

# Software Engineering (Sessional) Report

Course: CSE-434



---

## CUET Sports and Cultural Event Management System(WebApp)

---

### Team Members

---

Tahsin Azad Tias  
ID: 1804009

Md. Shahriar Ahmed  
ID: 1804020

Jamiul Hasan Joy  
ID: 1804028

## Team Members

Photo	Name	ID	Email	Contact NO.	Total Credit Passed	CGPA acquired
	Tahsin Azad Tias	1804009	tahsintias999@gmail.com	01942093490	140.75	3.74
	Md. Shahriar Ahmed	1804020	shahriarbadhon778@gmail.com	01746180312	140.75	3.25
	Jamiul Hasan Joy	1804028	jamiujoyhasan@gmail.com	01992917009	140.75	3.36

# **Executive Summary**

The CUET Sports and Cultural Event Management System marks a significant advancement in the organization and coordination of sports and cultural activities at Chittagong University of Engineering and Technology (CUET). Comprising two pivotal components, this web-based application addresses critical challenges in event management within the campus environment. The Event Organization module provides event organizers with a comprehensive platform to seamlessly plan, schedule, and manage sports and cultural events. From event creation to participant registration and communication, this module streamlines the entire event organization process, enhancing efficiency and participant engagement. Simultaneously, the Participant Experience component enhances the experience of event participants by providing them with intuitive interfaces for event registration, communication, and feedback submission. With real-time updates and user-friendly interfaces, the system promotes active participation and fosters a vibrant campus community. By leveraging technology to optimize event management processes, the CUET Sports and Cultural Event Management System exemplifies CUET's commitment to innovation and excellence in campus activities.

# Contents

<b>Executive Summary</b>	<b>1</b>
<b>List of Figures</b>	<b>7</b>
<b>List of Tables</b>	<b>9</b>
<b>1 Introduction</b>	<b>10</b>
1.1 Goals and Objectives of the project . . . . .	10
1.2 Scope of the work . . . . .	10
1.2.1 Current Situation and Context . . . . .	10
1.2.2 Competing Products (Available in Market) . . . . .	11
1.3 System overview . . . . .	11
1.4 Structure of the document . . . . .	11
1.5 Terms, Acronyms, and Abbreviations Used . . . . .	12
<b>2 Project Management Plan</b>	<b>13</b>
2.1 Project Organization . . . . .	13
2.1.1 Individual Contribution to the project . . . . .	13
2.2 Process Model Used . . . . .	13
2.2.1 Rationale for Choosing Prototyping Process Model for CUET Sports and Cultural Event Management System .	15
2.3 Risk Analysis . . . . .	16
2.4 Constraints to project implementation . . . . .	18
2.5 Hardware and Software Resource (Tools/Language) Requirements . . . . .	19
2.5.1 Software Requirements . . . . .	19

2.5.2	Hardware Requirements . . . . .	20
2.6	Project Timeline and Schedule . . . . .	20
2.7	Estimated Budget . . . . .	20
2.8	Social/Cultural/Environmental impact of the project . . . . .	22

### **3 Requirement Specifications 23**

3.1	Software Requirements Specification (SRS) of User Authentication Module of CUET Sports and Cultural Event Management System . . . . .	23
3.1.1	Introduction . . . . .	23
3.1.2	Overall Description . . . . .	23
3.1.3	Functional Requirements . . . . .	24
3.1.4	Pages . . . . .	25
3.1.5	Non-functional Requirements . . . . .	25
3.1.6	Constraints . . . . .	26
3.1.7	Assumptions and Dependencies . . . . .	26
3.1.8	Future Enhancements . . . . .	26
3.1.9	Conclusion . . . . .	26
3.2	Software Requirements Specification (SRS) of Event Management Module of CUET Sports and Cultural Event Management System . . . . .	27
3.2.1	Introduction . . . . .	27
3.2.2	Overall Description . . . . .	27
3.2.3	Functional Requirements . . . . .	27
3.2.4	Non-functional Requirements . . . . .	28
3.2.5	Constraints . . . . .	28
3.2.6	Assumptions and Dependencies . . . . .	28
3.2.7	Future Enhancements . . . . .	29
3.2.8	Conclusion . . . . .	29

3.3 Software Requirements Specification (SRS) of System Insight Module of CUET Sports and Cultural Event Management System . . . . .	29
3.3.1 Introduction . . . . .	29
3.3.2 Overall Description . . . . .	29
3.3.3 Functional Requirements . . . . .	30
3.3.4 Non-functional Requirements . . . . .	30
3.3.5 Constraints . . . . .	30
3.3.6 Assumptions and Dependencies . . . . .	31
3.3.7 Future Enhancements . . . . .	31
3.3.8 Conclusion . . . . .	31
3.4 Stakeholders for the system . . . . .	31
3.4.1 Users . . . . .	31
3.4.2 Developers . . . . .	31
3.5 Use case diagram with Graphical and Textual Description	31
3.6 Activity Diagram . . . . .	37
3.7 Swimlane Diagram . . . . .	39
3.8 Class diagram . . . . .	43
3.9 Sequence diagram . . . . .	44
3.10 State Diagram . . . . .	46
3.11 Safety and Security requirements . . . . .	49
3.11.1 Access Requirements . . . . .	49
3.11.2 Integrity Requirements . . . . .	49
3.11.3 Privacy Requirements . . . . .	49
<b>4 Architecture</b>	<b>50</b>
4.1 Architectural model/style used . . . . .	50
4.1.1 Rationale for choosing our architectural model/style . . . . .	52
4.2 Technology, software, and hardware used . . . . .	53

<b>5 Design</b>	<b>54</b>
5.1 Component level design following pattern . . . . .	54
5.2 Dataflow Diagram . . . . .	57
5.2.1 First Level DFD . . . . .	57
5.2.2 Admin DFD . . . . .	58
5.2.3 User DFD . . . . .	58
5.2.4 Entity Relationship Diagram . . . . .	59
5.3 API Documentation . . . . .	61
5.3.1 User Authentication . . . . .	61
5.3.2 Event Management . . . . .	63
5.4 GUI (Graphical User Interface) design . . . . .	64
5.4.1 Homepage . . . . .	64
5.4.2 List of upcoming events . . . . .	64
5.4.3 Create Events . . . . .	65
5.4.4 Event creation form . . . . .	65
5.4.5 Feedback Add . . . . .	66
5.4.6 Login . . . . .	66
5.4.7 Signup . . . . .	67
<b>6 Testing and sustainability plan</b>	<b>68</b>
6.1 Requirements/specifications-based system level test cases . . .	68
6.2 Traceability of test cases to use cases . . . . .	75
6.3 Techniques used for test generation . . . . .	75
6.4 Assessment of the goodness of your test suite . . . . .	75
6.5 Sustainability Plan . . . . .	76
6.5.1 Scalability . . . . .	76
6.5.2 Flexibility / Customization . . . . .	76
<b>Acknowledgement</b>	<b>76</b>



# List of Figures

1.1	System Overview . . . . .	11
2.1	Prototyping Process Model . . . . .	15
2.2	Project Gantt Chart . . . . .	20
3.1	Use Case Diagram for User Authentication Module . . . . .	32
3.2	Use Case Diagram for Event Management Module . . . . .	33
3.3	Use Case Diagram for System Insight Module . . . . .	33
3.4	Activity Diagram for User Authentication . . . . .	37
3.5	Activity Diagram for Event Management . . . . .	38
3.6	Activity Diagram for System Insight . . . . .	39
3.7	Swimlane Diagram for User Authentication . . . . .	40
3.8	Swimlane Diagram for Event Management . . . . .	41
3.9	Swimlane Diagram for System Insight . . . . .	42
3.10	Class Diagram for the proposed system . . . . .	43
3.11	Sequence Diagram for User Authentication . . . . .	44
3.12	Sequence Diagram for Event Management . . . . .	45
3.13	Sequence Diagram for System Insight . . . . .	45
3.14	State Diagram for User Authentication . . . . .	46
3.15	State Diagram for Event Management . . . . .	47
3.16	State Diagram for System Insight . . . . .	48
4.1	Data-centered architecture for the proposed system . . . . .	51
4.2	MVC architecture for the proposed system . . . . .	52
5.1	Structure chart of CUET Sports and Cultural Management System . . . . .	54
5.2	Component level design of User Authentication Module . . . . .	55

5.3	Component level design of Event Management Module . . . . .	56
5.4	Component level design of Feedback Management Module . . .	57
5.5	First Level DFD . . . . .	58
5.6	Admin DFD . . . . .	58
5.7	User DFD . . . . .	59
5.8	Relationships between different entities . . . . .	60
5.9	Entity Relationship Diagram of CUET Sports and Cultural Event Management System . . . . .	60
5.10	Request and Response if registration successful . . . . .	61
5.11	Request and Response if registration unsuccessful . . . . .	61
5.12	Request and Response if login successful . . . . .	62
5.13	Request and Response if login unsuccessful . . . . .	62
5.14	Request and Response for event post . . . . .	63
5.15	Request and Response for event edit . . . . .	63
5.16	Homepage . . . . .	64
5.17	List of upcoming events . . . . .	64
5.18	Create Events . . . . .	65
5.19	Event Creation Form . . . . .	65
5.20	Add Feedback . . . . .	66
5.21	Login . . . . .	66
5.22	SignUp . . . . .	67

# List of Tables

2.1	Individual Contribution to the project . . . . .	13
2.2	Risk Analysis for CUET Sports and Cultural Event Management System . . . . .	17
2.3	Team Size Outlook . . . . .	21
2.4	Estimated Cost Distribution for CUET Sports and Cultural Event Management System . . . . .	21
6.1	Test Cases for CUET Sports and Cultural Event Management System . . . . .	69
6.2	Requirements/specifications-based system level test cases (02)	70
6.3	Requirements/specifications-based system level test cases (03)	71
6.4	Requirements/specifications-based system level test cases (04)	72
6.5	Requirements/specifications-based system level test cases (05)	73
6.6	Requirements/specifications-based system level test cases (06)	74
6.7	Traceability of test cases to use cases . . . . .	75

# **Introduction**

In the rapidly evolving landscape of educational institutions, the Chittagong University of Engineering and Technology (CUET) recognizes the critical importance of ensuring the safety and security of its campus, students, faculty, and assets. To address these pressing needs, we introduce the "CUET Sports and Cultural Event Management System," a sophisticated software engineering endeavor designed to revolutionize the management of sports and cultural events within the CUET campus. Leveraging cutting-edge technologies and best practices in software engineering, this system aims to streamline event organization, enhance participant engagement, and elevate the overall experience of sports and cultural activities at CUET.

## **1.1 Goals and Objectives of the project**

The primary goal of the project is to provide CUET with a comprehensive platform for efficiently managing all aspects of sports and cultural events on campus. This includes simplifying event organization processes, facilitating participant registration and communication, and ensuring seamless coordination between event organizers and participants. Additionally, the system aims to promote greater student engagement in sports and cultural activities, fostering a vibrant campus community and enhancing the overall student experience.

## **1.2 Scope of the work**

The CUET Sports and Cultural Event Management System encompasses a wide range of functionalities to meet the diverse needs of event organizers and participants. From event creation and scheduling to participant registration and communication, the system provides a centralized platform for managing all aspects of sports and cultural events at CUET. By leveraging advanced technologies, the system aims to streamline event organization processes, improve participant engagement, and enhance the overall efficiency and effectiveness of sports and cultural activities on campus.

### **1.2.1 Current Situation and Context**

This subsection provides an in-depth analysis of the current situation and context surrounding sports and cultural event management at CUET. It explores existing challenges,

inefficiencies, and limitations faced by event organizers and participants within the campus community. By identifying key pain points and areas for improvement, this analysis sets the stage for the development and implementation of the CUET Sports and Cultural Event Management System.

### 1.2.2 Competing Products (Available in Market)

In this subsection, an assessment of competing products available in the market is conducted to understand the landscape of sports and cultural event management solutions. Various software platforms, applications, and tools designed for event organization and coordination are evaluated based on their features, functionalities, user experience, and adoption within similar educational institutions. This analysis helps in identifying potential competitors, benchmarking against industry standards, and identifying opportunities for differentiation and innovation in the CUET Sports and Cultural Event Management System.

## 1.3 System overview

The system features a user-friendly web-based interface that allows users to easily navigate and access the various functionalities offered by the platform. With intuitive design and seamless integration, the system provides event organizers and participants with a convenient and efficient way to plan, coordinate, and participate in sports and cultural events at CUET. Users can create new accounts, register for events, and communicate with event organizers seamlessly, enhancing the overall experience of sports and cultural activities on campus.

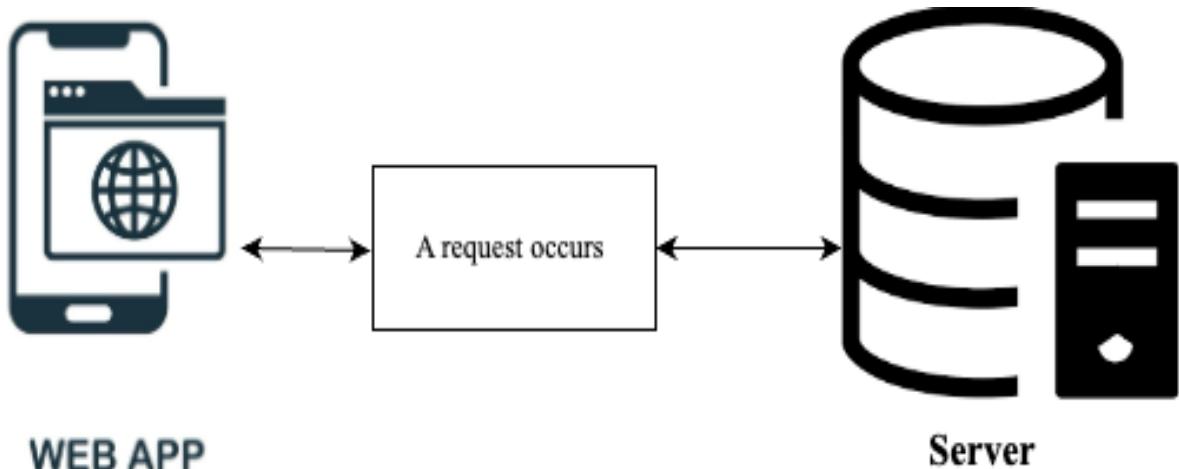


Figure 1.1: System Overview

## 1.4 Structure of the document

The entire document is divided into 6 chapters, these are given below:

1. **Chapter 1:** The goal of this chapter is to provide a thorough overview of the project. It contains an overview, the objectives and scope of the project, and a glossary of key terms.
2. **Chapter 2:** We go into detail about our project management approach for the app in this chapter. The project's social, cultural, and environmental effects are included, along with individual contributions, the selected process model, project risk analysis, implementation restrictions, necessary hardware and software resources, project timetable, schedule, and expected budget.
3. **Chapter 3:** Here we present a comprehensive overview of the requirement description for the project. The chapter covers an identification of stakeholders, activity diagrams, class diagrams, sequence diagrams, use case diagrams with textual and graphical explanations, and safety and security requirements.
4. **Chapter 4:** The architecture of the project is covered in this chapter. A detailed explanation is provided for the hardware, software, and technologies employed in the architecture.
5. **Chapter 5:** The architecture of the system is described in this chapter, including features of GUI and component-level design.
6. **Chapter 6:** This chapter covers system-level test cases based on requirements, test case traceability to use cases, test generation techniques, and the assessment of our test suite's quality, with a focus on our project's testing strategy and sustainability plan.  
Additionally, it explores the project's scope, goals, and sustainability strategy. References and acknowledgments are included at the end of the document.

## 1.5 Terms, Acronyms, and Abbreviations Used

Events, Stakeholders

# Project Management Plan

## 2.1 Project Organization

### 2.1.1 Individual Contribution to the project

Individual Contributions to the project are shown in Table-2.1

Member Name	Requirement Specification	Planning	Designing	Front End	Back End	Testing	Deployment
Tahsin Azad Tias	✓	✓	✓	✓		✓	
Md. Shahriar Ahmed	✓	✓	✓		✓		✓
Jamiul Hasan Joy	✓	✓				✓	

Table 2.1: Individual Contribution to the project

## 2.2 Process Model Used

For the development of the CUET Sports and Cultural Event Management System, our selected methodology is the Prototyping Approach. This iterative software development model involves creating an initial version, or prototype, to enhance our understanding and refine user requirements throughout the development stages. The key phases of the Prototyping Approach for our project are as follows:

### Initiation and Collaborative Definition:

Engaging stakeholders in open discussions to establish project objectives and gather preliminary requirements for the CUET Sports and Cultural Event Management System. Identification of critical features and functionalities through collaborative communication.

## **Quick Project Planning:**

Swiftly formulating a high-level plan that outlines crucial activities, necessary resources, and project timelines. Defining critical milestones and dependencies to ensure successful implementation.

## **Rapid Design Modeling:**

Creating a swift design that focuses on the visual aspects of the software. This may involve the generation of wireframes or mockups to highlight user interface elements. Visualization of the user journey guarantees alignment with stakeholder expectations and user experience.

## **Prototype Construction:**

Developing the actual prototype based on the quick design, utilizing existing program fragments or rapid development tools. Implementing core functionalities to provide a tangible representation of the CUET Sports and Cultural Event Management System.

## **Deployment, Distribution, and Feedback Collection:**

Deploying the constructed prototype to stakeholders, including users and organizers, for evaluation and interaction. Actively seeking feedback on usability, functionality, and overall user experience to facilitate continuous improvement.

## **Iterative Communication and Adaptation:**

Continuously revisiting the communication stage to incorporate valuable feedback obtained during deployment. Regular communication with stakeholders to adapt to evolving requirements and ensure alignment with the changing landscape.

Our choice of the Prototyping Process Model in Figure 5.22 for the CUET Sports and Cultural Event Management System project is influenced by Pressman's insights outlined in Figure 2.4 on page 27 of "Software Engineering: A Practitioner's Approach."

