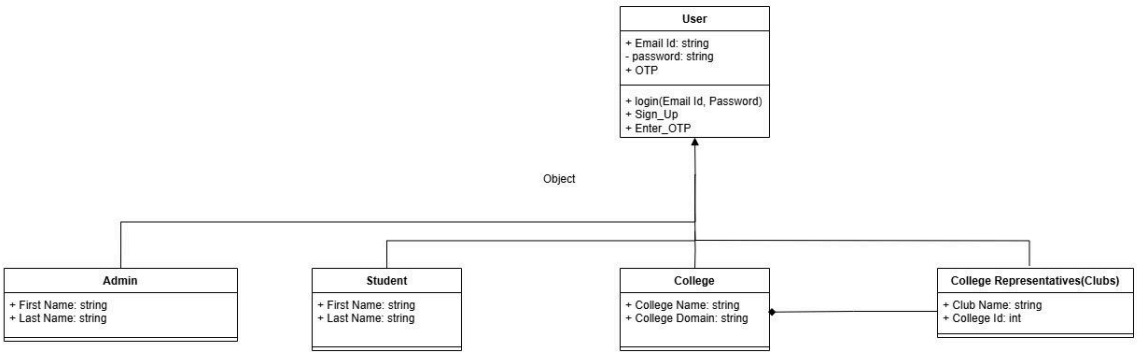
- Student Registration on Platform.
- College Registration on Platform.
- Login Functionality for Students, Colleges and clubs.

| Functionality | UFP | AFP | Total FP |
|---|---|---|---|
| Student Registration on Platform | EI = 1, EO = 1, EQ = 0, ILF = 1, EIF = 1 | 0.65 | 21 |
| CollegeRegistration on Platform | EI = 1, EO = 1, EQ = 0, ILF = 1, EIF = 1 | 0.65 | 21 |
| Login Functionality for Students, Colleges and clubs. | EI = 1, EO = 1, EQ = 1, ILF = 1, EIF = 1 | 0.65 | 24 |