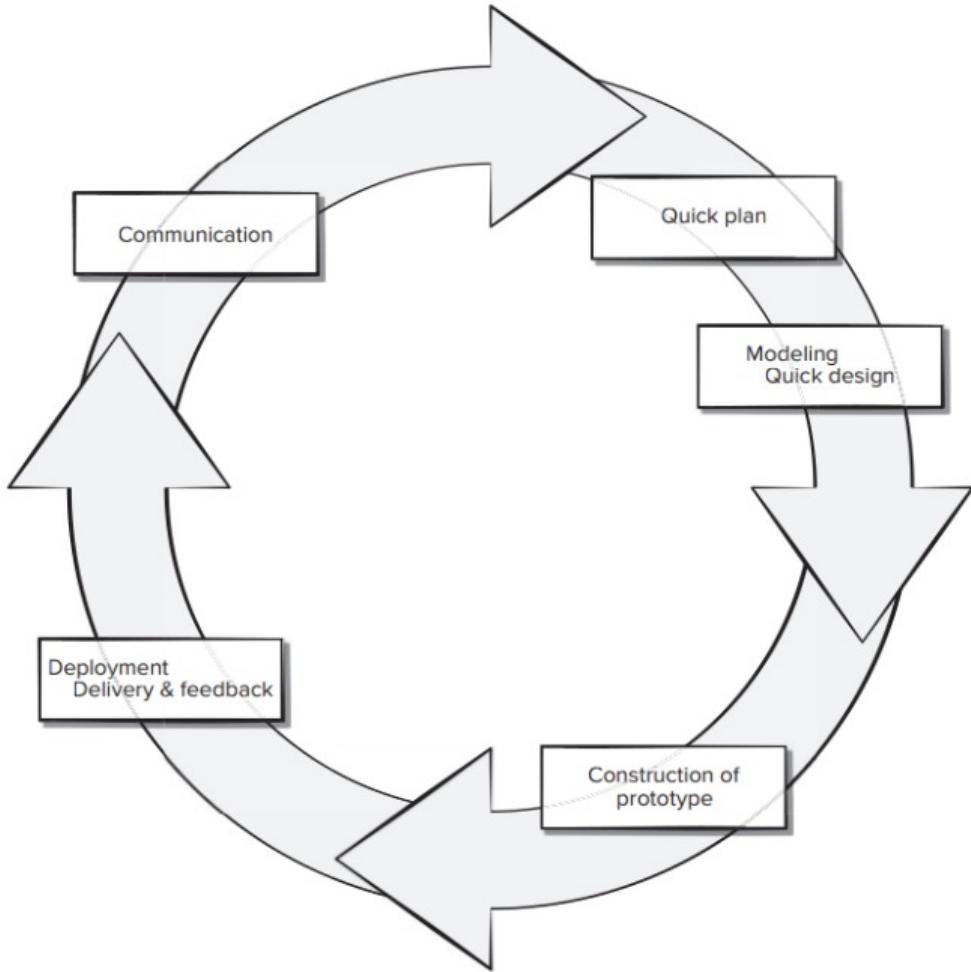


Figure 2.1: Prototyping Process Model

### **2.2.1 Rationale for Choosing Prototyping Process Model for CUET Sports and Cultural Event Management System**

The decision to embrace the Prototyping Process Model in the development of the CUET Sports and Cultural Event Management System is grounded in several project-specific considerations. The rationale supporting the adoption of the prototyping model for the CUET Sports and Cultural Event Management System is as follows:

#### **CUET Sports and Cultural Event Management System:**

The CUET Sports and Cultural Event Management System caters to a diverse array of dynamic event management requirements. The prototyping model is chosen to accommodate evolving needs, allowing for iterative refinement as stakeholders gain a clearer understanding of their requirements during the development process.

#### **Modular Functionalities:**

Given the project's scope, which includes diverse functionalities such as Event Creation, Scheduling, Participant Management, and Real-time Updates, the Prototyping Process

supports an incremental development approach. This enables the phased implementation of different modules, facilitating early feedback gathering and continuous improvement of the system.

### **Active Involvement of Stakeholders:**

Prototyping encourages ongoing engagement with stakeholders, including event organizers, participants, and administrators, throughout the development lifecycle. Regular iterations and feedback loops empower stakeholders to actively contribute to shaping the system, ensuring that the CUET Sports and Cultural Event Management System aligns with the dynamic needs and preferences of the university community.

### **Risk Management in Event Coordination:**

Prototyping provides an early working model of the system, enabling stakeholders to visualize and interact with the evolving product. Early visibility aids in identifying and mitigating risks associated with misunderstood event management requirements, reducing the likelihood of significant issues emerging in later stages.

### **Agile Adaptability:**

The Prototyping Process allows for swift adjustments based on stakeholder feedback, a critical aspect in the rapidly evolving landscape of sports and cultural events. The iterative nature of prototyping ensures rapid adaptation to changing needs, thereby reducing the time required for major revisions.

### **Enhancing User Satisfaction:**

Prototyping offers stakeholders a tangible product early in the development process. Event organizers and participants experience hands-on interactions, fostering satisfaction as they witness their feedback being incorporated into subsequent iterations of the CUET Sports and Cultural Event Management System.

## **2.3 Risk Analysis**

Risk Analysis for CUET Sports and Cultural Event Management System Risk analysis is a crucial component in software development, aiding in the systematic identification and mitigation of potential challenges and uncertainties. Drawing insights from established practices, particularly focusing on risk management requirements as outlined in authoritative sources, our risk analysis is tailored to the specific needs of the CUET Sports and Cultural Event Management System. This structured approach facilitates the recognition, assessment, and proactive addressing of risks, contributing to the project's overall resilience and success.

In Table 1, we present the risk assessment table for our developed software.

N.B: BU - Business Impact, TE - Technology challenges, RD - Resource and Development issues, DE - Development Environment risks, TR - Data Security Concerns

Risks	Category	Probability (%)	Impact
Inadequate Participation by Event Organizers	BU	20%	3
Technological Integration Challenges	TE	25%	4
Integration Challenges with Systems	TE	40%	2
Limited Sponsorship Availability	RD	30%	2
Insufficient User Training	DE	20%	2
Data Security Concerns	TR	25%	4
Tightened Project Deadlines	BU	50%	2

Table 2.2: Risk Analysis for CUET Sports and Cultural Event Management System

### **Inadequate Participation by Event Organizers (BU):**

- Probability: 20%
- Impact: 3
- Mitigation: Scalability planning and continuous monitoring of user patterns.

### **Technological Integration Challenges (TE):**

- Probability: 25%
- Impact: 4
- Mitigation: Through integration testing

### **Integration Challenges with Systems (TE):**

- Probability: 40%
- Impact: 2
- Mitigation: Thorough testing and validation of system integrations.

### **Limited Sponsorship Availability (RD):**

- Probability: 30%
- Impact: 2
- Mitigation: Building a network of sponsors and efficient scheduling.

### **User Training Insufficiency (DE):**

- Probability: 20%
- Impact: 2
- Mitigation: Comprehensive training programs and user-friendly interfaces.

## **Data Security Concerns (TR):**

- Probability: 25%
- Impact: 4
- Mitigation: Robust encryption, regular security audits, and compliance with data protection standards.

## **Tightened Project Deadlines (BU):**

- Probability: 50%
- Impact: 2
- Mitigation: Agile project management methodologies and continuous communication with stakeholders.

The impact values (1, 2, 3, 4) denote catastrophic, critical, marginal, and negligible impacts, respectively. To address and minimize these risks throughout the CUET Sports and Cultural Event Management System project, strategies such as robust system testing, proactive regulatory compliance measures, efficient user training, and stringent security protocols will be implemented. Regular monitoring and adaptation of risk management strategies will be crucial for ensuring the success of the project.

## **2.4 Constraints to project implementation**

### **Schedule Constraints:**

The implementation of the CUET Sports and Cultural Event Management System encounters challenges related to scheduling, especially during the iterative refinement phase inherent in the chosen Prototyping Process Model. The iterative nature of the model may extend the development timeline as feedback and adjustments are incorporated.

### **Budget Constraints:**

Achieving optimal performance for the system necessitates a robust server infrastructure. However, reconciling the need for high-performance servers with budget limitations poses significant challenges in resource allocation for the project. Striking a balance between performance and budget becomes a critical aspect of strategic project management.

### **Software Constraints:**

The iterative refinement process within the Prototyping Process Model may introduce challenges during the software development phase for the CUET Sports and Cultural Event Management System. These challenges could impact the overall project timeline, emphasizing the need for meticulous planning and efficient management of software-related constraints.

## **Hardware Constraints:**

Ensuring the optimal performance of the CUET Sports and Cultural Event Management System relies heavily on a robust server infrastructure. Inadequate processing power or hardware capabilities may lead to potential delays in system responsiveness, presenting a critical hardware constraint for the project. Overcoming these constraints requires careful planning and strategic management to address scheduling, budgeting, and both software and hardware limitations during the system's implementation.

## **2.5 Hardware and Software Resource (Tools/Language) Requirements**

### **2.5.1 Software Requirements**

#### **Frontend Development:**

- Languages:
  1. JavaScript
  2. HTML (HyperText Markup Language)
  3. CSS (Cascading Style Sheets)
- Framework:
  1. React

#### **Backend Development:**

- Framework:
  1. Spring Boot (Web application framework)

#### **Database Connectivity:**

- Database Management System:
  1. MySQL
- Database Administration:
  1. MySQL Workbench

#### **Development Environment:**

- Integrated Development Environment (IDE):
  1. Visual Studio Code
  2. IntelliJ IDEA IDE

#### **Version Control:**

- Git for version control management.

## 2.5.2 Hardware Requirements

- A fast personal computer with suitable specifications for development purposes. This includes sufficient processing power, memory, and storage.

## 2.6 Project Timeline and Schedule

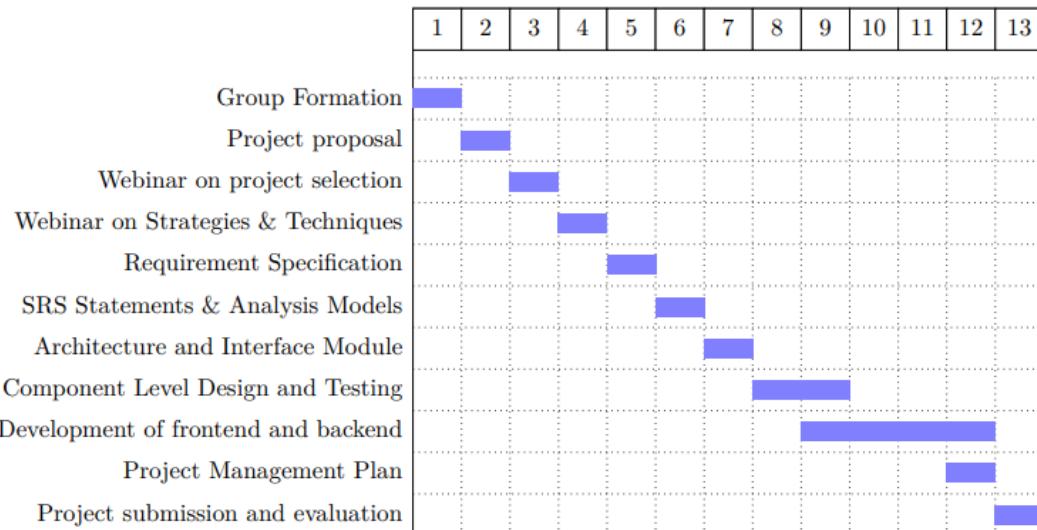


Figure 2.2: Project Gantt Chart

## 2.7 Estimated Budget

The anticipated budget for the CUET Sports and Cultural Event Management System undergoes a comprehensive evaluation, considering various critical factors to ensure the system's successful development and deployment within the specified scope and requirements. The budget considerations are outlined in Table 3. Influencing factors include:

### Project Type:

The CUET Sports and Cultural Event Management System is categorized as a comprehensive software development project. It involves creating an integrated system for managing sports and cultural events, representing a substantial development effort.

### Project Size:

Classified as a medium-sized project, the system acknowledges the intricacies involved in managing modules such as event creation, scheduling, participant management, and real-time updates.

## **Development Team Size:**

The team size for the CUET Sports and Cultural Event Management System will be determined based on the project's unique requirements. The overall composition can be seen in Table 2. Key roles include Project Manager, UI/UX Designer, Database Specialist, Developers (more than 3), and potential QA Testers. The team structure will adapt to the workload and specific role demands.

## **Additional Considerations:**

Other factors, such as the chosen technology, infrastructure prerequisites, and potential integrations with third-party systems, contribute significantly to the overall budget estimation. The system's complexity, integration demands with existing healthcare infrastructure, and the necessity for a responsive server are pivotal factors influencing budget considerations.

Role	Requirements
Project Coordinator	Likely Required
User Experience Expert	Essential
Database Expert	Essential
Developer	More than 2
QA Tester	Must Needed

Table 2.3: Team Size Outlook

Area	Specification	Area
Hardware	Laptops*3	1,70,000/=
Software	Other Tools	2500/=
Server		45,00/=
Ux Designer		6,000/=
Developer Cost	Developers*3	22,000/=
Total		2,04,000/=

Table 2.4: Estimated Cost Distribution for CUET Sports and Cultural Event Management System

## **2.8 Social/Cultural/Environmental impact of the project**

### **Social Impact:**

The CUET Sports and Cultural Event Management System plays a pivotal role in fostering social connectivity and engagement within the university community. It serves as a dynamic platform that brings together students, faculty, and staff, enhancing collaboration and coordination in organizing and participating in various sports and cultural events. By facilitating seamless communication and interaction, the system contributes to the development of a vibrant social ecosystem within the campus.

### **Cultural Impact:**

In alignment with the cultural values of diversity and inclusivity, the system supports a wide array of cultural events that represent the rich tapestry of traditions within the university. It acts as a catalyst for cultural exchange, enabling students from different backgrounds to showcase and celebrate their unique cultural identities. The CUET Sports and Cultural Event Management System embraces cultural diversity by accommodating various event formats, ensuring that each cultural expression finds its place within the university community.

### **Environmental Impact:**

The CUET Sports and Cultural Event Management System contributes to environmental sustainability by reducing the need for physical promotional materials and administrative paperwork. By promoting digital communication and event management, the system minimizes the environmental impact associated with traditional promotional methods. Additionally, the system encourages efficient use of resources by providing centralized event information, reducing the need for excessive printing and distribution of event-related materials.

In summary, the CUET Sports and Cultural Event Management System not only enhances social cohesion and cultural diversity within the university but also aligns with contemporary values of sustainability by minimizing environmental impact through digital event management practices.

# Requirement Specifications

## 3.1 Software Requirements Specification (SRS) of User Authentication Module of CUET Sports and Cultural Event Management System

### 3.1.1 Introduction

#### Purpose

The purpose of this document is to provide a detailed description of the requirements for the Authentication module of the CUET Sports and Cultural Event Management System. This module is responsible for handling user registration, authentication, and user profile management.

#### Scope

The User Management module will be an integral part of the CUET Sports and Cultural Event Management System, ensuring secure and efficient user interactions with the system. It covers user registration, login and profile management.

### 3.1.2 Overall Description

#### Product Perspective

The Authentication module interacts closely with other modules of the CUET Sports and Cultural Event Management System, including the Task Management and Feedback modules. It relies on a database to store user information securely.

#### User Classes and Characteristics

- **Guests/Visitors:** Visitors who have not registered or logged in.
- **Registered Users:** Users who have created an account on the system.
- **Administrators:** Users with special privileges to manage user accounts and system settings.

## **Operating Environment**

The system will be web-based, accessible through standard web browsers. It should support common web technologies and databases.

### **3.1.3 Functional Requirements**

#### **User Registration**

- The system shall provide a registration form for users to create an account.
- Users must provide a unique username, email address, and password during registration.
- The system shall validate and ensure the uniqueness of usernames and email addresses.

#### **User Authentication**

- Users must be able to log in using their registered username/email and password.
- The system shall implement secure password storage techniques.

#### **User Authorization and Roles**

- Admin can manage (add, edit, delete) events
- Admin can manage system users.
- Admin can manage the redundancy of similar events.

#### **User Profile Management**

- Users shall be able to edit their profiles, including changing their password and updating personal information.
- The system shall allow users to view their previous history and events.
- The system shall allow users to search for events.
- Admin shall have the ability to view and manage user profiles.
- The system shall allow admin to create, update, and delete user accounts.
- The system shall allow admin to view user details, including previous history and events.
- The system shall allow admin to update and delete the events.

#### **User Roles**

- Users can update and maintain personal information, including user name and password.

### **3.1.4 Pages**

- **Home:** The main landing page providing an overview and navigation to different events of CUET Sports and Culture.
- **Registration:** Allows users to create new accounts by providing necessary information.
- **Login:** Provides a secure authentication mechanism for users to access their accounts.
- **Profile:** Allows users to view and manage their personal information and settings within the platform.

### **Modules**

The system admin shall manage the authentication and task modules.

### **Sub-Modules**

The system admin shall manage signup, task sub-modules.

### **3.1.5 Non-functional Requirements**

#### **Performance**

- The system should support a moderate number of simultaneous user registrations and logins.
- User authentication process should be completed within few seconds
- The system shall be accessible via the internet.
- The system shall be available 24/7.
- The system shall be secure and protect user data.
- The system shall be easy to use and navigate.
- The system shall be scalable to accommodate a growing number of users and events.

#### **Security**

- Passwords must be securely stored using encryption techniques.
- All user interactions must be transmitted over HTTPS.

#### **Usability**

- The user interface for registration and profile management should be intuitive and user-friendly.
- Error messages should be clear and helpful.

### **3.1.6 Constraints**

- The system must be developed using JAVA programming language, Spring Boot, Thymeleaf Template Engine including HTML, CSS and JavaScript (if needed) and MySQL database.
- The system must comply with relevant data protection regulations and standards.
- The system should use react, Apis in Spring Boot, MySQL.

### **3.1.7 Assumptions and Dependencies**

- The system assumes that users have access to the internet and a web browser.
- The system depends on the availability of sufficient events and the accuracy of user data

### **3.1.8 Future Enhancements**

- The system could be enhanced to allow users to search in more advanced way.
- The system could be enhanced to provide additional flexibility and convenience to its sports and cultural related users.
- The system could be enhanced to provide better reliability and security of event information.

### **3.1.9 Conclusion**

This Software Requirements Specification outlines the detailed requirements for the Authentication module of the CUET Sports and Cultural Event Management System. The successful implementation of these requirements will ensure a secure and user-friendly experience for all users of the system.

## **3.2 Software Requirements Specification (SRS) of Event Management Module of CUET Sports and Cultural Event Management System**

### **3.2.1 Introduction**

#### **Purpose**

The purpose of this document is to outline the requirements for the Task Management module in the CUET Sports and Cultural Event Management System. This module will handle the insert, delete and update sports and cultural events for the registered user, approval from admin and the visits of unregistered user.

#### **Scope**

The Event Management module is a crucial component of the CUET Sports and Cultural Event Management System, responsible for the addition, modification, and retrieval of event information. It interacts with other modules such as Authentication and Search. It will provide a complete user interface for Sports and Cultural Event Management. The system will be used by any existing or new students of CUET to manage its sports and cultural events.

### **3.2.2 Overall Description**

#### **Product Perspective**

The CUET Sports and Cultural Event Management System module is tightly integrated with other modules to provide a seamless experience for users. It relies on a database to store and retrieve event-related information.

#### **User Classes and Characteristics**

- **Registered Users:** Users who have created an account on the system.
- **Administrators:** Users with special privileges to manage user accounts and system settings.

#### **Operating Environment**

The system will be web-based, accessible through standard web browsers. It should support common web technologies and connect to a reliable database.

### **3.2.3 Functional Requirements**

#### **Event Inserting and Editing**

- Required information for event addition includes title, club name, time and date.
- The system shall allow administrators to manage registered users.
- The system shall allow administrators to view event details.

- The system shall eliminate the events immediately after occurring.

### **Event Search**

Users shall be able to search for events based on criteria such as title, club name

#### **3.2.4 Non-functional Requirements**

##### **Performance**

- The system shall be accessible via the internet.
- The system shall be available 24/7.
- The system shall be secure and protect user and event data.
- The system shall be easy to use and navigate.
- The system shall be scalable to accommodate a growing number of events and users.
- The system should support simultaneous event additions and edits by multiple users.
- Event search results should be displayed within few seconds

##### **Security**

- User data should be encrypted during transmission
- Access to event addition, editing, and deletion functionalities should be restricted to unregistered users.

##### **Usability**

- The interface should be intuitive, requiring minimal training for users.
- Error messages should be clear and provide guidance on corrective actions.

#### **3.2.5 Constraints**

- The system must be developed using JAVA programming language, Spring Boot, Thymeleaf Template Engine including HTML, CSS and JavaScript (if needed) and MySQL database.
- The system development should adhere to the budget and timeline constraints.

#### **3.2.6 Assumptions and Dependencies**

- The system assumes that users have access to the internet and a web browser.
- The system depends on the availability of events and the accuracy of user data.

### **3.2.7 Future Enhancements**

- The system could be enhanced to allow users to view events of sports and culture from their phones.
- The system could be enhanced to provide additional flexibility and convenience to users.
- The system could be enhanced to provide better reliability and security of event information.

### **3.2.8 Conclusion**

This Software Requirements Specification outlines the detailed requirements for the Event Management module of the CUET Sports and Cultural Event Management System. The successful implementation of these requirements will ensure effective organization of the events' collection, enhancing the overall user experience.