Estimated Effort: 66FP

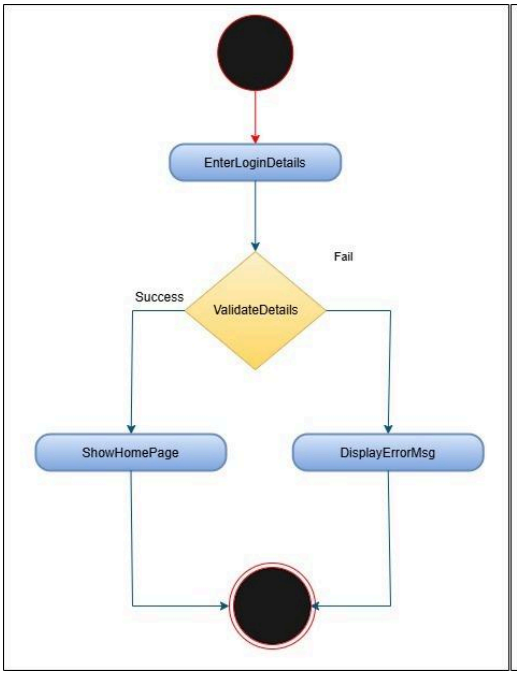
**Diagrams:**

- **Class diagram**

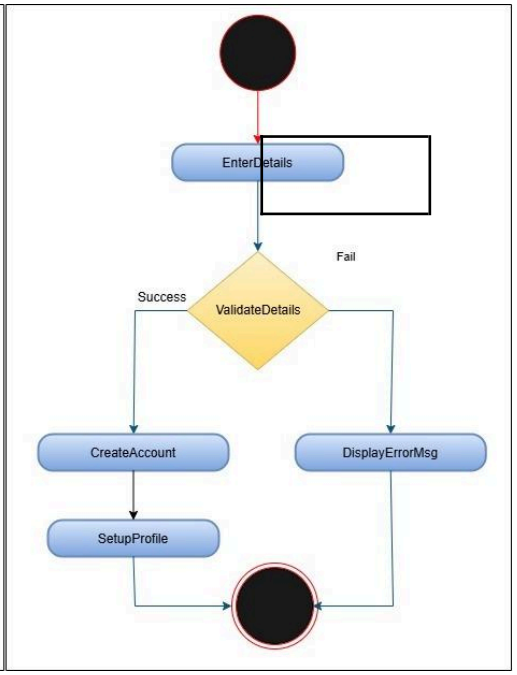


- **Activity Diagram**

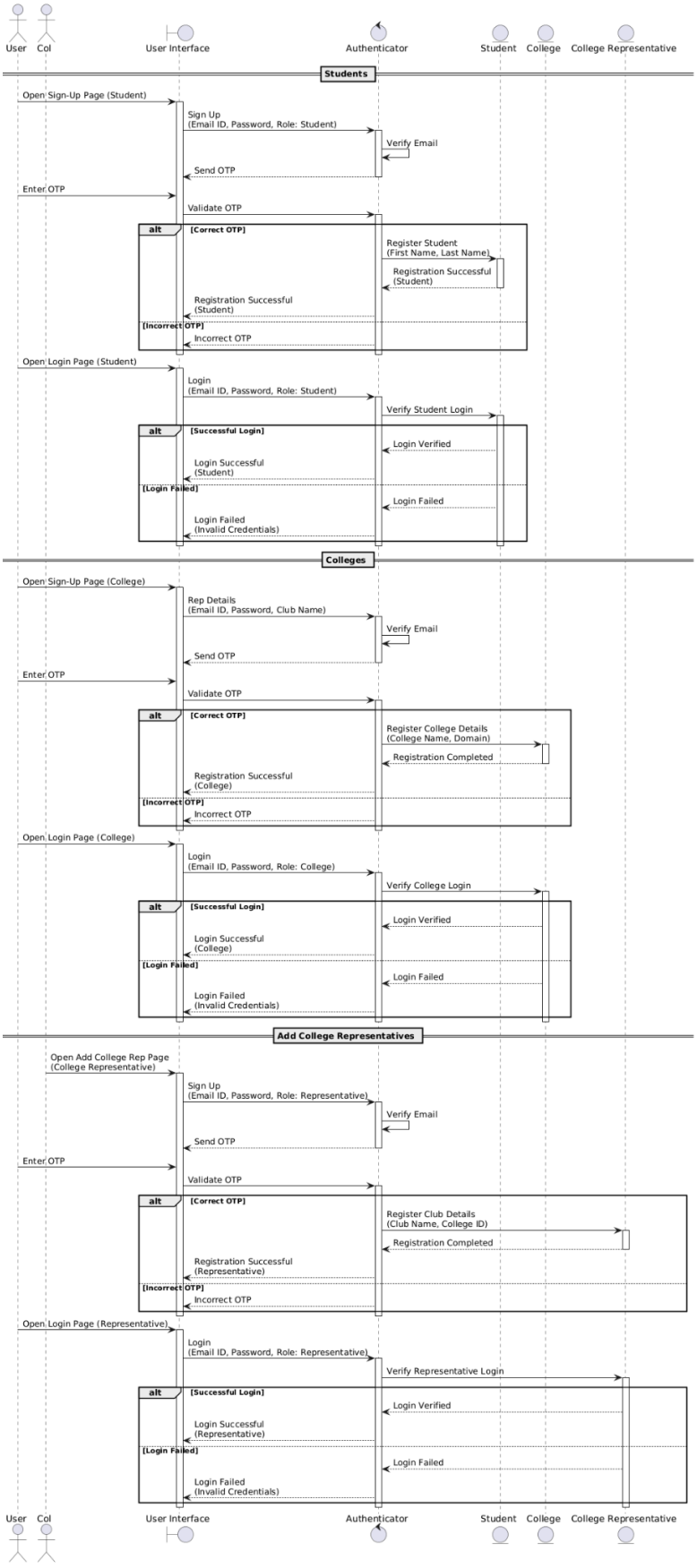
**Login :**




**SignUp :**



- **Sequence diagram**



## 4.2 Sprint 2: Event Listing & Updating (Duration - 2 weeks)

## Stories:

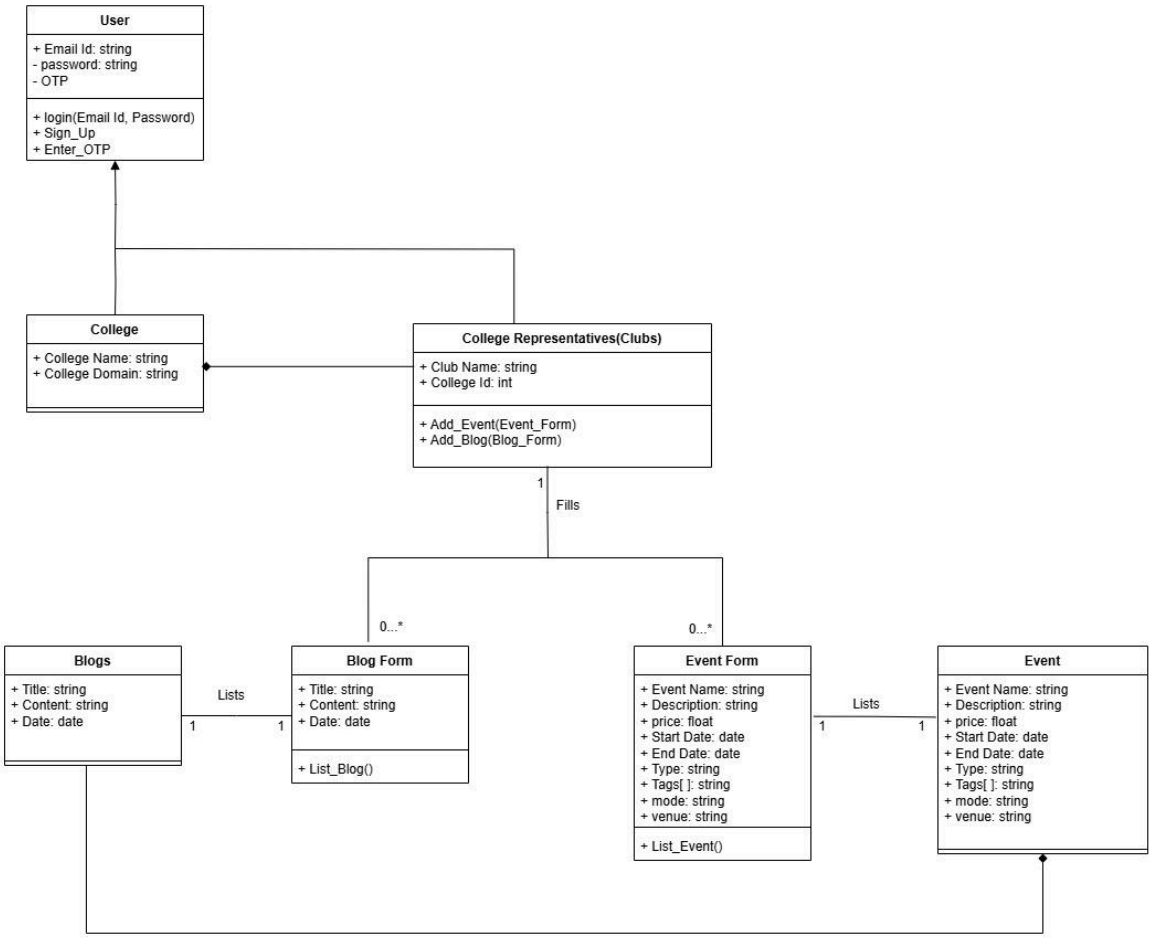
- Basic Event Listing by College Representatives
- Updating Event Details by College Representatives
- Creating Blog for Events by College Representatives

| Functionality | UFP | AFP | Total FP |
|---|---|---|---|
| Basic Event Listing by College Representatives | EI = 1, EO = 1, EQ = 0, ILF = 1, EIF = 0 | 0.65 | 14 |
| Updating Event Details by College Representatives | EI = 1, EO = 2, EQ = 0, ILF = 1, EIF = 1 | 0.65 | 26 |
| Creating Blog for Events by College Representatives | EI = 1, EO = 1, EQ = 0, ILF = 1, EIF = 0 | 0.65 | 16 |

Estimated Effort: 56FP

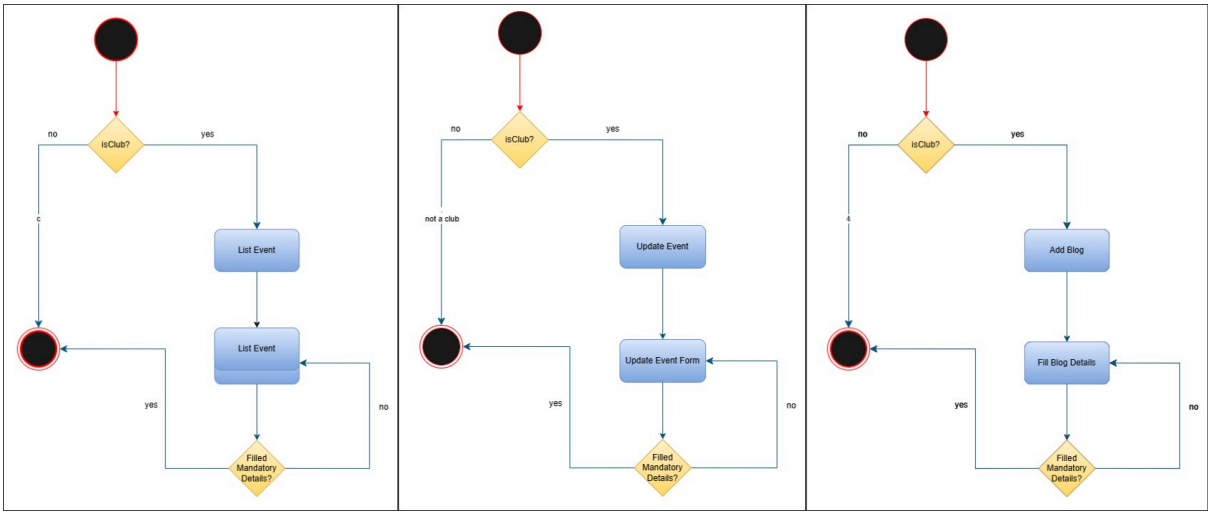
## Diagrams:

- **Class diagram**

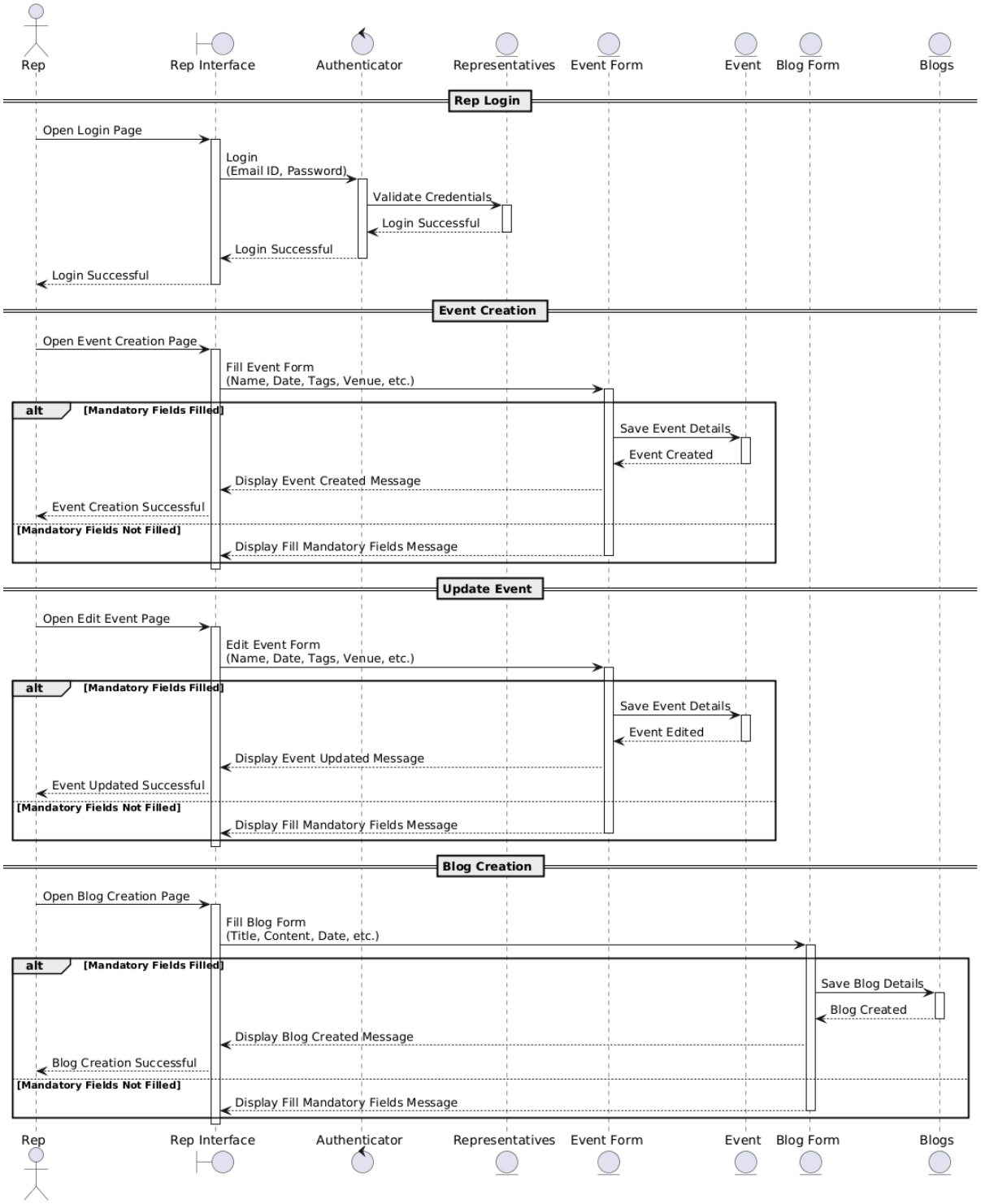



- **Activity Diagram**

List Event : Update Event : Add Blog :



- **Sequence diagram**



## 4.3 Sprint 3 : Advanced Event Management & Search (Duration - 3 weeks)

**Stories:**

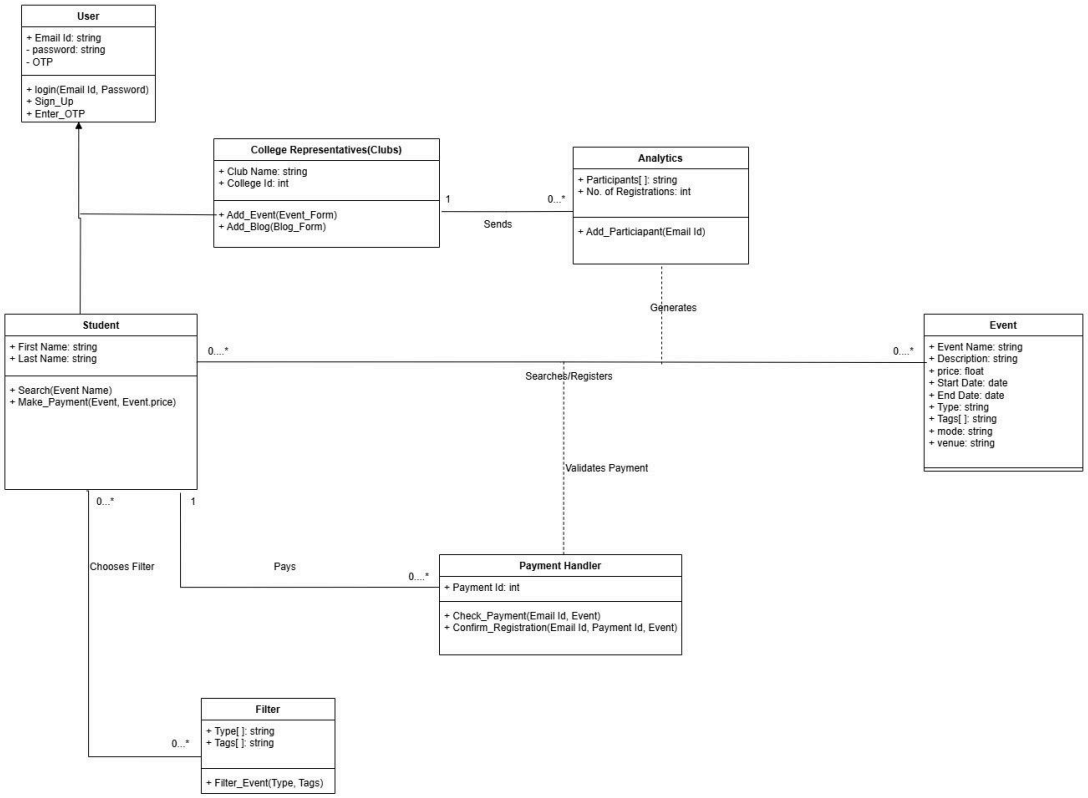
- Event Search Functionality
- Student Event Registration & Payment
- Event Filtering Options