## **3.3 Software Requirements Specification (SRS) of System Insight Module of CUET Sports and Cultural Event Management System**

### **3.3.1 Introduction**

#### **Purpose**

The purpose of this document is to outline the requirements for the system insight module in the CUET Sports and Cultural Event Management System. This module will mainly handle collecting feedback from its users. Users can see others feedback and update their feedback if needed.

#### **Scope**

The system insight module is a crucial component of the CUET Sports and Cultural Event Management System, responsible for the collection the feedback from users, allowing them to update their feedback and also showing it to its unregistered users, registered users and admin etc. It interacts with other modules such as Authentication and event management. It will provide user interface for collecting user feedback. This feature will be used by any existing or new students of CUET.

### **3.3.2 Overall Description**

#### **Product Perspective**

The CUET Sports and Cultural Event system insight module is tightly integrated with other modules to provide a seamless experience for users. It relies on a database to store and retrieve feedback information.

## User Classes and Characteristics

- **Unregistered Users:** : Visitors who have not registered or logged in.
- **Registered Users:** : Users who have created an account on the system.
- **Administrator:** : Users with special privileges to manage user accounts and event settings.

## Operating Environment

The system will be web-based, accessible through standard web browsers. It should support common web technologies and connect to a reliable database.

### 3.3.3 Functional Requirements

- Providing feedback by registered user after use.
- Visiting others feedback by registered user.
- Unregistered users can also get an overall idea of the system by seeing the feedback.
- Updated existing feedback for the registered user.

### 3.3.4 Non-functional Requirements

#### Performance

- The system shall be accessible via the internet.
- The system shall be secure and protect user and event data.
- The system shall be easy to use and navigate.
- The system shall be scalable to accommodate a growing number of events and users.
- The system should support simultaneous event additions and edits by multiple users.
- Event search results should be displayed within few seconds.

#### Security

Access to feedback providing.

#### Usability

- The user interface for collecting feedback is smooth and user-friendly.
- Error messages should be clear and provide guidance on corrective actions.

### 3.3.5 Constraints

The system will be developed using JAVA programming language, Spring Boot, Thymeleaf template engine including HTML, CSS and JavaScript (if needed) and MySQL database.

### **3.3.6 Assumptions and Dependencies**

- The system assumes that users have access to the internet and a web browser.
- The system depends on the availability of users' feedback and the accuracy of user feedback data.

### **3.3.7 Future Enhancements**

- The systems feedback feature could be enhanced to provide advance feedback system.
- The system could be enhanced to provide additional flexibility and convenience to users.
- The system could be enhanced to provide better reliability and security of event information.

### **3.3.8 Conclusion**

This Software Requirements Specification outlines the detailed requirements for the feedback collection module of the CUET Sports and Cultural Event Management System. The successful implementation of these requirements will ensure effective organization of the feature collection, enhancing the overall user experience.

## **3.4 Stakeholders for the system**

### **3.4.1 Users**

Users are integral stakeholders in the CUET Sports and Cultural Event Management System. They are mainly from CUET Campus e.g., teachers, students, officers, staff, and all types of vendors related to the institution. They request for gate pass for the outing of their product and the security officer or the commander of the campus can check their listing and update the status.

### **3.4.2 Developers**

One stakeholder in charge of designing, developing, and maintaining the system is the development team. They work together to design software, correct bugs, and add new features as needed.

## **3.5 Use case diagram with Graphical and Textual Description**

A use case diagram is a visual aid that shows how different players, such as users or external systems, interact with the system that is being studied. It provides a high-level overview of the features offered by the system and illustrates the many methods by which users or outside organizations might interact with it. Use case diagrams are especially helpful in the early phases of system development because they help express and represent

the desired behavior of the system from the viewpoint of the user. Actors, which stand in for users or external systems, and use cases, which show certain tasks or functions, make up the diagram. The flow of interactions is illustrated by arrows connecting actors and use cases. This visual aid aids in the understanding of the system's general operation and essential components by stakeholders, including developers, analysts, and clients.

The Use Case Diagram for the CUET Sports and Cultural Event Management System illustrates the interactions between different stakeholders and the system itself with three key use cases. A brief overview of our system is shown below:

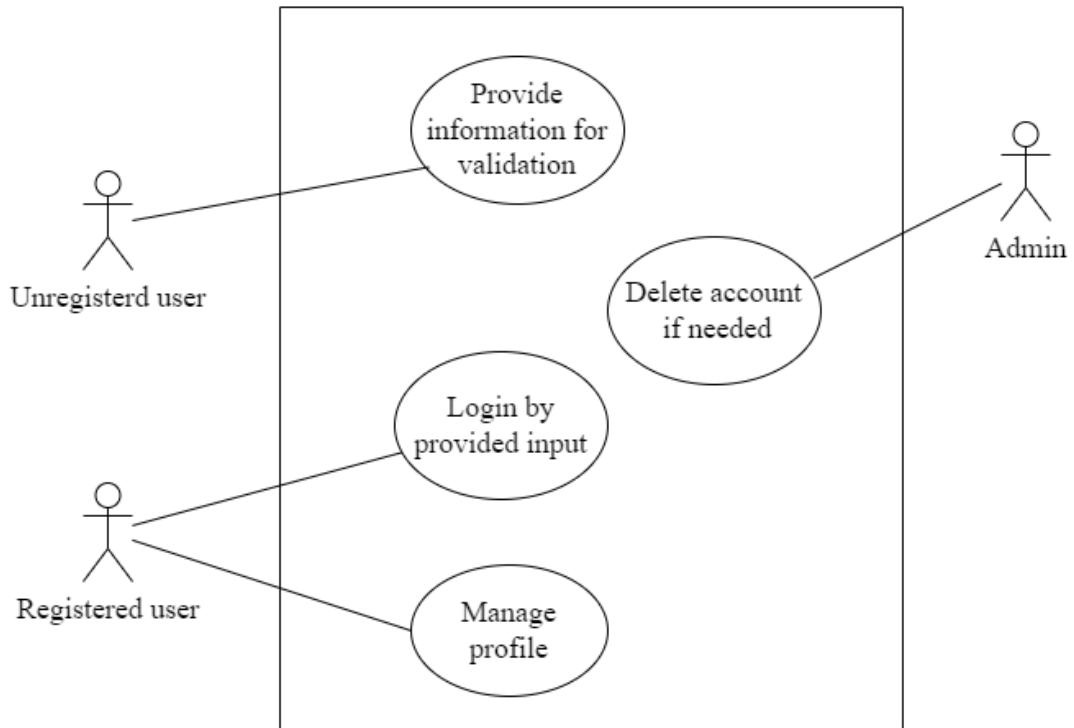


Figure 3.1: Use Case Diagram for User Authentication Module

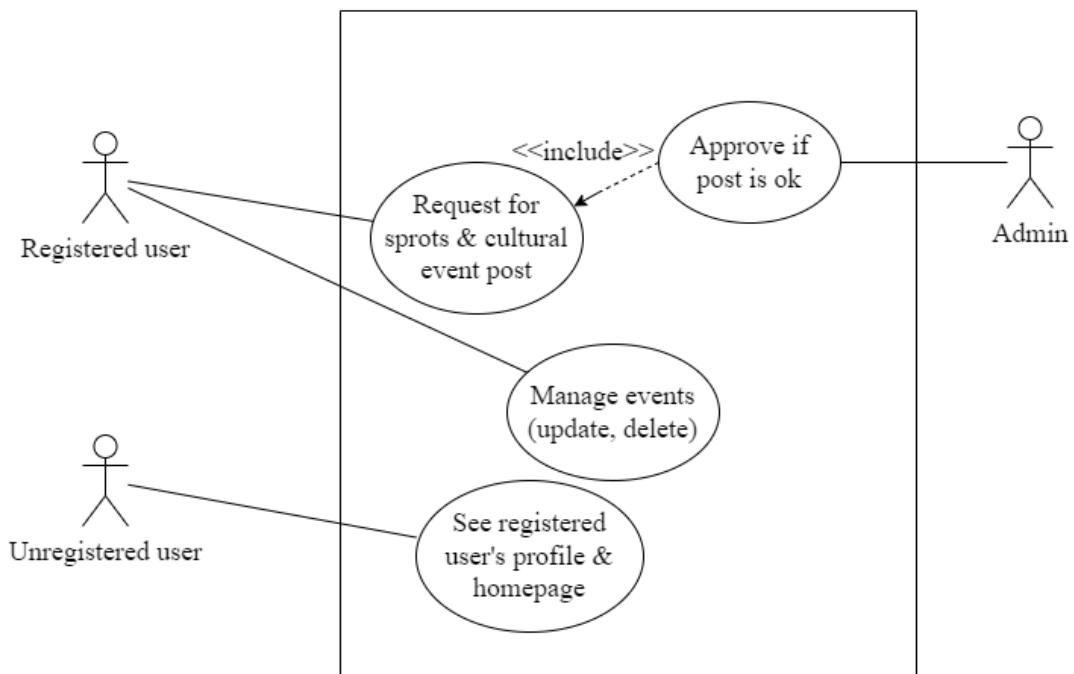


Figure 3.2: Use Case Diagram for Event Management Module

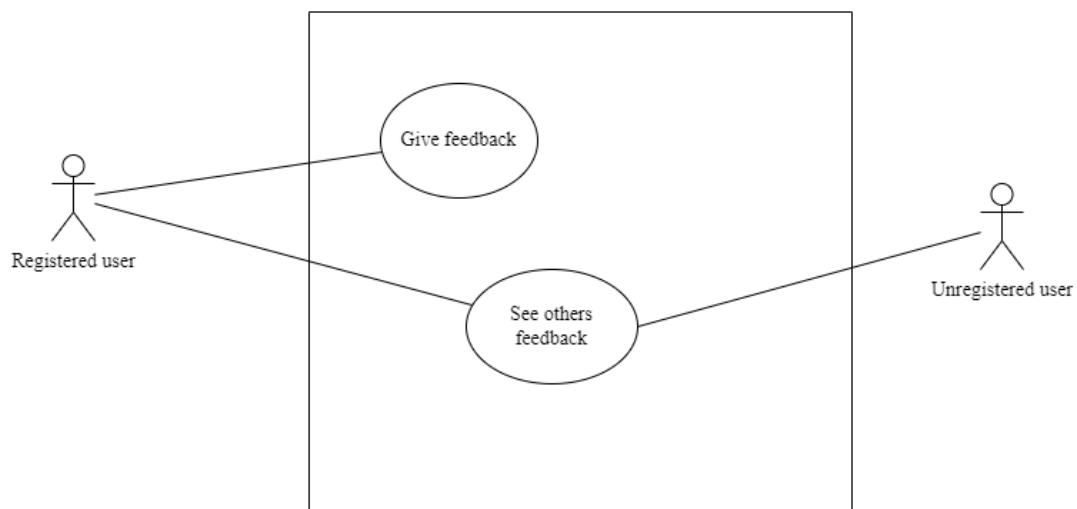


Figure 3.3: Use Case Diagram for System Insight Module

**Use Case:**

User Authentication

**Iteration:**

2, last modification: November 24, 2023

**Primary Actor:**

Registered users of CUET

**Goal in Context:**

- To signup, login for the sports and cultural interested students of CUET.
- To verify with student mail and password.

**Preconditions:**

User must be fully validated; appropriate emails (CUET mails) and passwords must be obtained.

**Trigger:**

The new users can take a look about the upcoming sports and cultural events and the registered users can manage the task management module.

**Scenario:**

- The new user signs up onto the sport and cultural event management system.
- The registered user and admin enters his or her email and password.
- The system displays all major function buttons.
- The new user visits the homepage of events.
- The new user and registered user searches for the upcoming events.

- The new user visits profiles of the registered users.

**Exceptions:**

- Email or passwords are incorrect or recognized – see use case ‘Verify account’.
- User can create only one account with his/her email.

**Priority:**

High priority, to be implemented before event management module.

**When Available:**

First increment

**Frequency of Use:**

Frequent

**Channel to Actor:**

Via PC-based browser and internet connection.

**Secondary Actors:**

Server

**Channels to Secondary Actors:**

Accessed through a desktop web browser using an internet connection.

**Open Issues:**

- What mechanisms protect unauthorized use of this capability by users of the software?
- Will system response via mails outside of CUET mails, i.e. is security sufficient?

**Use Case:**

Event Management

**Iteration:**

2, last modification: November 24, 2023

**Primary Actor:**

Users

**Goal in Context:**

- To add, edit and delete the sports and cultural events of CUET
- To notify after the event completion
- To manage the redundancy of same events

**Preconditions:**

User must be fully validated; appropriate emails (CUET mails) and passwords must be obtained

**Trigger:**

The new users can take a look about the upcoming sports and cultural events and the registered users can manage the task management module.

**Scenario:**

- The registered user and admin enters his or her email and password.
- The registered user selects ‘add event’ button.
- The system displays title of the event, time and date and club name.
- The admin monitors events and deletes events after completion of an event.
- The new user visits the homepage of events

- The new user and registered user searches for the upcoming events.
- The new user visits profiles of the registered users.

**Exceptions:**

Unregistered user selects ‘add event’ button, system displays appropriate error message

**Priority:**

Moderate priority, to be implemented after Authentication module.

**When Available:**

Second increment

**Frequency of Use:**

Frequent.

**Channel to Actor:**

Via PC-based browser and internet connection.

**Secondary Actors:**

Server

**Channels to Secondary Actors:**

System administrator: Web-based system

**Open Issues:**

- What mechanisms protect unauthorized use of this capability by users of the software?
- Will system response via mails outside of CUET mails?

**Use Case:**

System Insight

**Iteration:**

3, last modification: November 24, 2023

**Primary Actors:**

Registered users of CUET.

**Goal in Context:**

- To give feedback through text
- To see others' feedback

**Preconditions:**

User must not be fully validated; users should able to visit the website.

**Trigger:**

The new users can take a look about the feedback given by the other users and the registered users can give feedback through text and email.

**Scenario:**

- The registered user can give feedback to the event management system.
- The registered user selects add 'feedback' button.
- Admin validates, ensuring all fields are filled
- The system displays the list of feedback given by the users
- The admin monitors if any fake users add feedback
- The new user visits the list of feedbacks, thus he/she can know about the system properly

**Exceptions:**

Unregistered user selects 'Add Feedback' button, system displays appropriate error message.

**Priority:**

Low priority, to be implemented after Event Management Module.

**When Available:**

Third increment

**Frequency of Use:**

Occasional.

**Channel to Actors:**

Via PC-based browser and internet connection

**Secondary Actors:**

System administrator, unregistered users.

**Channels to Secondary Actors:**

System administrator: Web-based system

**Open Issues:**

- What mechanisms protect unauthorized use of this capability by users of the software?
- Is security sufficient?
- Will system response via mails outside of CUET mails?

## 3.6 Activity Diagram

An activity diagram is a type of flowchart that represents the flow of activities or actions within a software system. It provides a graphical representation of the steps involved in a process, starting from the initiation of an activity and ending with the completion of the same. The diagram also aids in identifying potential bottlenecks or areas for improvement in the process flow, helping to optimize the project timeline and resources. In short, activity diagrams are a valuable tool for visualizing, analyzing, and communicating the flow of activities in a software project, making it easier to understand, manage, and improve the project's work processes. 3.4 shows the activity diagram of our user authentication module.

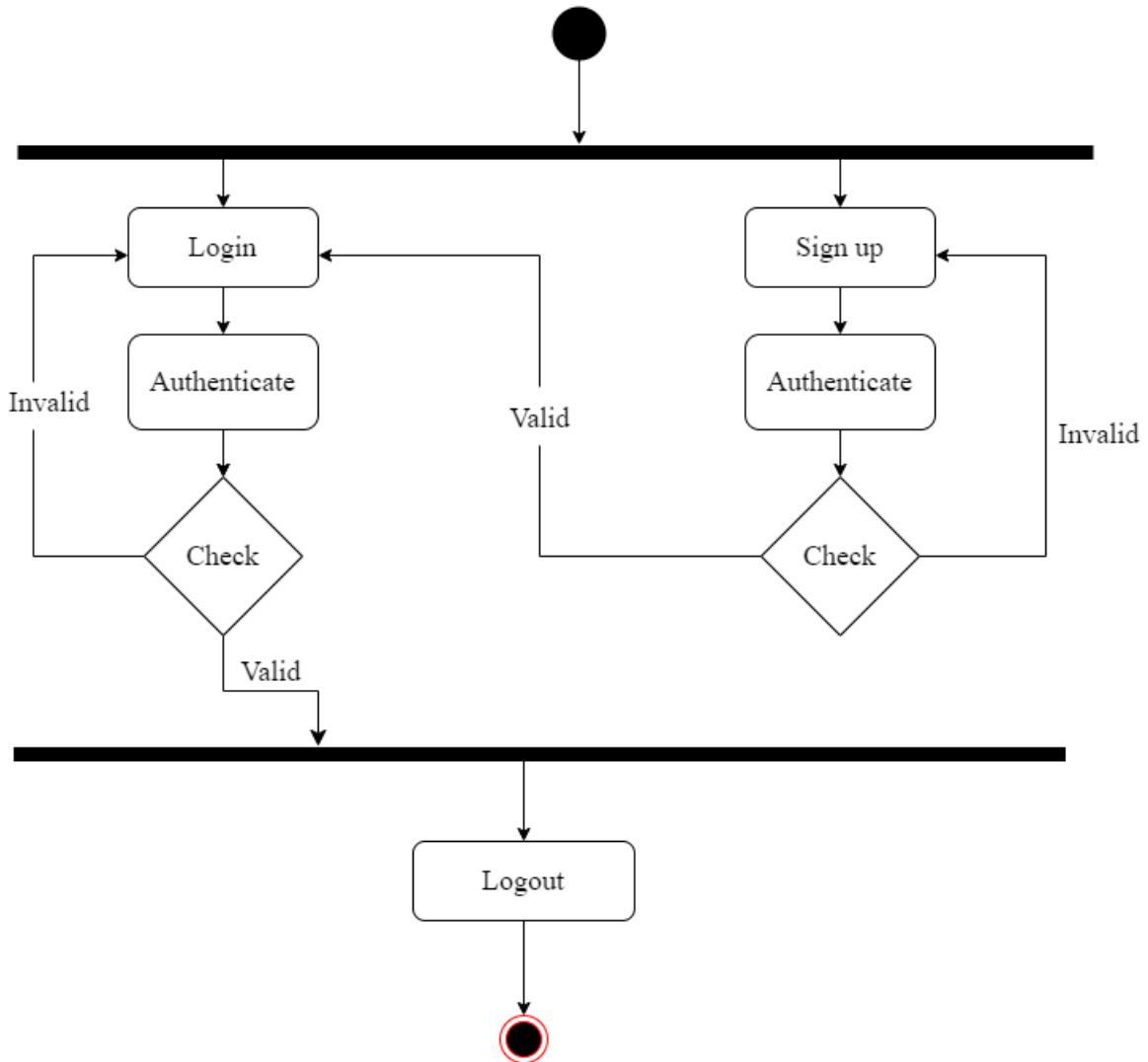


Figure 3.4: Activity Diagram for User Authentication

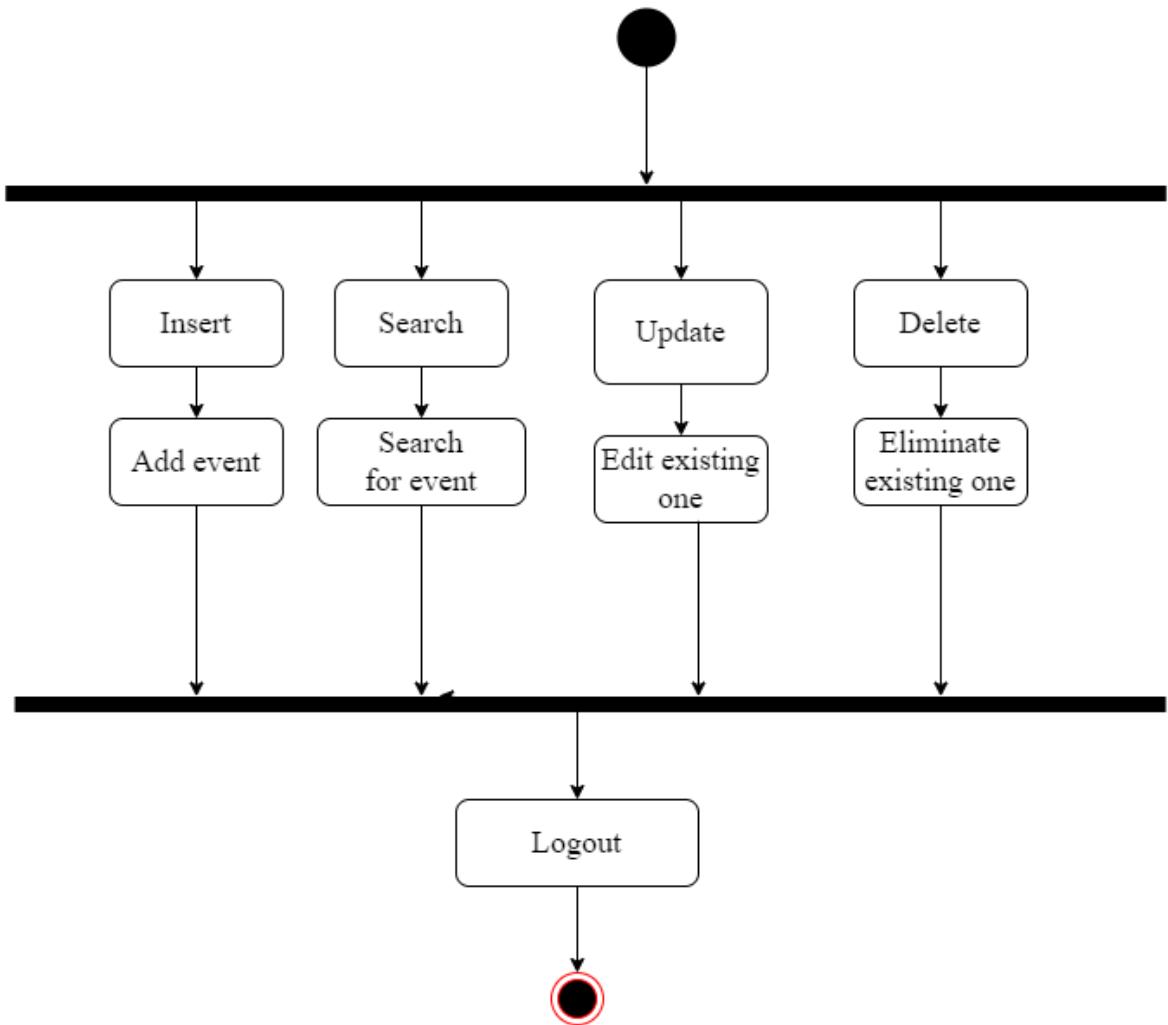


Figure 3.5: Activity Diagram for Event Management

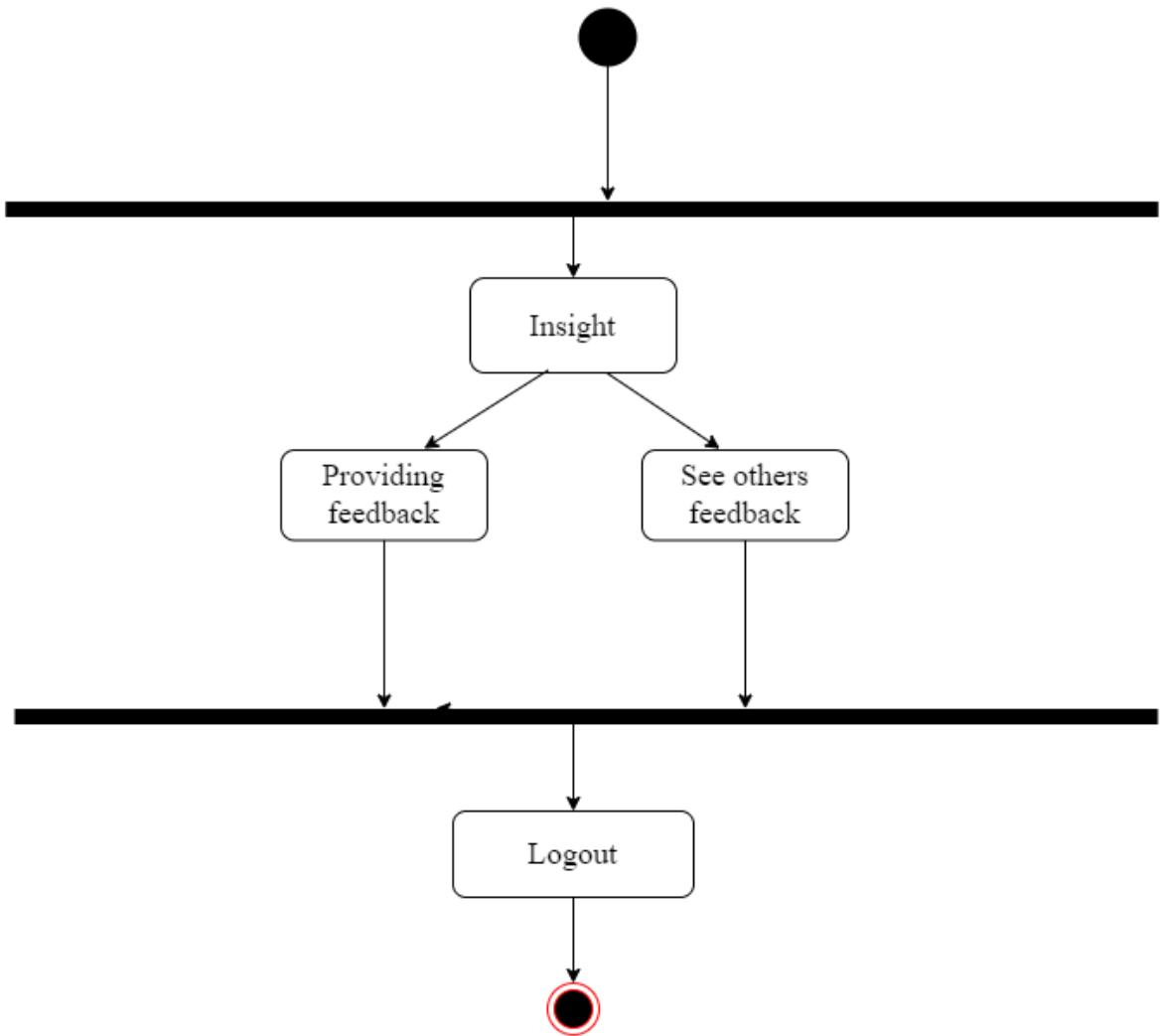


Figure 3.6: Activity Diagram for System Insight

### 3.7 Swimlane Diagram

A swimlane diagram, also known as a cross-functional flowchart or a swimlane flowchart, is a type of process diagram that provides a visual representation of the flow of activities within a system or business process. It organizes these activities into lanes, or "swimlanes," each representing a different participant or department involved in the process. Swimlane diagrams are particularly useful for illustrating the interactions and responsibilities of various entities in a process. Arrows or lines connect the activities, indicating the flow and sequence of tasks. This diagram type is widely used for process mapping, analysis, and improvement in fields such as business process management and system development.

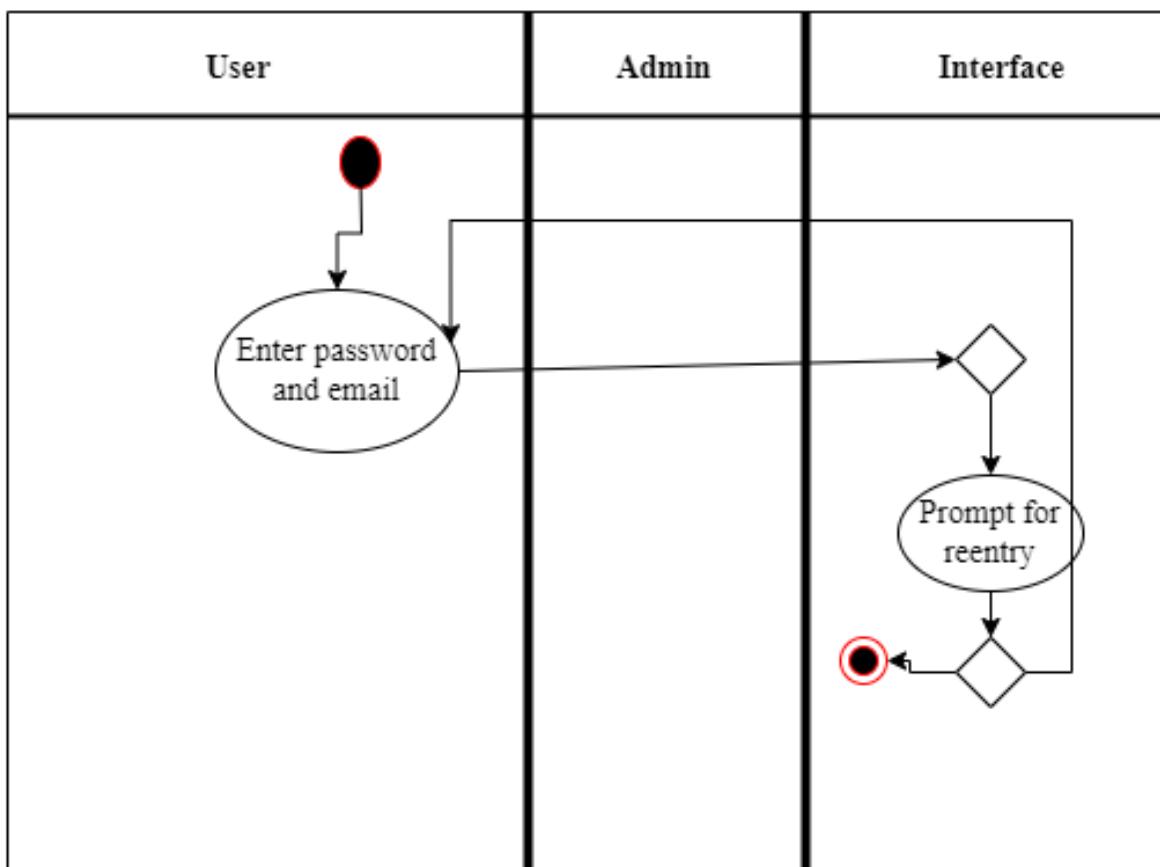


Figure 3.7: Swimlane Diagram for User Authentication

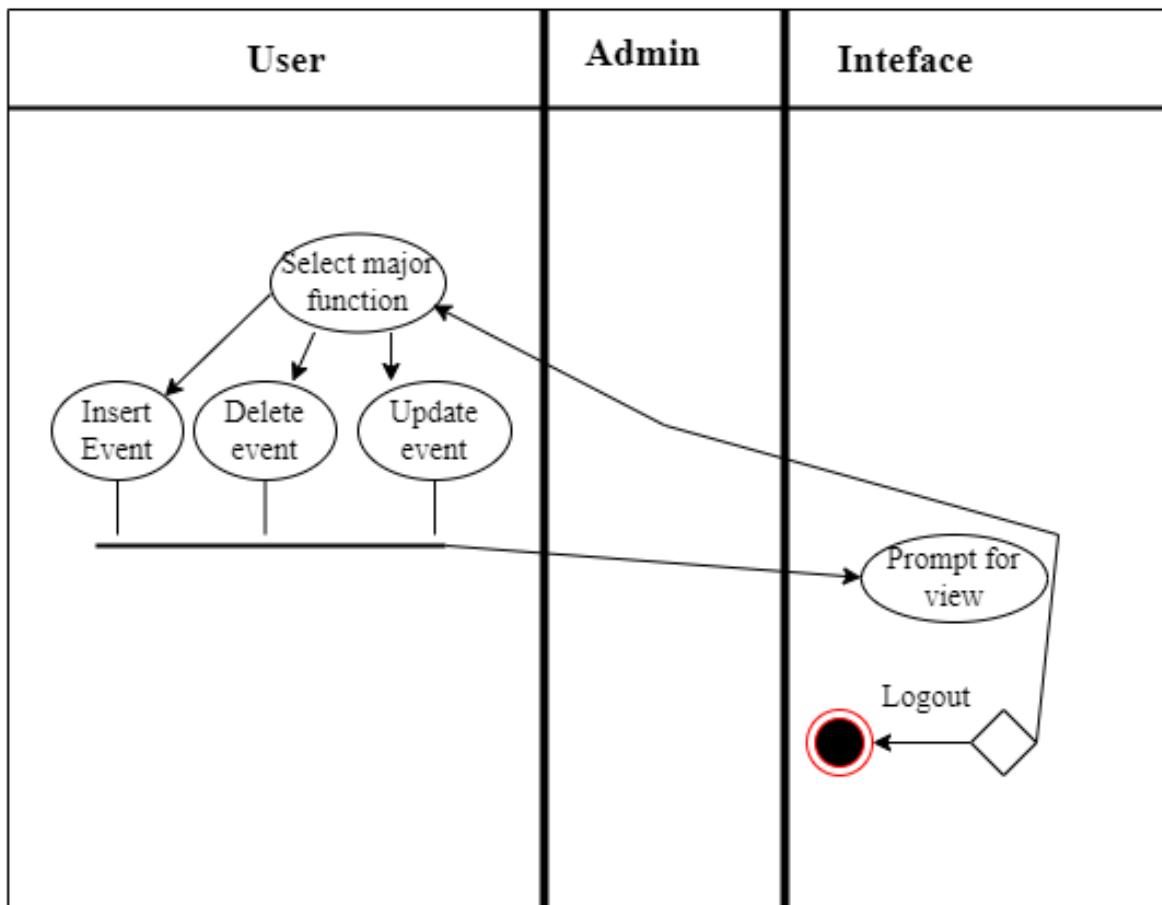


Figure 3.8: Swimlane Diagram for Event Management

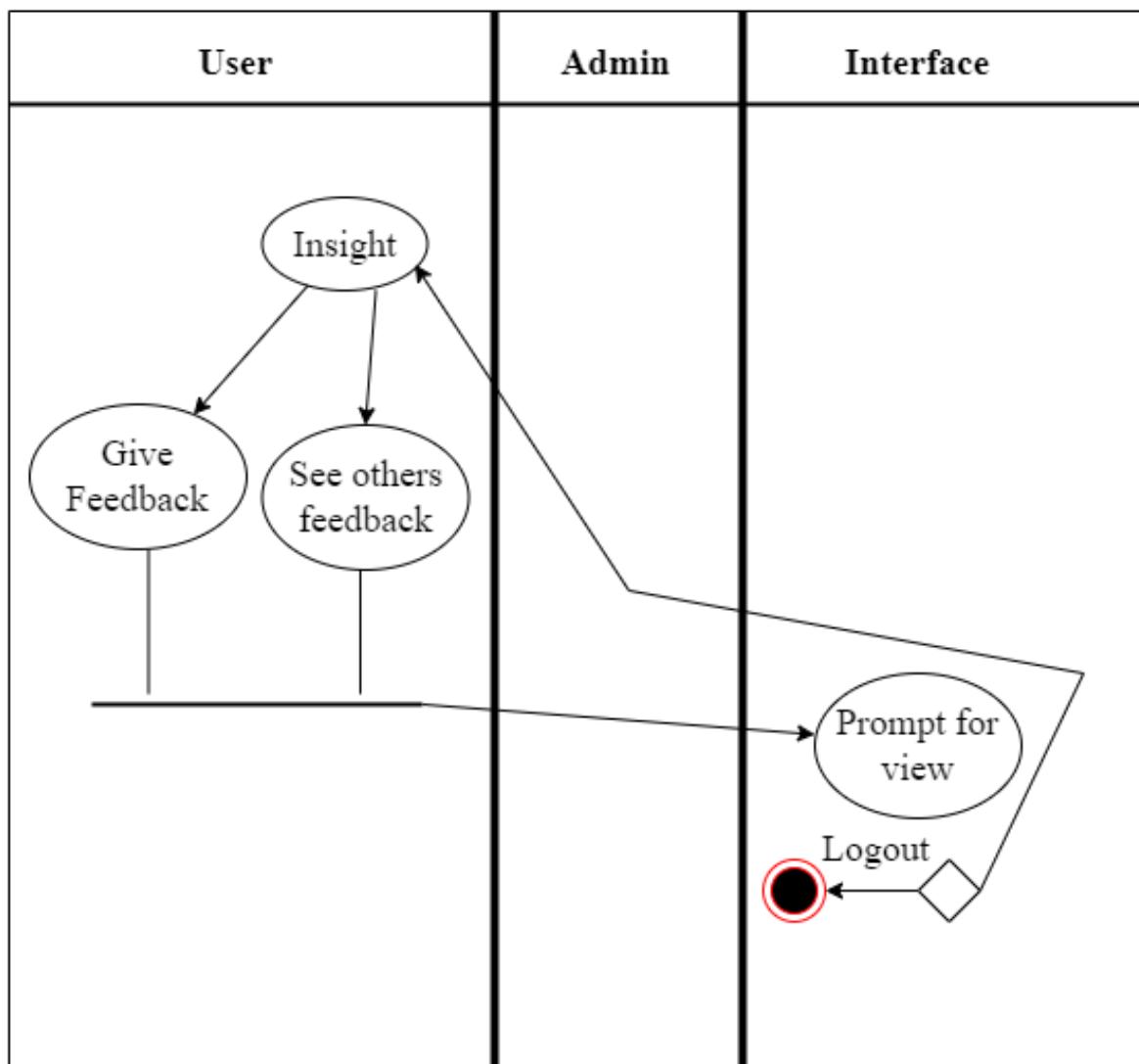


Figure 3.9: Swimlane Diagram for System Insight

## 3.8 Class diagram

The class diagram for the Sports and Culture Event Management project includes essential classes representing the core entities and their relationships. At the center of the system is the "User" class, which serves as the base class for both "RegisteredUser" and "UnregisteredUser." The "Admin" class holds the attributes and methods necessary for administrative functions. The "Event" class encapsulates information about sports and cultural events, including details such as event name, date, time, and venue. It establishes associations with the "RegisteredUser" class, indicating the users who are registered for a particular event. The "Event" class also has a relationship with the "Admin" class, as administrators play a role in managing and overseeing events. Overall, the class diagram represents a structured architecture for the CUET Sports and Culture Event Management system, emphasizing the relationships and interactions between users, administrators, events, registrations.