| Functionality | UFP | AFP | Total FP |
|---|---|---|---|
| Event Search Functionality | EI=1,EO=1,EQ=1,ILF=1, EIF =0 | 0.65 | 21 |
| Student Event Registration & Payment | EI=2,EO=2,EQ=0,ILF=2, EIF=2 | 0.65 | 42 |
| Event Filtering Options | EI=1,EO=1,EQ=1,ILF=1, EIF =0 | 0.65 | 20 |

Estimated Effort: 83 FP

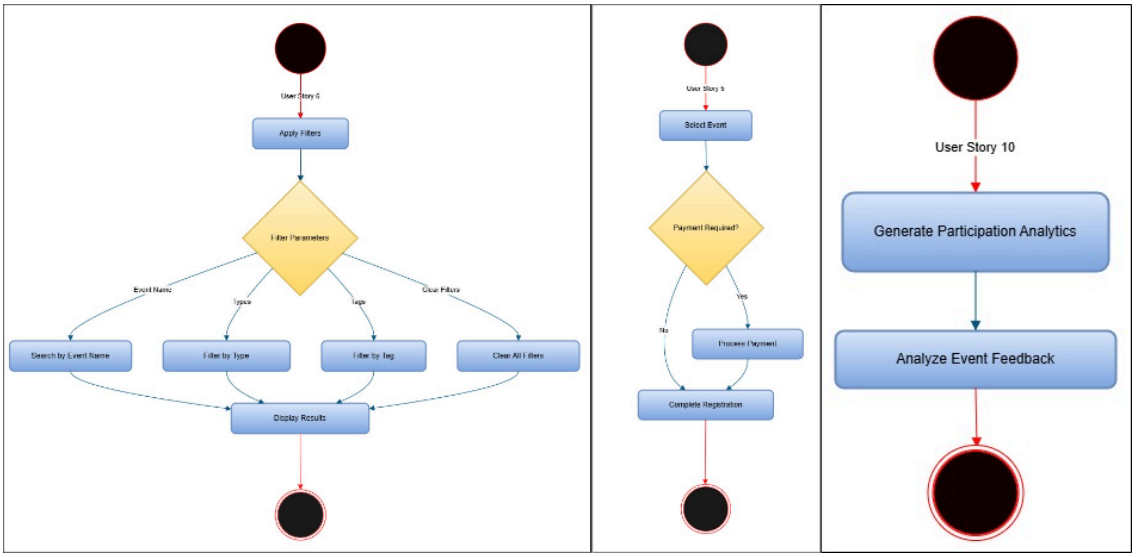
# Diagrams:

- ## Class diagram

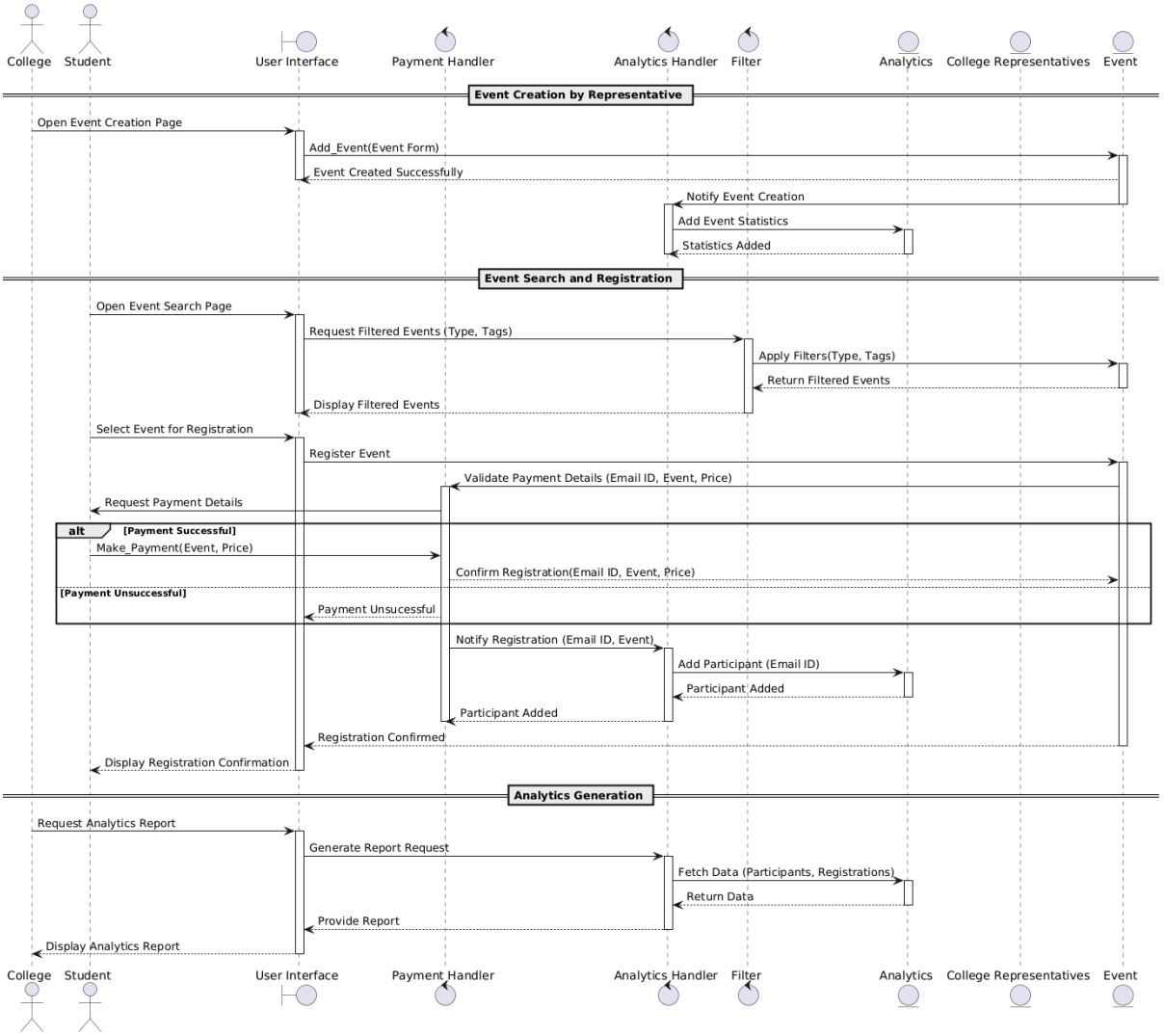



- ## Activity Diagram

Event Filter : Event Reg. : Event Analytics :