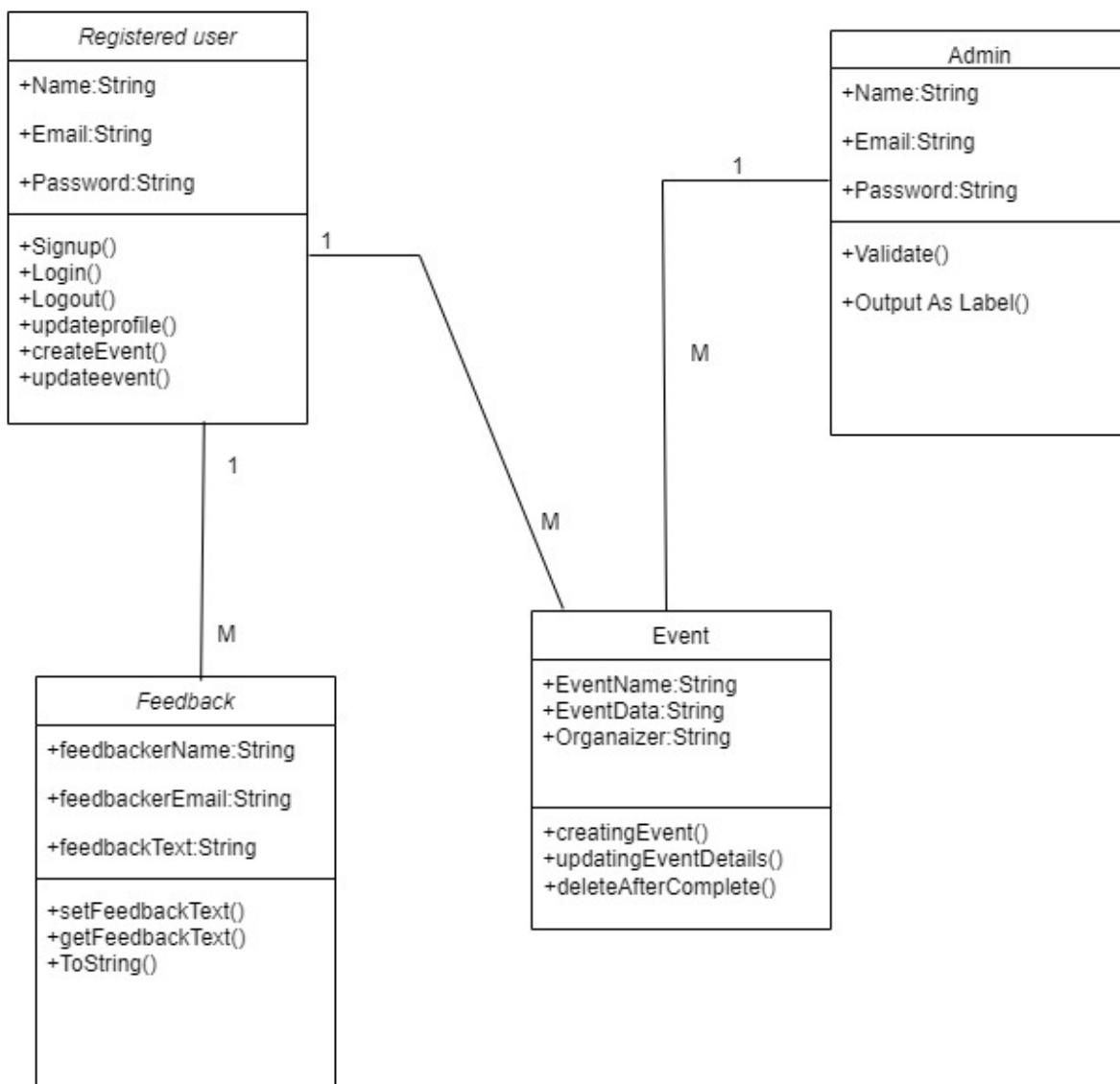


Figure 3.10: Class Diagram for the proposed system

### 3.9 Sequence diagram

A sequence diagram is a visual representation in UML showcasing the chronological flow of interactions between objects or components in a system. Lifelines depict the existence of objects, while arrows represent messages exchanged between them. Synchronous messages show direct, waiting interactions, and asynchronous messages depict non-blocking communication. Activation bars and focus of control indicate when objects are actively processing messages. These diagrams are invaluable for understanding the dynamic behavior of a system during software design and analysis.

The sequence diagram illustrates the flow of interactions in the CUET Sports and Cultural Event Management System. Initially, the user, whether commander, officer, Ansar member, or regular user, interacts with the login/signup page. If the user is new, they create an account; otherwise, they log in with authentic credentials. This sequence diagram captures the user interactions and the flow of information in the CUET Sports and Cultural Event Management System.

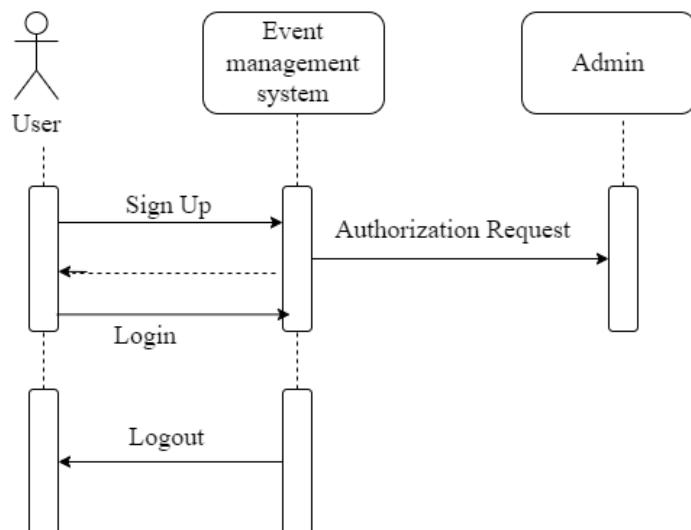


Figure 3.11: Sequence Diagram for User Authentication

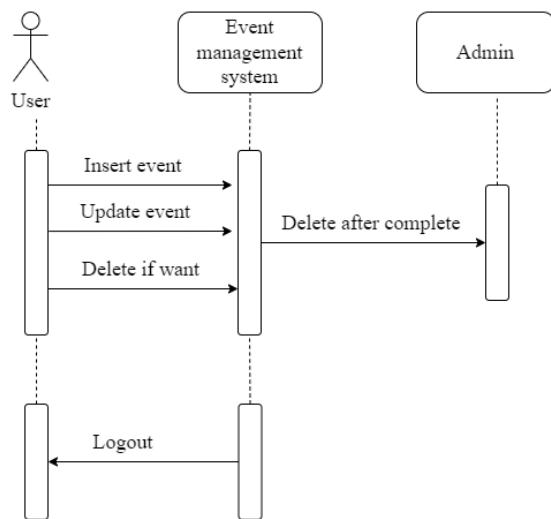


Figure 3.12: Sequence Diagram for Event Management

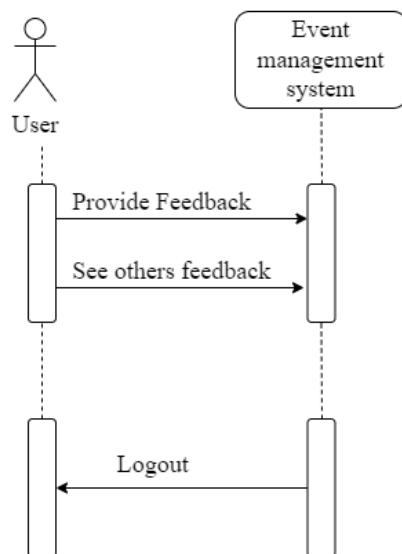


Figure 3.13: Sequence Diagram for System Insight

### 3.10 State Diagram

A state diagram, a key component of Unified Modeling Language (UML), is a graphical representation that models the dynamic behavior of a system by depicting its various states, transitions between states, and the events triggering these transitions. States are shown as rounded rectangles, transitions as arrows, and events as triggers for state changes. It is a valuable tool in software engineering and other domains for visualizing and designing systems that undergo different states and transitions based on external stimuli or conditions.

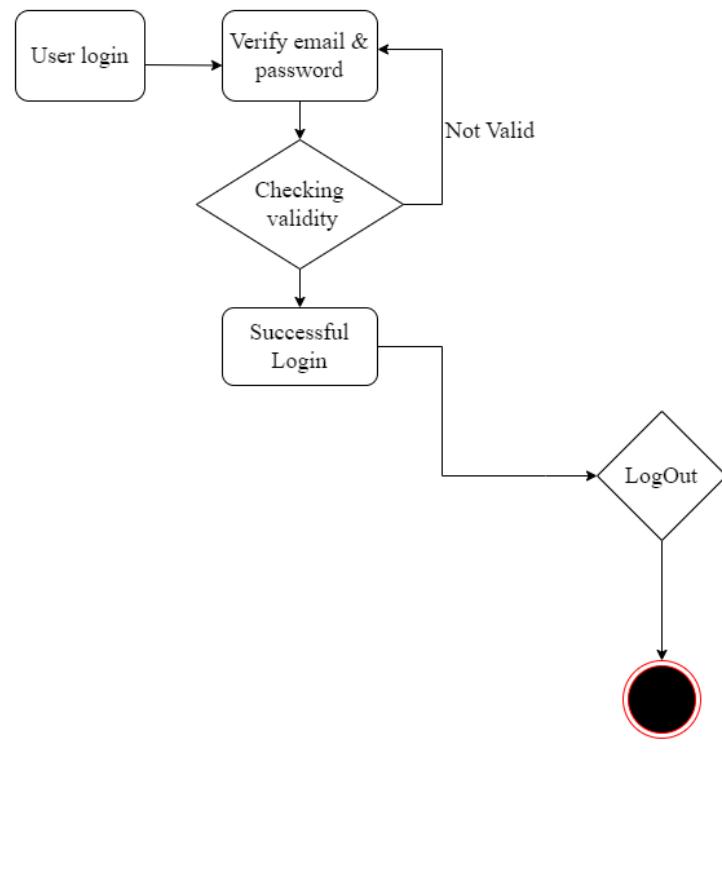


Figure 3.14: State Diagram for User Authentication

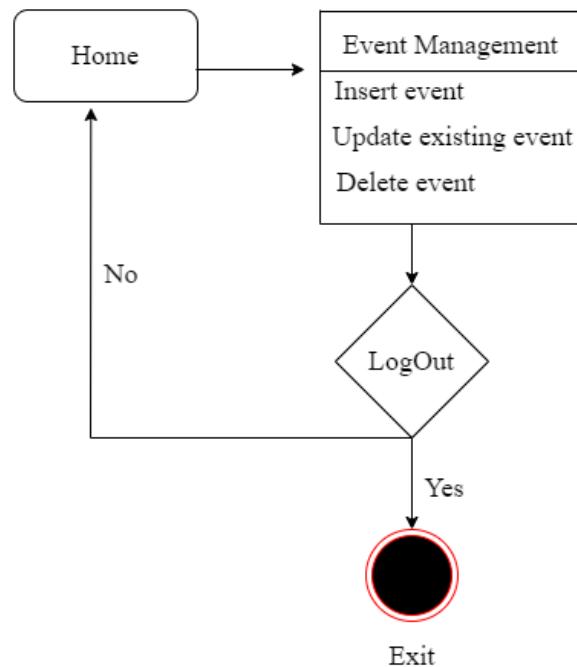


Figure 3.15: State Diagram for Event Management

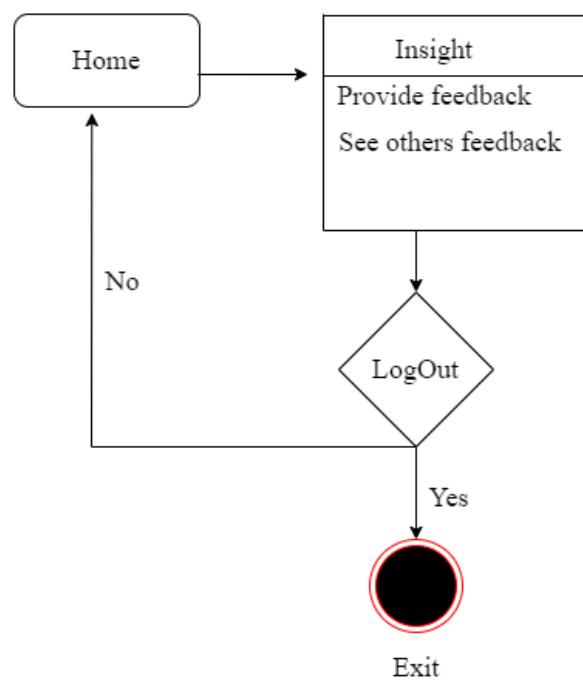


Figure 3.16: State Diagram for System Insight

## **3.11 Safety and Security requirements**

### **3.11.1 Access Requirements**

#### **Users**

- Users should be able to log in to their accounts.
- Update their personal information, including contact details.
- View their available gate pass history, including detailed information.
- Ansar member can see their checkpoint list in every day.

### **3.11.2 Integrity Requirements**

#### **Database Integrity**

- Implement data validation checks to ensure that only valid data is entered.
- Use database constraints to prevent the entry of inconsistent or incomplete data.

#### **Product Integrity**

- Regular system audits to identify and address any potential integrity issues.
- Implementing version control for the software to track changes and updates.
- Regularly testing and validating the system to ensure that it functions as intended without compromising data integrity.

### **3.11.3 Privacy Requirements**

#### **Data Privacy Measures**

- Encryption of sensitive data during transmission and storage.
- Anonymizing users information when possible.
- Regularly auditing and monitoring access to sensitive data.

# Architecture

An architectural diagram is a visual representation that illustrates the structure, components, and relationships within a system, building, or organization. It provides a high-level overview, aiding stakeholders in understanding the design and interactions of key elements. These diagrams can take various forms, such as system architecture diagrams, enterprise architecture diagrams, network architecture diagrams, software architecture diagrams, building architecture diagrams, and cloud architecture diagrams. They serve as valuable communication tools, facilitating collaboration, decision-making, and troubleshooting by simplifying complex structures into comprehensible visuals.

## 4.1 Architectural model/style used

### Data-centered Architecture

An architectural style constitutes a modification to the overarching design of a system, aimed at establishing a structured framework for its various components. This style serves as both a descriptive mechanism and a blueprint for the construction of the system, providing a roadmap for its implementation. In the case of our event management app, a data-centered architectural design was employed. In such an architecture, the database is positioned at the center of the system and is subject to frequent access by other components that perform operations such as updating, adding, deleting, or otherwise modifying the data stored within it. As illustrated in figure-4.1, the database constitutes the nucleus of this architecture and is consistently interacted with by other components.

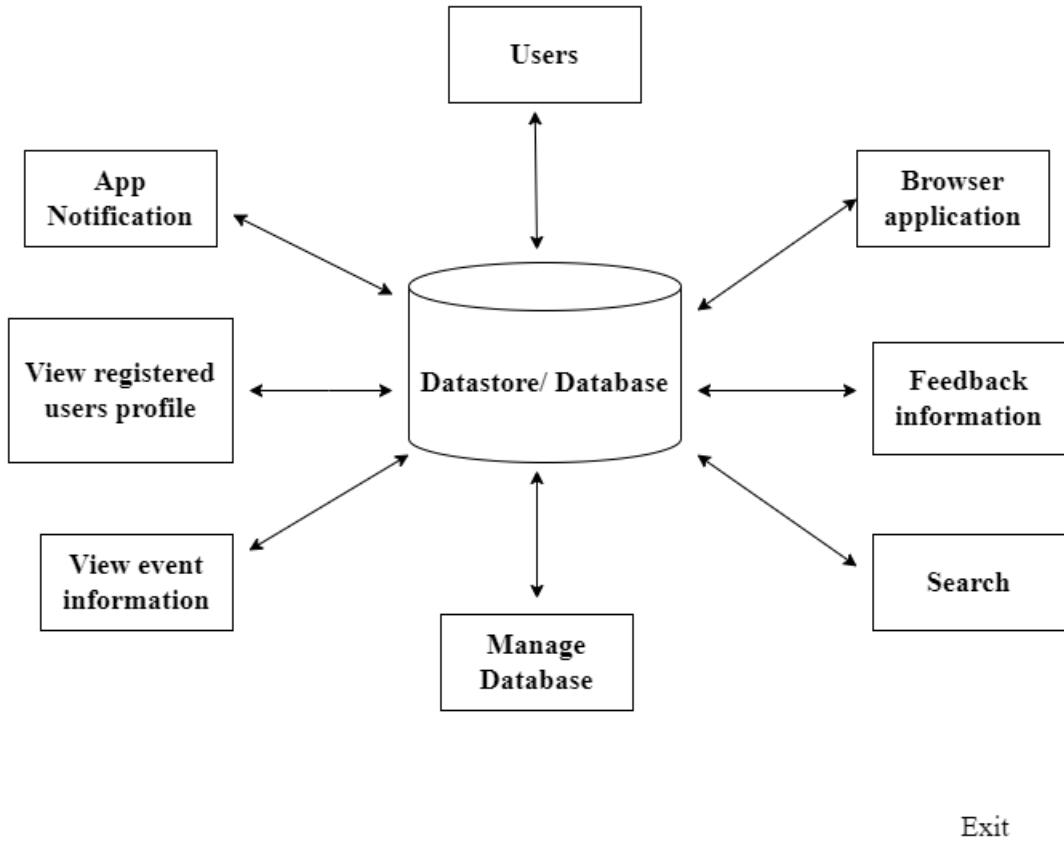


Figure 4.1: Data-centered architecture for the proposed system

### MVC Architecture

The model contains all application-specific content and processing logic. The view contains all interface-specific functions and enables the presentation of content and processing logic required by the end user. The controller manages access to the model and the view and coordinates the flow of data between them. A schematic representation of the MVC architecture is shown in figure-4.2.

Referring to the figure, user requests are handled by the controller. The controller also selects the view object that is applicable based on the user request. Once the type of request is determined, a behavior request is transmitted to the model, which implements the functionality or retrieves the content required to accommodate the request. The model object can access data stored in a corporate database, as part of a local data store, or as a collection of independent files. The data developed by the model must be formatted and organized by the appropriate view object and then transmitted from the application server back to the client-based browser for display on the user's machine.

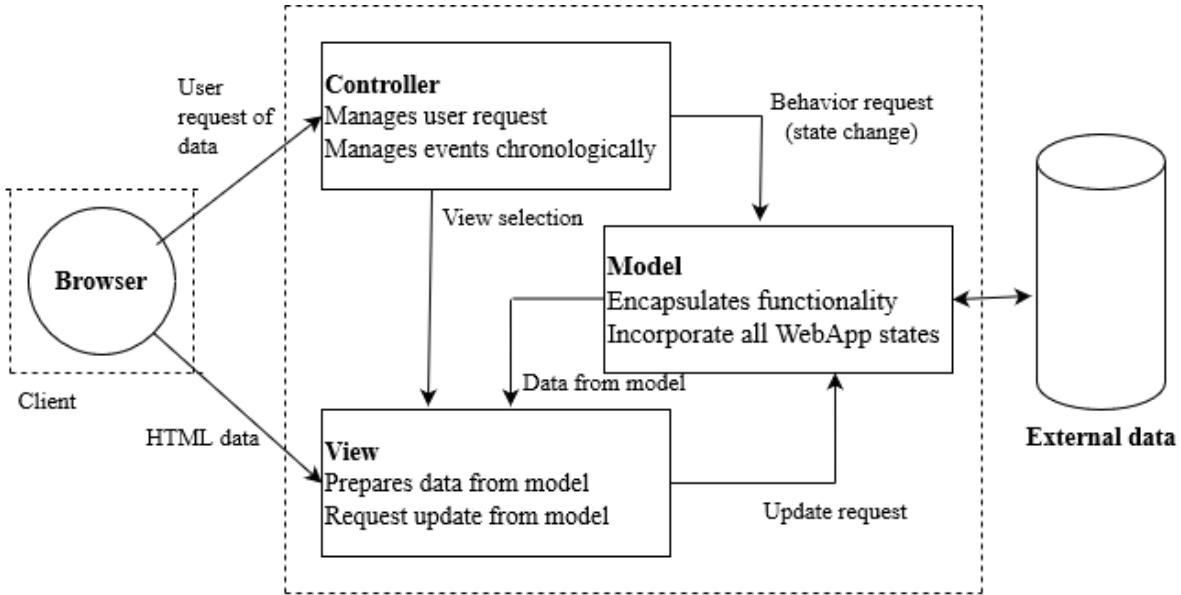


Figure 4.2: MVC architecture for the proposed system

#### 4.1.1 Rationale for choosing our architectural model/style

The rationale for choosing Data-Centered Architecture and MVC (Model-View-Controller) Architecture for our project are given below:

- **Data Integrity and Consistency:** Data-Centered Architecture places a strong emphasis on data integrity and consistency. This is particularly crucial for our WebApp where accurate and reliable data about events, tasks, and user information is paramount.
- **Scalability:** The architecture is inherently scalable, allowing for the efficient handling of growing amounts of data. This is essential for an app that involves managing and coordinating various activities and events.
- **Optimized for Data-Intensive Operations:** Since your app involves managing and coordinating activities, a data-centered approach is well-suited for the data-intensive operations associated with organizing and updating events, tasks, and user details.
- **Modular Design:** MVC architecture provides a modular design where different aspects of the application (Model, View, Controller) are separated. This separation facilitates independent development, testing, and maintenance of each component.
- **Enhanced Maintainability:** The separation of concerns in MVC makes the codebase more maintainable. Changes in one component (e.g., the View) do not directly impact the others, fostering a more flexible and adaptable code structure.
- **Parallel Development:** MVC allows for parallel development, enabling different teams or developers to work on different components concurrently. This can significantly speed up the development process.

## 4.2 Technology, software, and hardware used

- **Thymeleaf:** It is a template engine for frontend and make the web pages dynamic to create web applications. It provides a structure for content on the web and uses a set of tags and attributes to define the content and its appearance.
- **CSS:** Cascading Style Sheets (CSS) is a stylesheet language used for describing the look and formatting of a document written in frontend technology like thymeleaf here. CSS allows developers to separate the presentation of a web page from its content, making it easier to maintain and update the visual design.
- **Spring boot:** Spring boot is backend technology to create the backend for different types of application. It is built on Java and provides various features including reusability, manageable, secured, robust application.
- **MySQL:** MySQL is the database for our project to store and perform operation on data. It includes data insert, update , delete, search specific record with help SQL query. Database is immensely used over traditional file system.

# Design

## 5.1 Component level design following pattern

A component is a software element that exports and encapsulates its implementation through a higher-level interface. It is a modular, portable, replaceable, and reusable entity that encompasses well-defined functionality. Essentially, a component serves as a distinct piece of software, designed to interact with other components, offering a set of functionalities. The defining characteristics of a software component include being a composition unit with explicit contextual dependencies, a contractually defined interface, and various compositional elements.

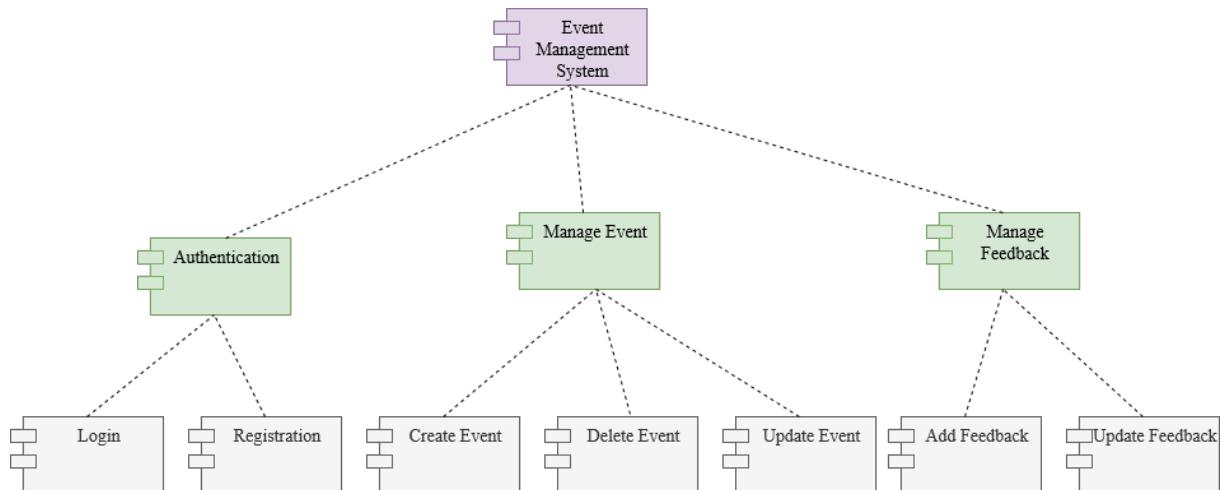


Figure 5.1: Structure chart of CUET Sports and Cultural Management System

During the component-level design phase in software engineering, the focus shifts from high-level system architecture to the granular details of individual components or modules. This phase involves specifying the intricate design aspects of each module, such as data structures, algorithms, interfaces, and dependencies. It serves as a bridge between the architectural blueprint and the practical implementation, providing a roadmap for developers to transform conceptual ideas into functional and cohesive software units. The precision in component-level design ensures that each module operates seamlessly within the larger system, contributing to the overall efficiency, maintainability, and reliability of the software solution.

## User Authentication

The user authentication module in CUET Sports and Cultural Event Management System serves as a crucial component responsible for ensuring secure access to the system's features. This module of our project encompasses functionalities such as credential management, authentication engines supporting various methods, and access control policies based on user roles. It interfaces with the user interface for login processes and communicates securely with the system database. Security measures include encryption for credential storage, secure coding practices, and mechanisms for detecting and responding to suspicious activities. The module is designed for scalability, performance, and undergoes rigorous testing, including security testing and regular penetration testing. Comprehensive documentation is provided to aid in maintenance and future enhancements of this vital authentication component.

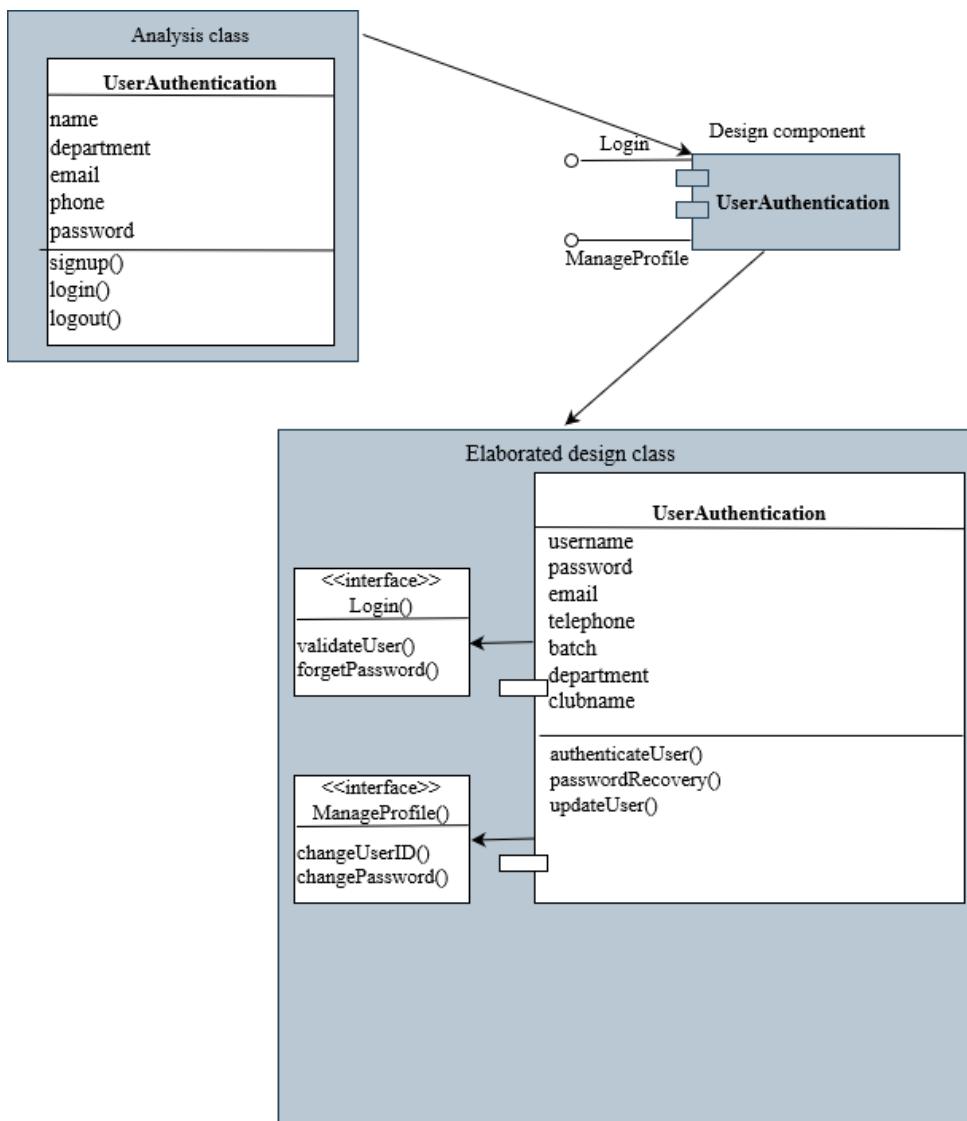


Figure 5.2: Component level design of User Authentication Module

## Event Management

The Event Management Module in the CUET Sports and Cultural Event Management System is a vital component designed for streamlined event handling. It enables users to create, schedule, and manage events, integrating seamlessly with the user interface and securely communicating with the system database. Security measures include access control, data encryption, and secure coding practices. The module is scalable, optimized for performance, and undergoes rigorous testing with comprehensive documentation for maintenance and future enhancements.

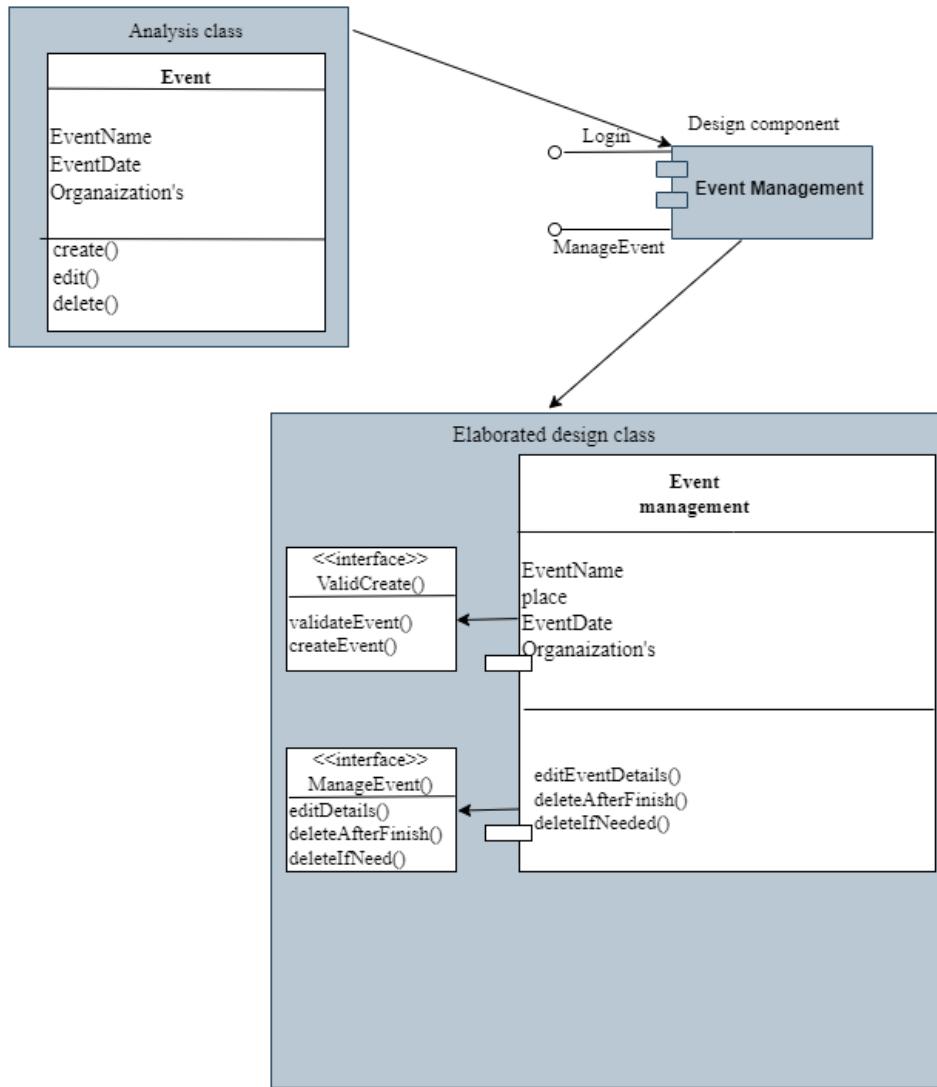


Figure 5.3: Component level design of Event Management Module

## System Insight

The system insight (Feedback) Module in the CUET Sports and Cultural Event Management System plays a pivotal role in collecting valuable insights from users. This module facilitates the submission and management of feedback, allowing registered users to provide input on events. The module interfaces seamlessly with the user interface for feedback submission and securely communicates with the system database. Security

measures include access control policies, data encryption, and adherence to secure coding practices. The module is designed for scalability, optimized performance, and undergoes rigorous testing, complemented by comprehensive documentation for ease of maintenance and future enhancements.

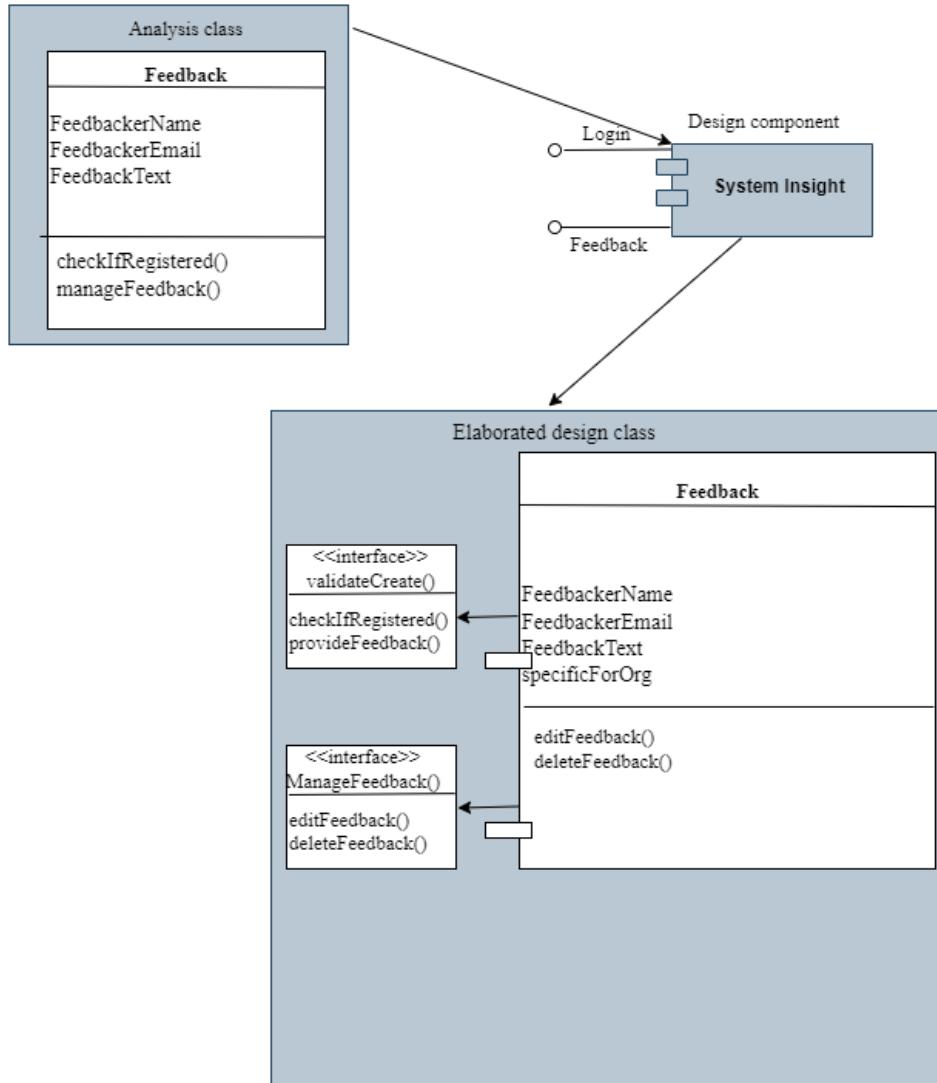


Figure 5.4: Component level design of Feedback Management Module

## 5.2 Dataflow Diagram

### 5.2.1 First Level DFD

The first-level Data Flow Diagram (DFD) of CUET Sports and Cultural Event Management System revolves around the three core modules: user authentication, event management, and feedback management. User authentication serves as the gateway, handling user access and ensuring secure storage of user profiles. The event management module oversees the creation and maintenance of events containing basic information. Meanwhile, feedback management focuses on the handling and organization of feedback within the system. These modules constitute the fundamental pillars of the system's functionality,

governing its essential operations and interactions. This high-level overview forms the basis for further detailed system design and development.

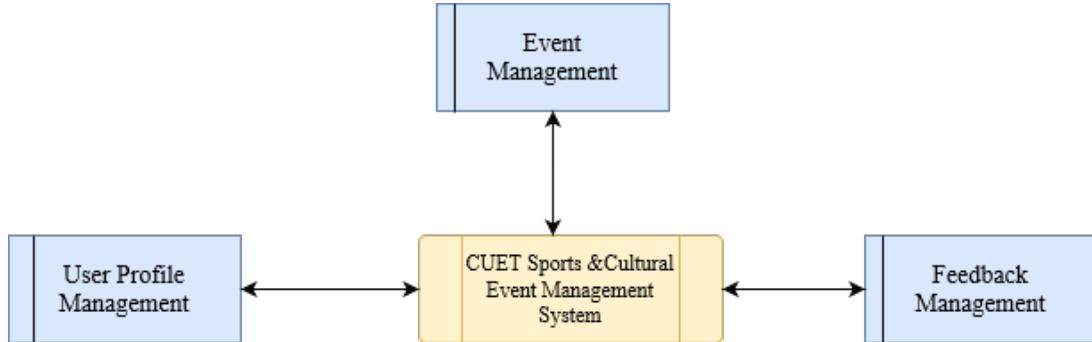


Figure 5.5: First Level DFD

### 5.2.2 Admin DFD

The admin DFD represents the dataflow of the admin in managing the three modules: event management, user profile management and feedback management. This representation illustrates how the admin interacts with and supervises these modules, ensuring smooth operations, data integrity, and efficient management across all three crucial facets of the system.

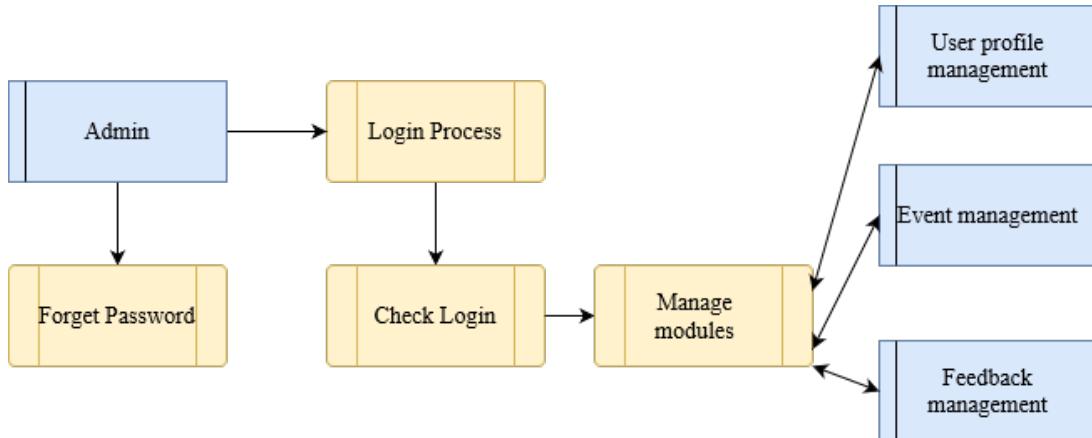


Figure 5.6: Admin DFD

### 5.2.3 User DFD

The user's Data Flow Diagram (DFD) delineates the flow of information and interactions within the CUET Sports and Cultural Event Management System from the user's perspective. It outlines how users engage with the system's functionalities across its six key modules: update profile, event post, event update, event delete, feedback post, event search. The users can be both doctor and patient who are mainly students of CUET. The user DFD visualizes the seamless interaction of users with the system's core features, emphasizing their ability to navigate, input, and retrieve information effectively.