- **Sequence diagram**



## 4.4 Sprint 4 : Notifications, Feedback, and Profile Management (2 Weeks)

## Stories:

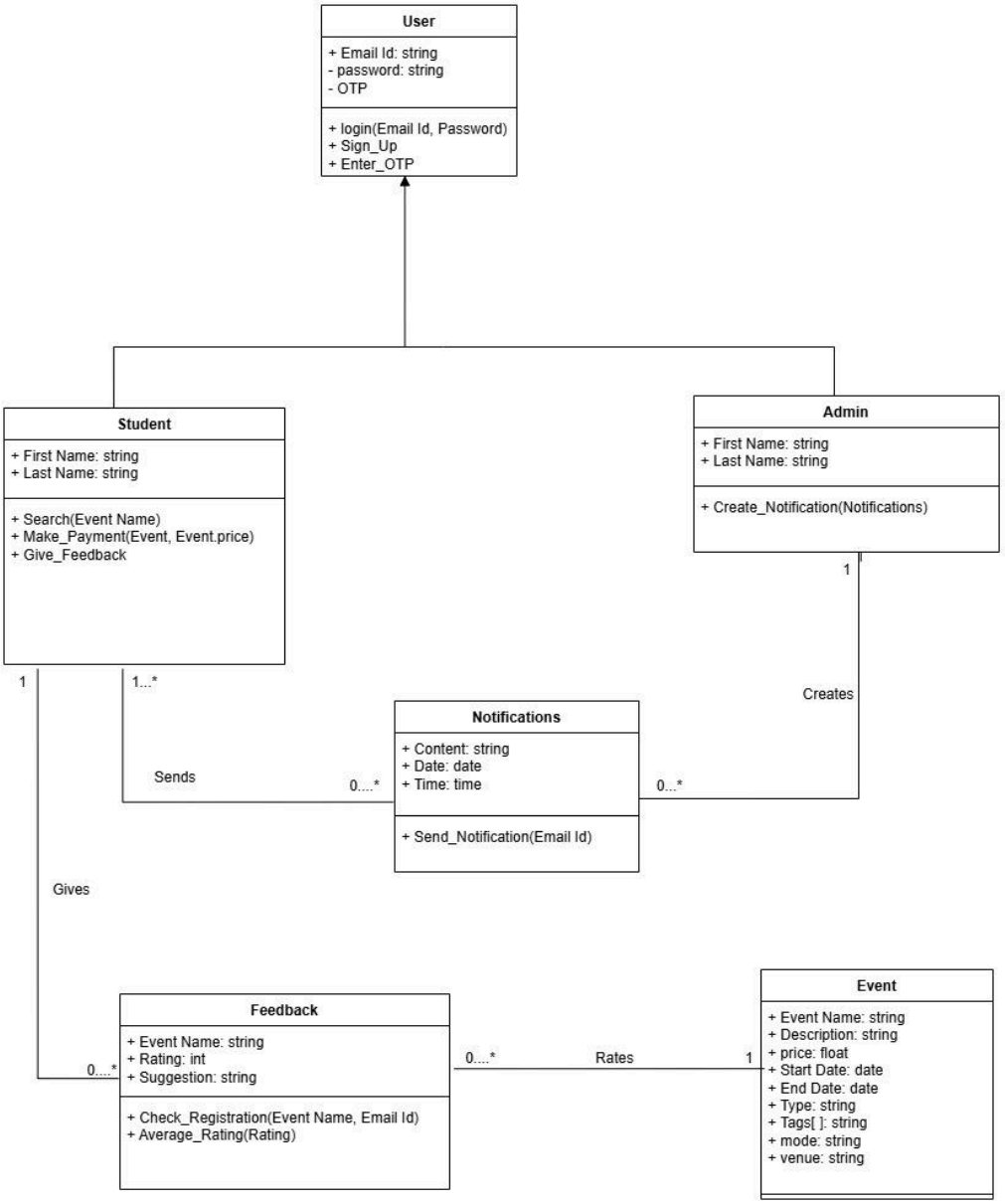
- Event Notifications and Reminders
- Feedback Submission Post Event
- Profile Management & Participation History
- Event Analytics for College Representatives

| Functionality | UFP | AFP | Total FP |
|---|---|---|---|
| Event Notifications and Reminders | EI=0,EO=1,EQ=0,ILF=1, EIF=0 | 0.65 | 15 |
| Feedback Submission Post Event | EI=1,EO=1,EQ=0,ILF=1, EIF=0 | 0.65 | 15 |
| Profile Management & Participation History | EI=1,EO=1,EQ=0,ILF=2, EIF=0 | 0.65 | 21 |
| Event Analytics for College Representatives | EI=0,EO=1,EQ=0,ILF=1, EIF=0 | 0.65 | 12 |