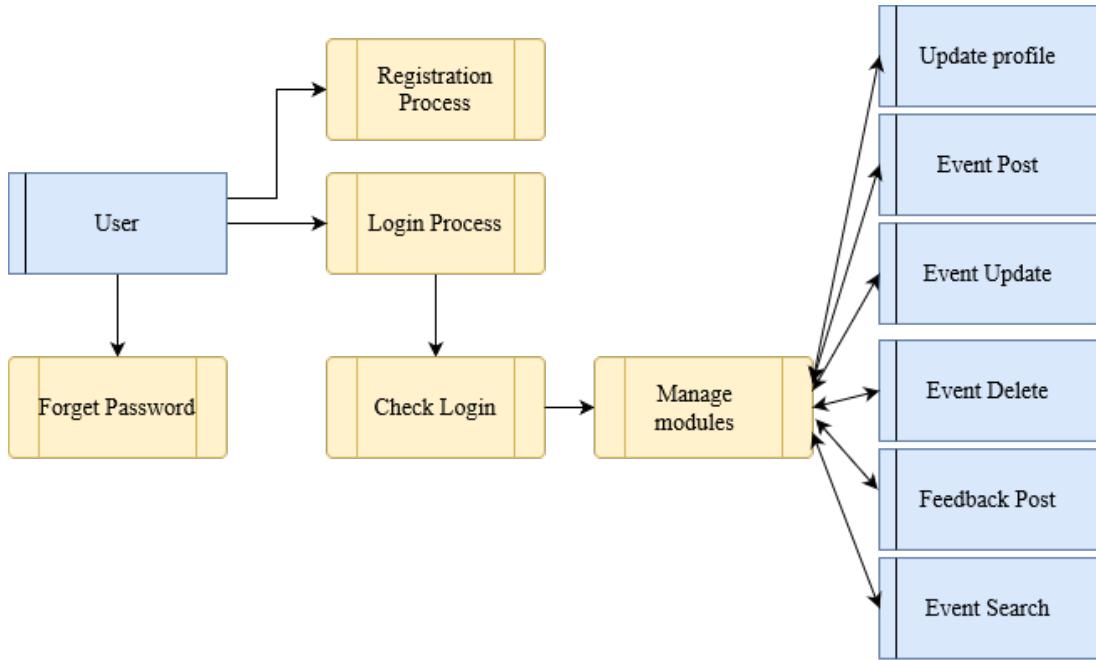


Figure 5.7: User DFD

#### 5.2.4 Entity Relationship Diagram

In developing the Entity-Relationship Diagram (ERD) for the CUET Sports and Cultural Event Management System, the focus lies on mapping the integral entities, their attributes, and the interrelationships crucial for efficient healthcare record management. The primary entities identified include Users, events, feedback, account, admin. Users encompass attributes such as ID, Name, Email, Password, and Phone, serving as the core individuals interacting with the system. . This ERD acts as a foundational blueprint, illustrating how users' events and their feedback are structured and connected within the CUET Sports and Cultural Event Management System, facilitating efficient event data organization and accessibility.

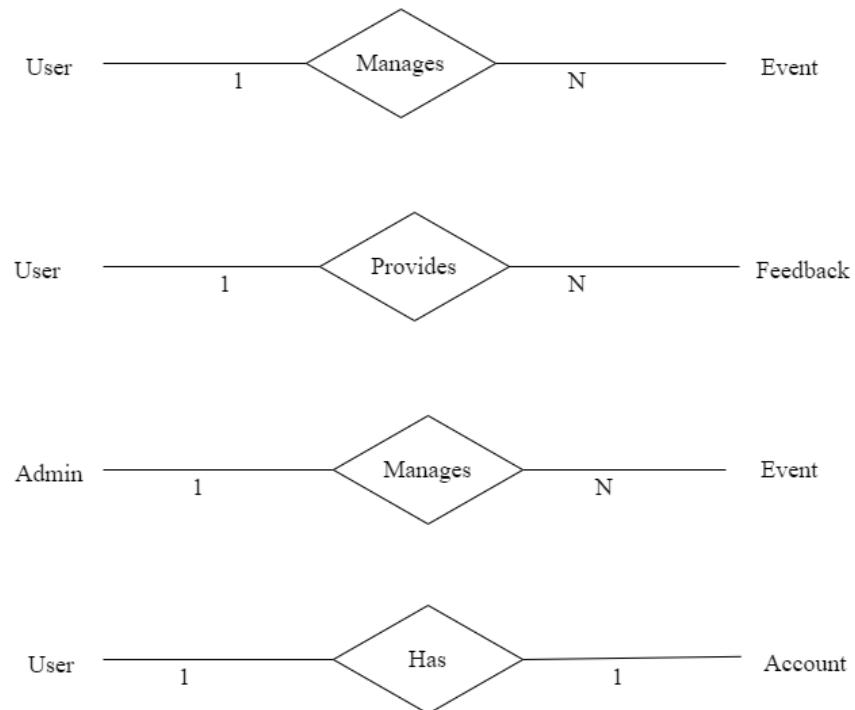
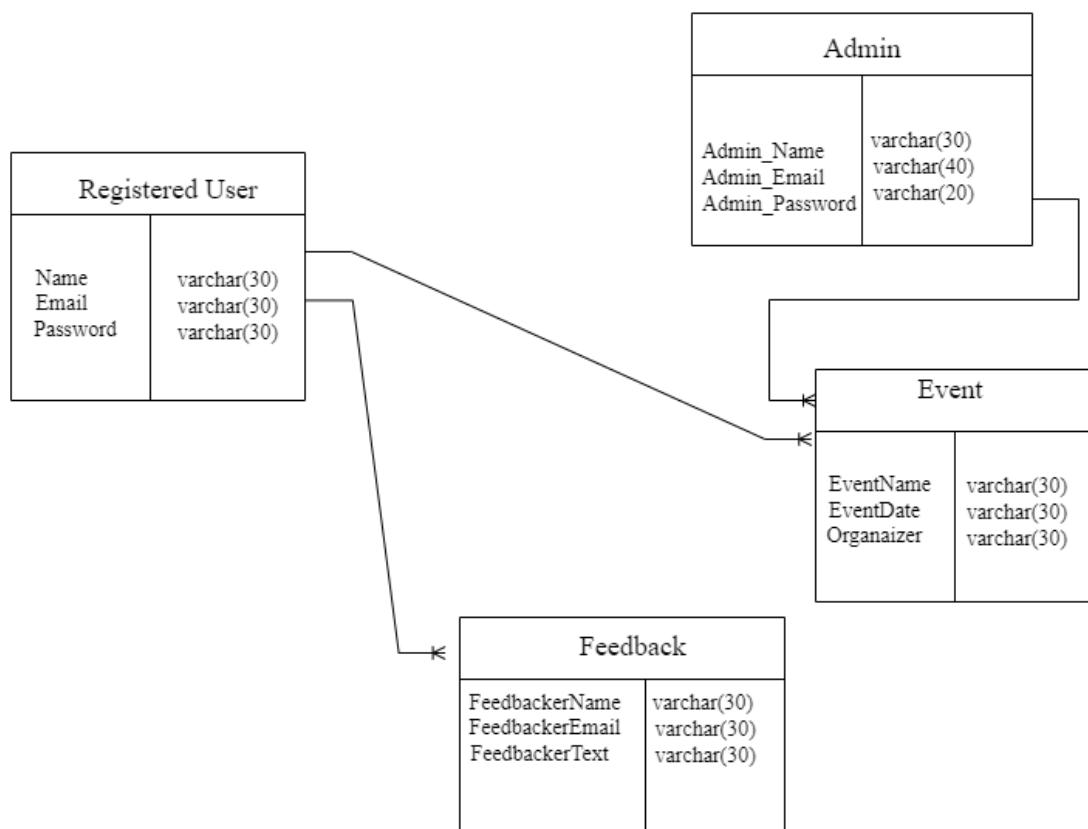


Figure 5.8: Relationships between different entities



## 5.3 API Documentation

### 5.3.1 User Authentication

#### Register

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:9090/logPOST`. The request body is a JSON object with fields `"email": "u1804020@student.cuet.ac.bd"` and `"password": "12sfdsfsd6"`. The response status is 400 Bad Request, with a duration of 56 ms and a size of 162 B. The response body contains the message `1 Incorrect password`.

Figure 5.10: Request and Response if registration successful

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:9090/logPOST`. The request body is a JSON object with fields `"email": "u1804020@student.cuet.ac.bd"` and `"password": "123456"`. The response status is 200 OK, with a duration of 46 ms and a size of 181 B. The response body contains the message `1 Login successful!`.

Figure 5.11: Request and Response if registration unsuccessful

## Log-in

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:9090/regPOST`. The request body contains the following JSON:

```
1
2
3   "name": "Tahsin Azad Tias",
4   "studentId": "1804067",
5   "phoneNo": "01947562345",
6   "email": "u1804067@student.cuet.ac.bd",
7   "password": "111111"
```

The response status is 200 OK, with a timestamp of 11.96 s and a size of 303 B. The response body is:

```
1
2
3   "id": 35,
4   "name": "Tahsin Azad Tias",
5   "studentId": "1804067",
6   "phoneNo": "01947562345",
7   "email": "u1804067@student.cuet.ac.bd",
8   "password": "111111"
```

Figure 5.12: Request and Response if login successful

The screenshot shows the Postman application interface. A POST request is being made to `http://localhost:9090/regP`. The request body contains the same JSON as Figure 5.12:

```
1
2
3   "name": "Tahsin Azad Tias",
4   "studentId": "1804067",
5   "phoneNo": "01947562345",
6   "email": "u1804067@student.cuet.ac.bd",
7   "password": "111111"
```

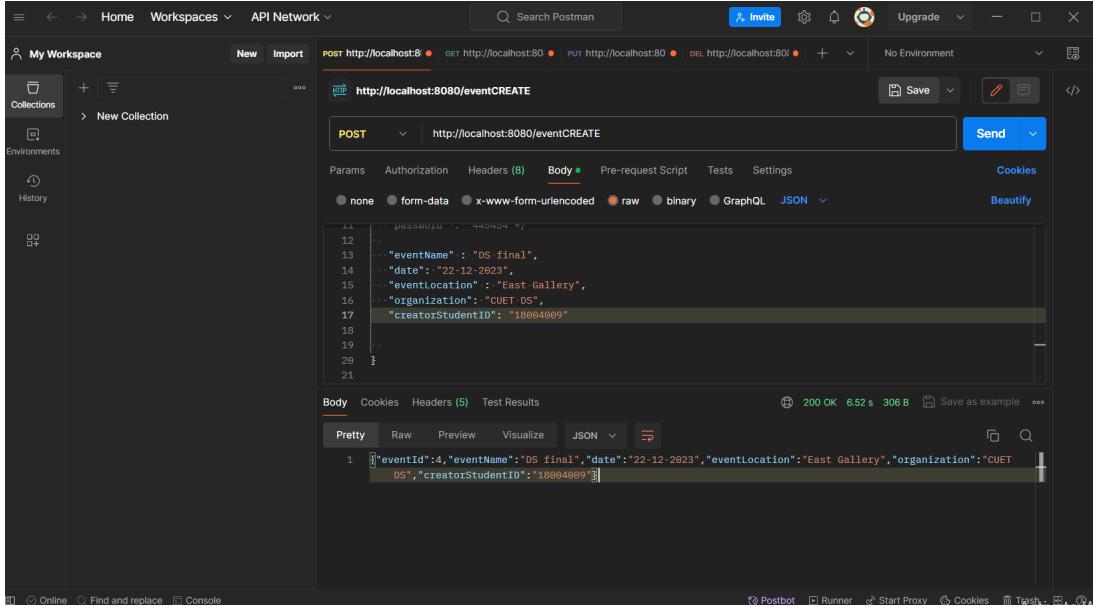
The response status is 404 Not Found, with a timestamp of 291 ms and a size of 353 B. The response body is:

```
1
2
3   "timestamp": "2024-01-14T18:46:26.421+00:00",
4   "status": 404,
5   "error": "Not Found",
6   "path": "/regP"
```

Figure 5.13: Request and Response if login unsuccessful

## 5.3.2 Event Management

### Event Post



The screenshot shows the Postman interface with a POST request to `http://localhost:8080/eventCREATE`. The request body is a JSON object:

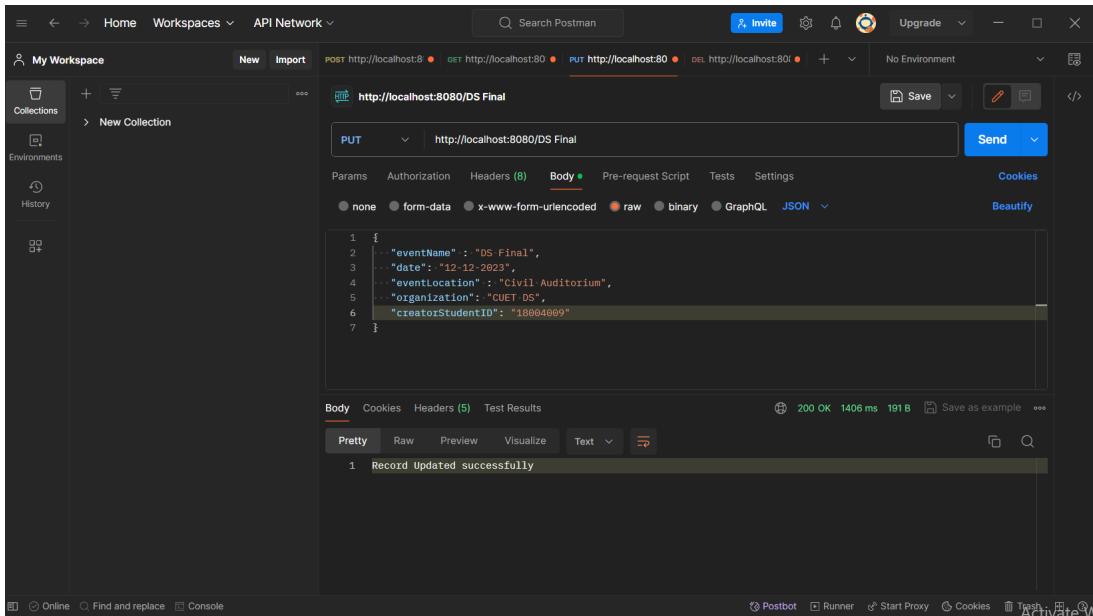
```
1 {  
2   "eventId": 4,  
3   "eventName": "DS final",  
4   "date": "22-12-2023",  
5   "eventLocation": "East Gallery",  
6   "organization": "CUET DS",  
7   "creatorStudentID": "18004069"  
8 }  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21
```

The response status is 200 OK with a time of 6.52 s and a size of 306 B. The response body is:

```
1 {  
2   "eventId": 4,  
3   "eventName": "DS final",  
4   "date": "22-12-2023",  
5   "eventLocation": "East Gallery",  
6   "organization": "CUET  
7   DS",  
8   "creatorStudentID": "18004069"  
9 }  
10  
11
```

Figure 5.14: Request and Response for event post

### Event Edit



The screenshot shows the Postman interface with a PUT request to `http://localhost:8080/DS Final`. The request body is a JSON object:

```
1 {  
2   "eventId": 4,  
3   "eventName": "DS Final",  
4   "date": "12-12-2023",  
5   "eventLocation": "Civil Auditorium",  
6   "organization": "CUET DS",  
7   "creatorStudentID": "18004069"  
8 }  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21
```

The response status is 200 OK with a time of 1406 ms and a size of 191 B. The response body is:

```
1 Record Updated successfully
```

Figure 5.15: Request and Response for event edit

## 5.4 GUI (Graphical User Interface) design

### 5.4.1 Homepage

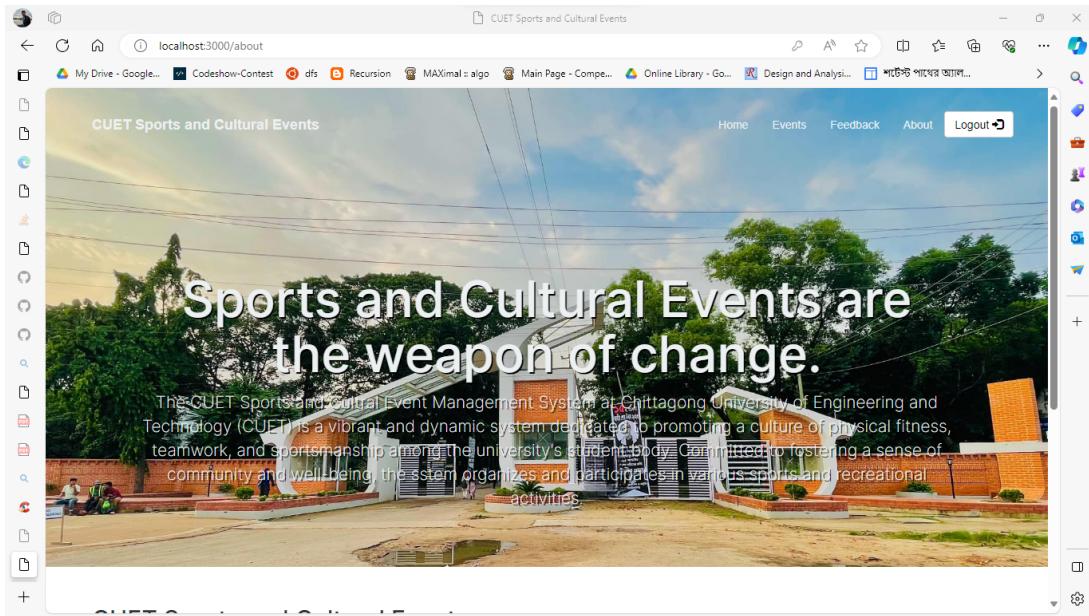


Figure 5.16: Homepage

### 5.4.2 List of upcoming events

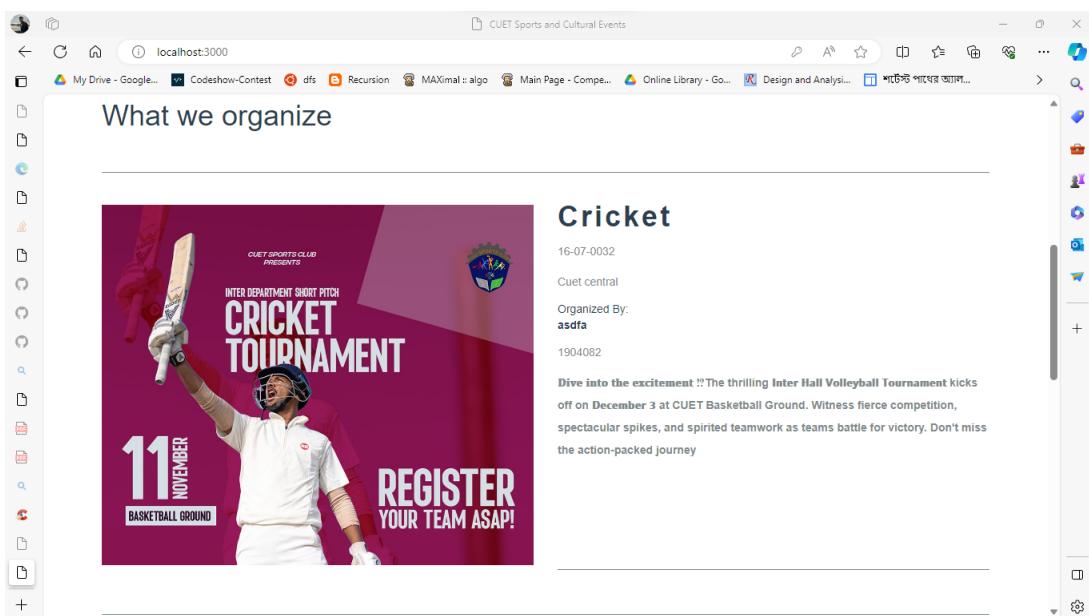


Figure 5.17: List of upcoming events

### 5.4.3 Create Events

The screenshot shows a web browser window titled "CUET Sports and Cultural Events" at "localhost:3000/viewEvents". The page displays a photograph of a university campus building with students in the foreground. Below the photo is a text banner: "teamwork, and sportsmanship among the university's student body. Committed to fostering a sense of community and well-being, the system organizes and participates in various sports and recreational activities." A message box says "Message goes here". A table lists three events:

Event ID	Event Name	Date	Event Location	Organization	Event Description	Options
6	Cricket	16-07-0032	Cuet central	asdfa	Dive into the excitement !The thrilling Inter Hall Volleyball Tournament kicks off on December 3 at CUET Basketball Ground. Witness fierce competition, spectacular spikes, and spirited teamwork as teams battle for victory. Don't miss the action-packed journey	<button>Delete</button> <button>Edit</button>
7	CPL	17-07-0007	CUET central field	CUET sports club	sdf	<button>Delete</button> <button>Edit</button>
8	CFL	16-07-0029	CUET central field	CUET sports club	sdf	<button>Delete</button> <button>Edit</button>

At the bottom left is a "Create Event" button. The footer contains address information, social media links (Facebook, Google+, YouTube), and contact details: Address: Kaptai, Highway রাউন্ডাবলি সড়ক, Chattogram 4349; Chittagong - 4349, Bangladesh; CALL TO. Tel: +880-31-714946, +880-31-714911; Mail: registrar@cuet.ac.bd, iict@cuet.ac.bd.