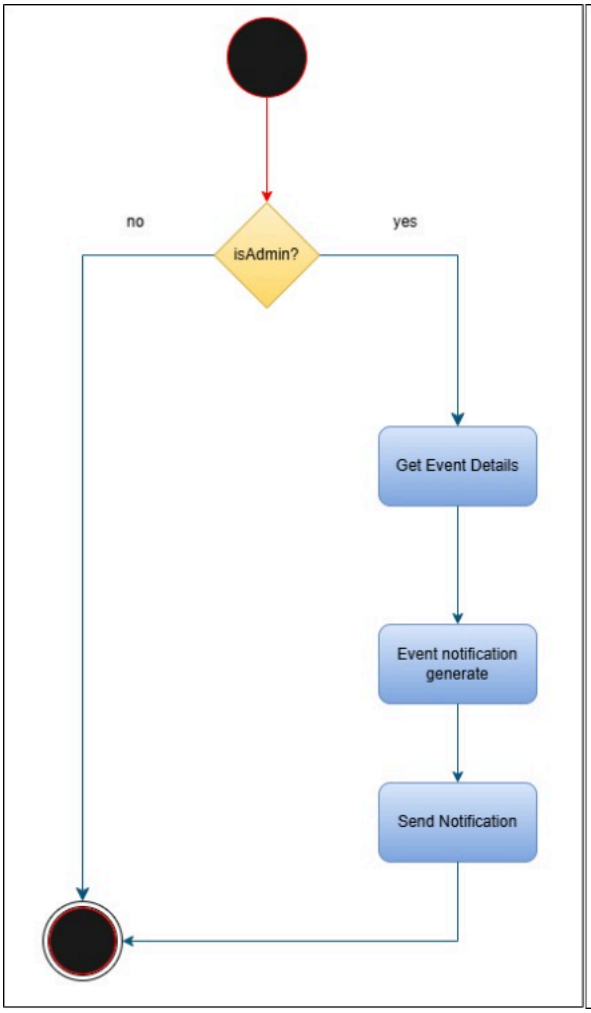
Estimated Effort: 63 FP

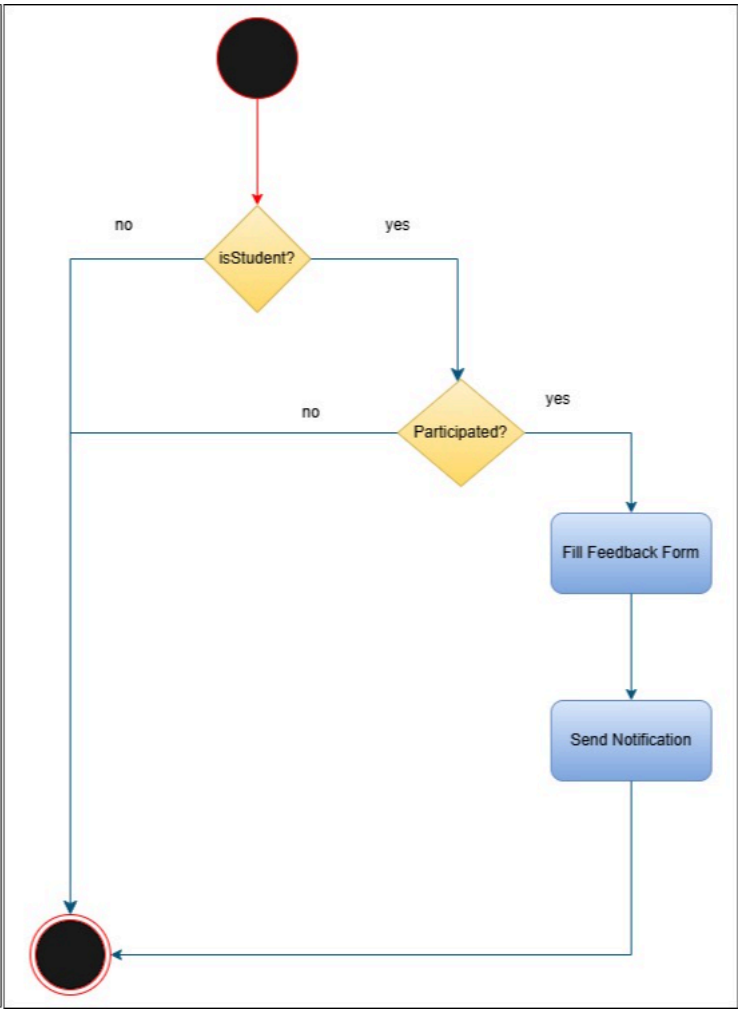
# Diagrams:

- **Class diagram**

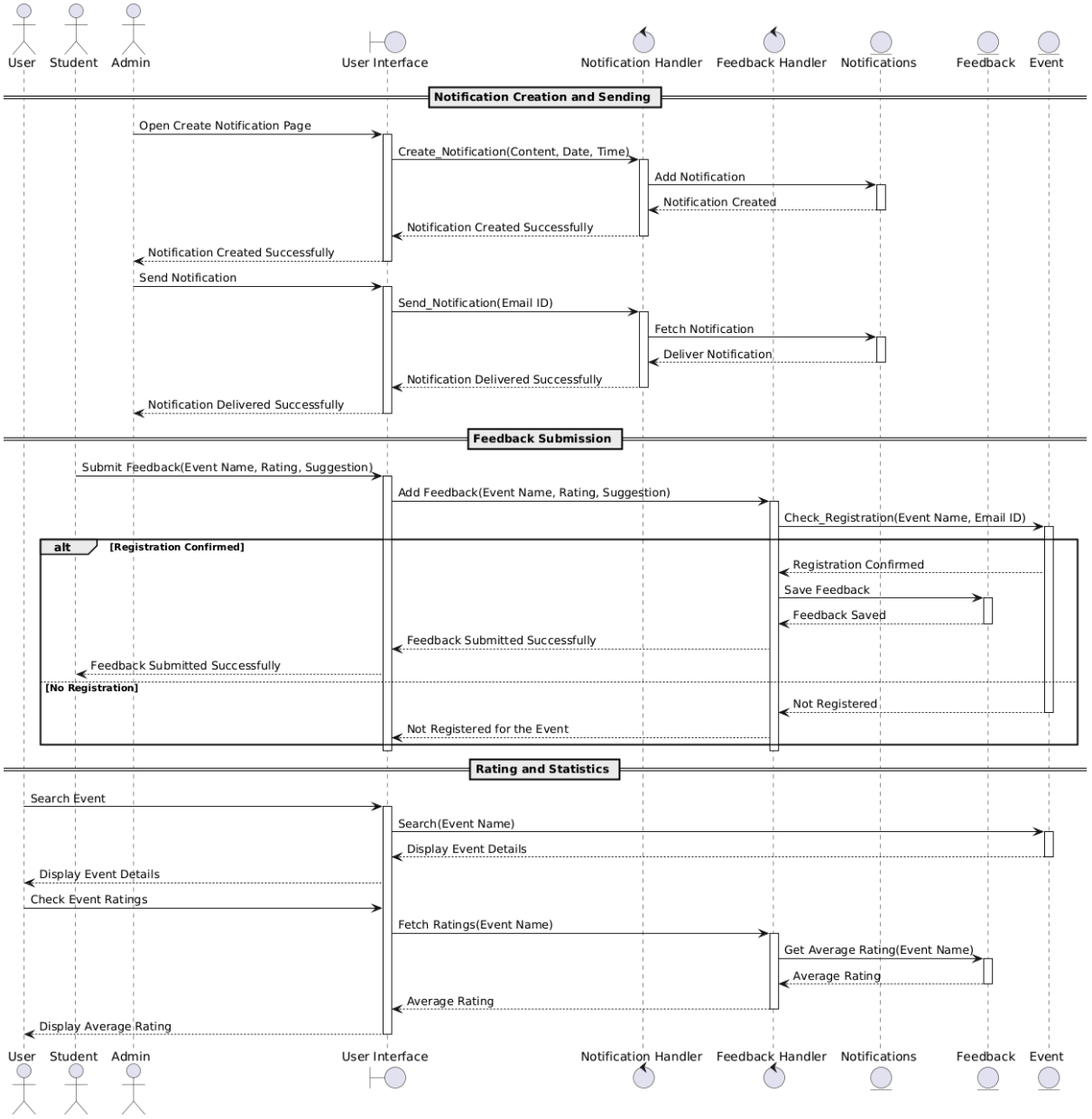


- **Activity Diagram**



- **Sequence diagram**



### 4.5 Sprint 5 : Security, Recommendations, and Admin Controls (3 Weeks)

**Stories:**

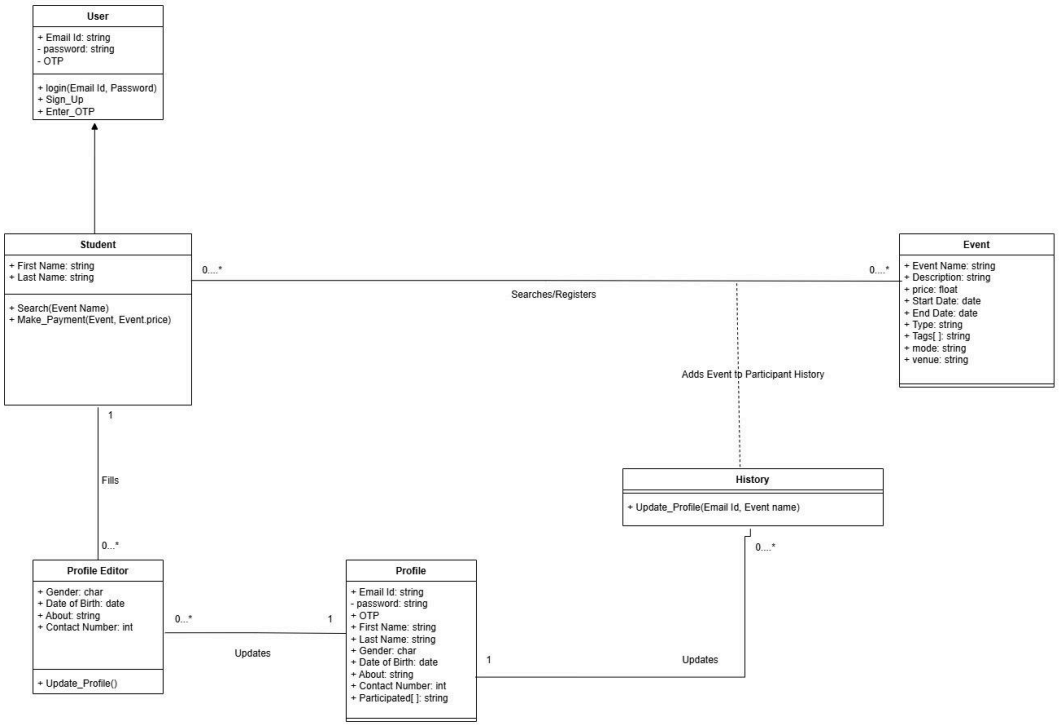
- Secure Payment Integration
- Personalized Event Recommendations
- Admin Moderation and Management
- Implement Two-Factor Authentication

| Functionality | UFP | AFP | Total FP |
|---|---|---|---|
| Secure Payment Integration | EI=1,EO=2,EQ=0,ILF=0, EIF=1 | 0.65 | 23 |
| Personalized Event Recommendations | EI=1,EO=1,EQ=0,ILF=2, EIF=0 | 0.65 | 25 |
| Admin Moderation and Management | EI=2,EO=1,EQ=0,ILF=2, EIF=0 | 0.65 | 27 |
| Implement Two-Factor Authentication | EI=2,EO=1,EQ=0,ILF=1, EIF=1 | 0.65 | 27 |

Estimated Effort: 102 FP

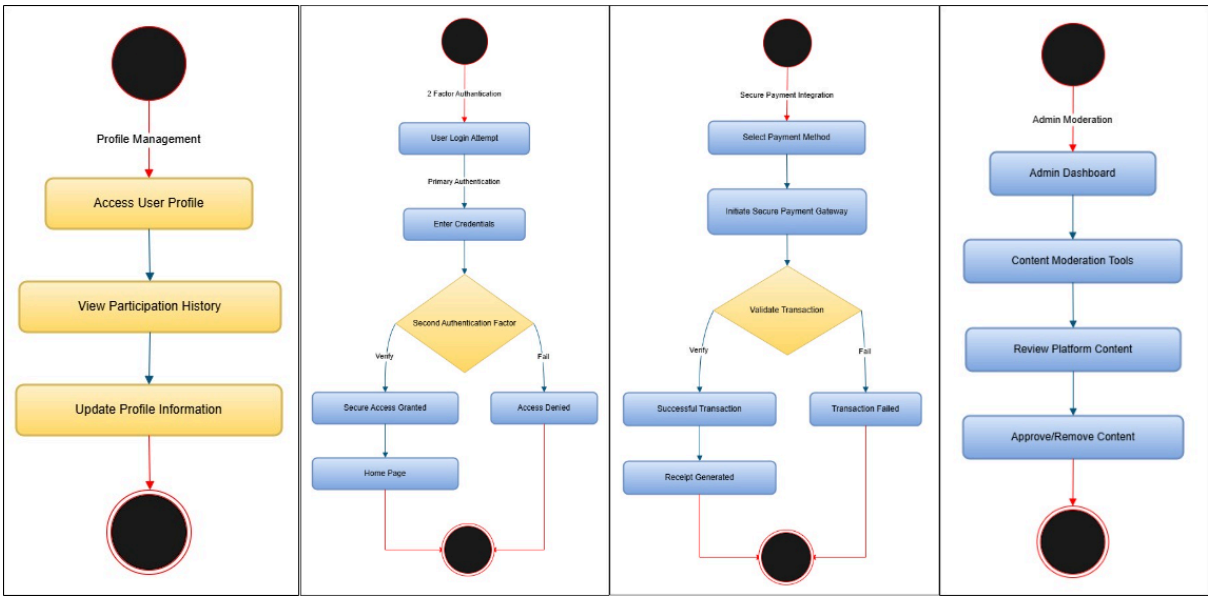
# Diagrams:

- ## Class diagram

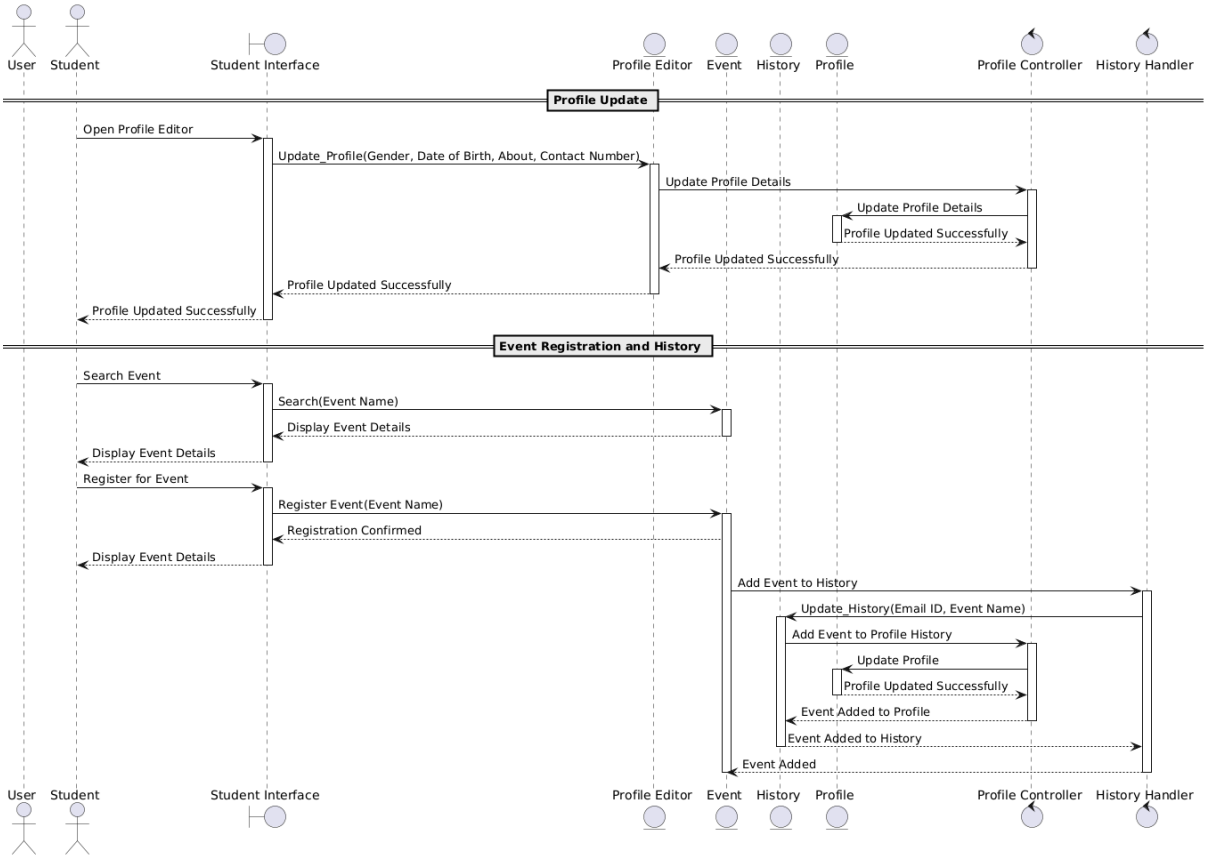



- ## Activity Diagram

Profile Manage : 2FA : Payment Integration : Admin :



- **Sequence diagram**



# Chapter 5

# Process Model for Development

## 5.1 Explanation of Process Model

The Agile Model was selected for the development of the Event Sphere platform. This decision ensures an iterative approach to design, development, and delivery, enabling continuous improvement based on stakeholder feedback. Each sprint is structured to deliver functional increments that align with user needs, promoting collaboration and adaptability.

## 5.2 Reason for Selection

The Agile Model is ideal for the Event Sphere platform due to its dynamic and evolving requirements. Features like event listing, student participation tracking, and admin controls require iterative development and constant feedback to achieve user satisfaction. The model supports:

- Rapid delivery of functional components.
- Continuous testing and integration for quality assurance.
- Stakeholder involvement to refine and prioritize features.

## 5.3 Implementation of Agile Process

### 5.3.1 Requirements Gathering

**Objective:** Collaborate with colleges, clubs, committees, and students to gather insights for defining platform requirements.

**Activities:**

- Conduct interviews with stakeholders to capture event management needs.
- Create initial wireframes to validate core functionalities like user registration and event search.
- Use surveys to understand desired features such as notifications and personalized recommendations.

**Output**: A validated and comprehensive requirement list, prioritizing features like registration, event cataloging, and blog creation.

### 5.3.2 Sprint Planning

**Objective:** Break down requirements into actionable tasks and organize them into sprints.

**Activities:**

- Divide requirements into user stories (e.g., student registration, event updates, analytics).
- Prioritize user stories based on impact and dependencies.
- Establish sprint goals for delivering functional increments (e.g., event filtering, notification system).

**Output**: A sprint backlog with clearly defined tasks, milestones, and objectives.

### 5.3.3 Development and Testing

**Objective:** Develop platform features and ensure quality through iterative testing.

**Activities:**

- Develop core functionalities for each sprint, such as profile management and event payment integration.
- Perform unit tests to validate individual components.
- Conduct testing to ensure seamless interaction between modules.
- Use automated tools to identify and fix bugs early in the development process.

**Output**: Incrementally developed and tested features, ensuring a high-quality platform.

## 5.3.4 Sprint Review

**Objective:** Present sprint outcomes to stakeholders, gather feedback, and identify areas for improvement.

**Activities:**

- Showcase completed features like event search, analytics, and blog creation.
- Collect feedback on usability and functionality.
- Make adjustments to address user needs and enhance feature performance.

**Output**: Actionable feedback to refine the platform and guide subsequent sprints.

### 5.3.5 Sprint Retrospective

**Objective:** Reflect on team performance and processes to improve efficiency in future sprints.

**Activities:**

- Discuss successes and challenges encountered during the sprint.
- Identify areas for improvement, such as task allocation or communication tools.
- Develop an action plan for enhanced productivity in the next sprint cycle.

**Output**: Insights and action items to streamline future development cycles.

### 5.3.6 Continuous Iteration

**Objective:** Use stakeholder feedback and retrospective insights to refine the platform until all goals are met.

**Activities:**

- Apply lessons learned to subsequent sprints for better outcomes.
- Integrate additional features or enhancements based on emerging needs.
- Test and deliver a robust, user-friendly platform.

**Output**: A feature-complete Event Sphere platform with optimized usability, scalability, and performance.

# Chapter 6

# Technology Used

**We have used Mern Stack to develop this project.**

## 5.1 Frontend

**Technologies**: ReactJS, CSS, SCSS, Tailwind CSS

## 5.2 Backend

**Technologies**: node.js, express.js framework

## 5.3 Database

**Technologies**: mongoDB, mongoose

## 5.4 Development Tools

**Technologies**: Visual Studio Code (VSCode), GitHub

# Chapter 7

# Bugs in our Project

There are several such cases where a bug is found in our development.

1. HomePage Slider button (Need to refresh page for activating)
2. Blog detail slider for change image
3. List Event button not show without refresh
4. Rarely verify as student - OTP not verify

# Chapter 8

# Part Not Implemented

As we were learning each and every tech stack from scratch and did not have any prior learnings, we could not implement the following discussed functionalities in our project.

1. Secure Payment Integration
2. Event Notifications and Reminder
3. Feedback Submission
4. Personalized Event Recommendation
5. System Responsiveness
6. Security of User Data

# Chapter 9

# References


1. ChatGPT: https://chatgpt.com
2. Early AI: https://www.startearly.ai