Figure 5.18: Create Events

### 5.4.4 Event creation form

The screenshot shows a web browser window titled "CUET Sports and Cultural Events" at "localhost:3000/createnewevent". The page displays a photograph of a university campus building. A form is centered on the page with fields for "Event Name", "Date", "Event Location", "Organization", "Creator Student ID", and "Description". A "Submit" button is at the bottom. The footer contains address information, social media links (Facebook, Google+, YouTube), and contact details: Address: Kaptai, Highway রাউন্ডাবলি সড়ক, Chattogram 4349; Chittagong - 4349, Bangladesh; CALL TO. Tel: +880-31-714946, +880-31-714911; Mail: registrar@cuet.ac.bd, iict@cuet.ac.bd.

Figure 5.19: Event Creation Form

### 5.4.5 Feedback Add

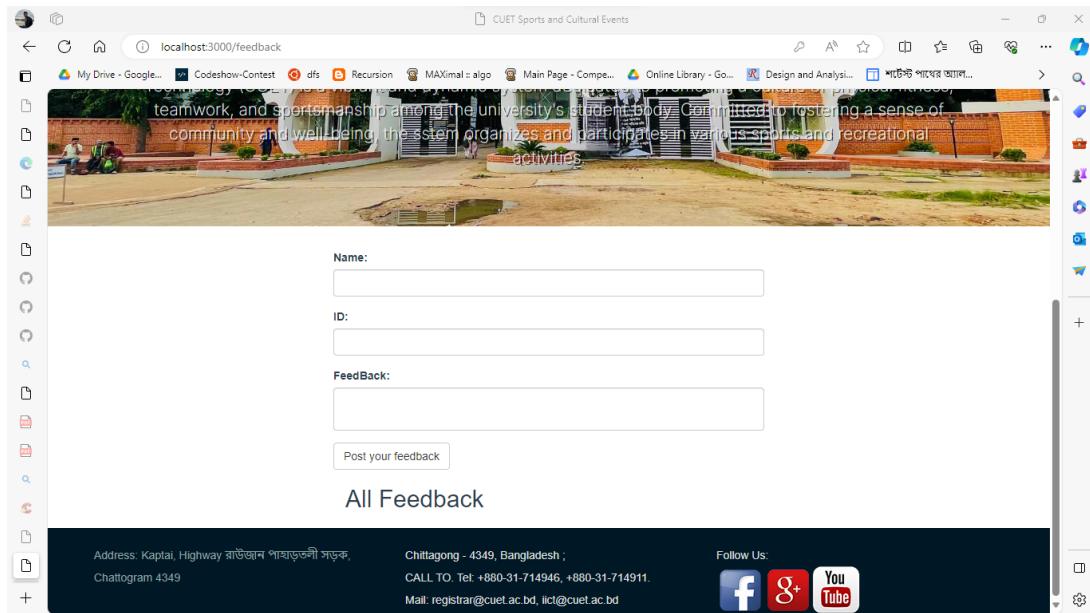


Figure 5.20: Add Feedback

### 5.4.6 Login

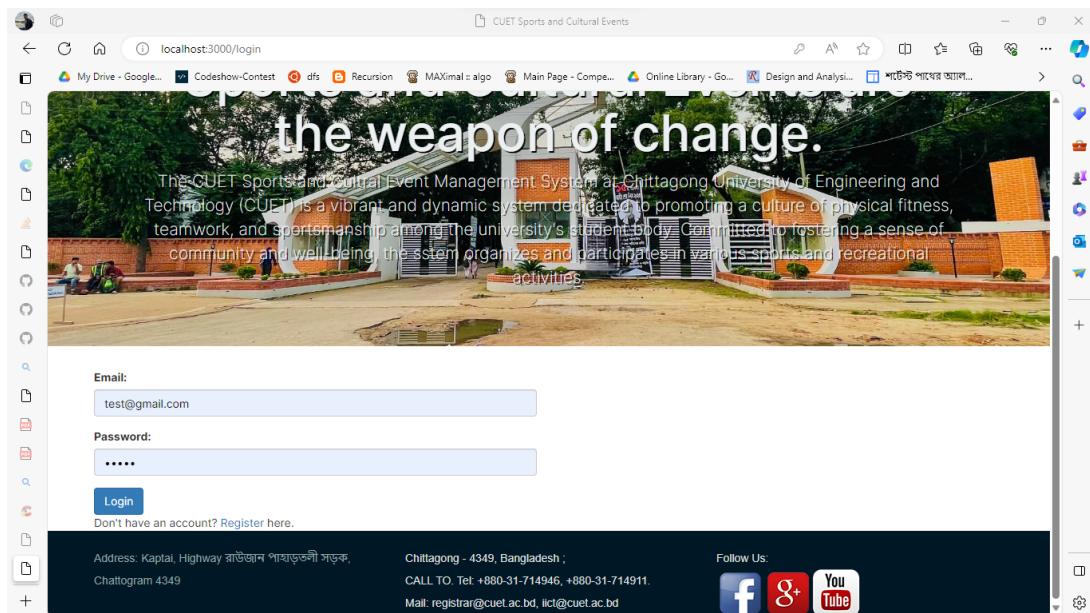


Figure 5.21: Login

### 5.4.7 Signup

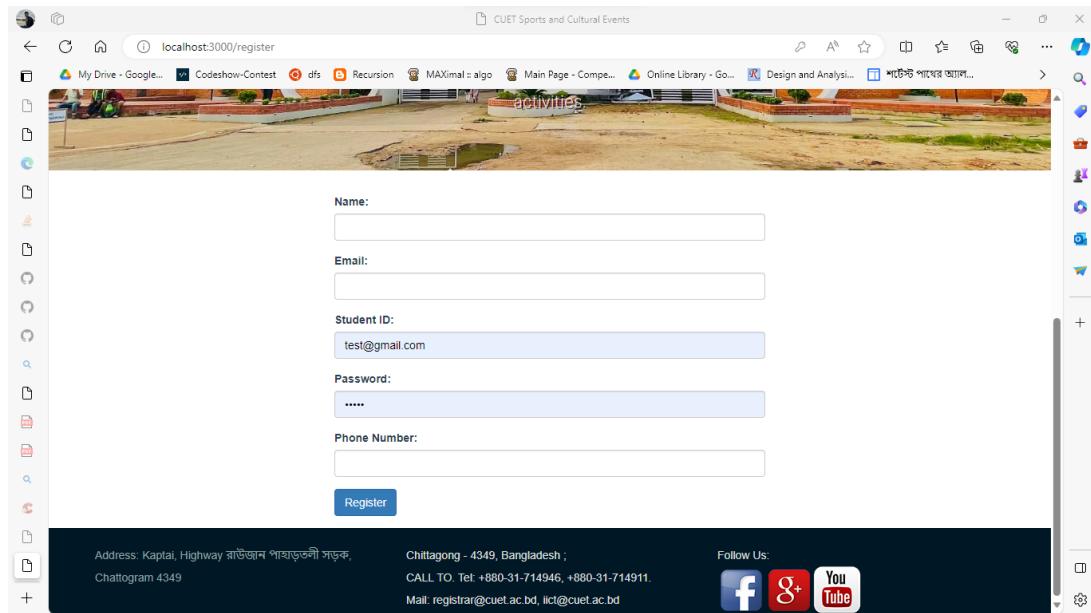


Figure 5.22: SignUp

# **Testing and sustainability plan**

## **6.1 Requirements/specifications-based system level test cases**

The CUET Sports and Cultural Event Management System project is geared towards providing a comprehensive platform enabling users to register, log in, manage their profile, events, and update their profiles. The primary focus is to ensure the system's robustness and reliability through a meticulous testing process encompassing unit, integration, system, and acceptance testing. The testing scope involves validating features such as user authentication, event and feedback management, user profile updates, and system notification functionalities. Additionally, performance, usability, and security aspects are thoroughly examined. This testing initiative aims to confirm that the system aligns with user expectations, adheres to specified criteria, and delivers a secure, user-friendly experience. A detailed table delineating system-level test cases, organized by requirements, is crucial for validating the system's reliability and conformance to specified criteria. Refer to Table for a breakdown of requirements and their corresponding test cases.

Table 6.1: Test Cases for CUET Sports and Cultural Event Management System

Requirement ID	Requirement Statement	Must/Want	Comment
R-Auth-01	Verify user registration process accepting only using CUET mail	Must	N/A
R-Auth-02	Display error message if the user and admin tries any wrong password or mail	Must	N/A
R-EM-01	Ensure registered user can create and manage events	Must	N/A
R-FM-01	Ensure registered user can post feedback, if not registered, give error message	Must	N/A
R-PM-01	Users must update their profiles with valid information	Must	N/A
R-Search-01	Enable users to search for event entries	Must	N/A
R-AM-01	Ensure the admin can manage the events' post the users' authentication	Must	N/A

<b>Project Name</b>	CUET Sports and Cultural Event Management System					
<b>Module Name</b>	Verify user registration process accepting only using CUET mail					
<b>Created By</b>	Tahsin Azad Tias					
<b>Reviewed By</b>	Md. Shahriar Ahmed, Jamiul Hasan Joy					
<b>Date of Creation</b>	28-11-23					
<b>Date of Review</b>	14-12-23					
Test Case ID	Scenario	Steps	Test Data	Expected Result	Actual Result	Status
R-Auth-01	Verify Registration	1. Goto register screen. 2.Enter information 3.Click Register button	Registration.js	Redirect to login screen	Redirect to the home page	Passed
R-Auth-01	Verify Login	1. Goto login screen. 2.Enter information 3.Click login button	Login.js	Redirect to Home screen	Redirect to the Home page	Passed

Table 6.2: Requirements/specifications-based system level test cases (02)

<b>Project Name</b>	CUET Sports and Cultural Event Management System					
<b>Module Name</b>	Display error message if the user and admin tries any wrong password or mail					
<b>Created By</b>	Md. Shahriar Ahmed					
<b>Reviewed By</b>	Tahsin Azad Tias, Jamiul Hasan Joy					
<b>Date of Creation</b>	18-11-23					
<b>Date of Review</b>	20-12-23					
Test Case ID	Scenario	Steps	Test Data	Expected Result	Actual Result	Status
R-Auth-02	Verify request post	1. Goto Post screen. 2.Enter information 3.Click Post button	Post.js	Redirect to Home screen and show post	Redirect to the Home page and show post	Passed

Table 6.3: Requirements/specifications-based system level test cases (03)

<b>Project Name</b>	CUET Sports and Cultural Event Management System					
<b>Module Name</b>	Ensure registered user can create and manage events					
<b>Created By</b>	Jamiul Hasan Joy					
<b>Reviewed By</b>	Tahsin Azad Tias, Jamiul Hasan Joy					
<b>Date of Creation</b>	22-12-23					
<b>Date of Review</b>	27-12-23					
Test Case ID	Scenario	Steps	Test Data	Expected Result	Actual Result	Status
R-EM-01	Available Members	1. Goto Members screen.	Members.js	Show all available Members	Show all available Members	Passed

Table 6.4: Requirements/specifications-based system level test cases (04)

<b>Project Name</b>	CUET Sports and Cultural Event Management System					
<b>Module Name</b>	Ensure registered user can post feedback, if not registered, give error message					
<b>Created By</b>	Tahsin Azad Tias					
<b>Reviewed By</b>	Md. Shahriar Ahmed, Jamiul Hasan Joy					
<b>Date of Creation</b>	2-1-24					
<b>Date of Review</b>	11-1-24					
Test Case ID	Scenario	Steps	Test Data	Expected Result	Actual Result	Status
R-FM-01	View Listings	1. Goto Post screen.	Listings.js	Show user all the requests for Gate Pass	Show user all the requests for Gate Pass	Passed

Table 6.5: Requirements/specifications-based system level test cases (05)

<b>Project Name</b>	CUET Sports and Cultural Event Management System					
<b>Module Name</b>	Users must update their profiles with valid information					
<b>Created By</b>	Md. Shahriar Ahmed					
<b>Reviewed By</b>	Tahsin Azad Tias, Jamiul Hasan Joy					
<b>Date of Creation</b>	12-1-24					
<b>Date of Review</b>	22-1-23					
Test Case ID	Scenario	Steps	Test Data	Expected Result	Actual Result	Status
R-PM-01	Viewing Roaster	1. Goto roaster screen. 2. View all available posts where members assigned	Roaster.js	Show updated posts to members	Show updated posts to members	Passed

Table 6.6: Requirements/specifications-based system level test cases (06)

## 6.2 Traceability of test cases to use cases

Test Case ID	R-Auth-01	R-Auth-02	R-EM-01	R-FM-01	R-PM-01
R-Auth-01	✓				
R-Auth-02	✓	✓			
R-EM-01			✓	✓	
R-FM-01		✓	✓	✓	
R-PM-01					✓

Table 6.7: Traceability of test cases to use cases

## 6.3 Techniques used for test generation

In our study, two methodologies were implemented for the generation of test cases, namely:

- White Box Testing Technique: Basis Path Testing
- Black Box Testing Technique: Equivalence Partitioning.

### Basis Path Testing

Basis Path Testing technique enables the test-case designer to derive a quantitative measure of the logical complexity of a procedural design through analysis of the flow graph. This measure serves as a guide for determining a foundational set of execution paths. The test cases derived to verify the foundational set ensure that every statement in the program is executed at least once during the testing process.

### Equivalence Partitioning

Equivalence Partitioning method which falls under the category of Black Box Testing techniques, involves the division of the input domain of a program into distinct classes of data. This division serves as the basis for the derivation of test cases. The design of test cases using Equivalence Partitioning is premised on an analysis of the equivalence classes for a given input condition. An equivalence class represents a set of valid or invalid states for the input conditions. In the case of our system, the input condition is Boolean, which defines a single valid class and a single invalid class.

## 6.4 Assessment of the goodness of your test suite

The testing phase of the CUET Sports and Cultural Event Management System requires a comprehensive and systematic approach in order to ensure the accuracy and reliability of the software. Both white box testing and black box testing are performed to thoroughly evaluate the software. The basis path testing method is utilized for white box testing, as it allows for a logical complexity analysis of the software's procedural design

and serves as a guide for defining a set of execution paths. The complex interplay of components in the CUET Sports and Cultural Event Management System necessitates the use of the basis path testing method. Complementary to the white box testing, black box testing focuses on the functional requirements of the software. The equivalence partitioning technique is used to evaluate the various input conditions in the CUET Sports and Cultural Event Management System. This technique involves dividing each input condition into two equivalence classes, valid and invalid, in order to thoroughly test the software's functionality. This comprehensive testing approach, utilizing both white box and black box testing methods, will ensure the robustness and reliability of the CUET Sports and Cultural Event Management System.

## 6.5 Sustainability Plan

The determinants of a system's sustainability are collectively referred to as a sustainability plan. The principle objective of such a plan is to facilitate a balanced and optimal rate of development that can be sustained over a prolonged period of time. Optimal development does not imply the fastest rate of progress, but rather a rate that strikes a balance between immediate and long-term requirements. The essence of sustainable development lies in the achievement of efficiency and the maintenance of a harmonious equilibrium between short-term and long-term considerations.

### 6.5.1 Scalability

The scalability of software is an important metric used to evaluate its capability for improvement. A software system is considered scalable if it is able to perform efficiently and effectively, both when serving a small number of users and a large number of users. The CUET Sports and Cultural Event Management System. provides ample facilities and enhances its agility, thereby creating a platform that reduces maintenance costs while improving user experience. This allows the application to accommodate a growing number of users simultaneously and perform optimally even when subjected to an increasing workload or expanded scope. The scalability of the software is therefore an essential characteristic that ensures its continued effectiveness and efficiency under varying conditions.

### 6.5.2 Flexibility / Customization

It's great to see that the ToyFriends application has been designed with flexibility in mind. Using this app like Flutter and a powerful development tool like Visual Studio Code certainly helps in this regard. Additionally, incorporating a server solution like Firebase allows for flexibility and easy modification as the user needs change over time. Having a straightforward and reliable design is also important for ensuring that any necessary changes can be made quickly and efficiently. This can ultimately save time and resources in the long run, as it eliminates the need for extensive redesigns or rebuilding. Overall, it's important to strike a balance between having a well thought out, reliable design and the ability to adapt to future changes. The CUET Sports and Cultural Event Management System seems to have done this well, and it's great to see the consideration given to the user's future needs

# Acknowledgement

We express our sincere gratitude for the dedicated efforts and valuable time generously contributed by our course teachers:

**Mir. Md. Saki Kowsar**

Assistant Professor,

Department of Computer Science and Engineering, CUET

**Moumita Sen Sarma**

Lecturer,

Department of Computer Science and Engineering, CUET

Their invaluable guidance and feedback have been instrumental in the successful completion of this project.

# References

- [1] Software Engineering: A Practitioner's Approach, 9th Edition By Roger Pressman and Bruce Maxim
- [2] <https://www.geeksforgeeks.org/>
- [3] <https://app.diagrams.net/>
- [4] "Postman",<https://www.postman.com/>
- [5] Soft Robotics", <https://www.softrobotics.com.bd/>
- [6] <https://www.tutorialride.com/software-engineering/evolutionary-process-models-in-software-engineering.htm>