North South University

Department of Electrical & Computer Engineering (ECE)

Software Requirements Specification (SRS)

Student IDs

Md. Abul Bashar Nirob – 2022198042

Istiak Ahasan – 2010282042

Email Addresses:

abul.nirob@northsouth.edu

istiak.ahasan@northsouth.edu

Campus Event Management System (CEMS)

Prepared By: Md Abul Bashar Nirob

Date: 17-07-25

Version: 1.0

---

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the Campus Event Management System (CEMS). The system is designed to facilitate the management of events within a campus environment, providing distinct functionalities for three user roles: general users (students), event organizers, and system administrators. This document outlines the functional and non-functional requirements of the system, as well as models to depict the architecture and flow of information.

1.2 Scope

The CEMS is a web-based platform that aims to simplify and digitize the process of managing campus events. Users can view, register for, and provide feedback on events. Organizers can create, manage, and track their events while reviewing participant data and user feedback. Administrators are given overarching control of the platform, with capabilities to manage users, events, and organizational data. The system ensures efficient interaction between the various stakeholders involved in campus event management.

1.3 Definitions, Acronyms, and Abbreviations

- **CEMS:** Campus Event Management System
- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **CRUD:** Create, Read, Update, Delete

1.4 Intended Audience and Reading Suggestions

This document is intended for:

- ✓ **Developers:** For understanding what needs to be implemented.
- ✓ **Testers:** To prepare test cases and verify system functionality.
- ✓ **Project Managers:** To track deliverables and milestones.
- ✓ **Stakeholders:** To ensure business requirements are addressed.

1.5 Document Overview

This document is divided into sections that describe the system's overall description, functional and non-functional requirements, UML diagrams, and additional models.

---

2. Overall Description

2.1 Product Perspective

CEMS is a standalone, full-stack web application designed to support campus-based event activities. It consists of:

- A front-end interface for users, organizers, and admins.
- A back-end server that handles logic, data processing, and database transactions.
- A centralized database storing users, events, feedback, and participant data.

2.2 Product Functions

- ✓ **User Functions:** Register, login, browse upcoming and past events, submit feedback.
- ✓ **Organizer Functions:** Create and manage events, view feedback and participant lists.
- ✓ **Admin Functions:** Manage user and event data, including deletion, updates, and creation of new entries.

2.3 User Classes and Characteristics

Role	Capabilities
User	Register, login, view events, submit feedback
Organizer	Login, create/manage events, view participants, review feedback
Admin	Full CRUD capabilities on users and events

2.4 Operating Environment

- **Client Side:** Compatible with modern browsers (Chrome, Firefox, Edge)
- **Server Side:** Hosted on a web server using technologies such as Node.js or Django
- **Database:** MySQL or PostgreSQL

2.5 Design and Implementation Constraints

- ❖ Must be responsive across devices.
- ❖ User passwords must be securely encrypted.
- ❖ Only authenticated users can access role-specific dashboards.

---

3. Functional Requirements

3.1 User Registration and Login

- Users must provide name, email, phone number, and password.
- Successful login redirects users to their appropriate dashboards.
- Authentication mechanisms should ensure secure login.

3.2 Event Management

- ✓ Events must be categorized as “Upcoming” or “Past”.
- ✓ Events should include title, description, date/time, and organizer.
- ✓ Users can view all events and register or provide feedback.

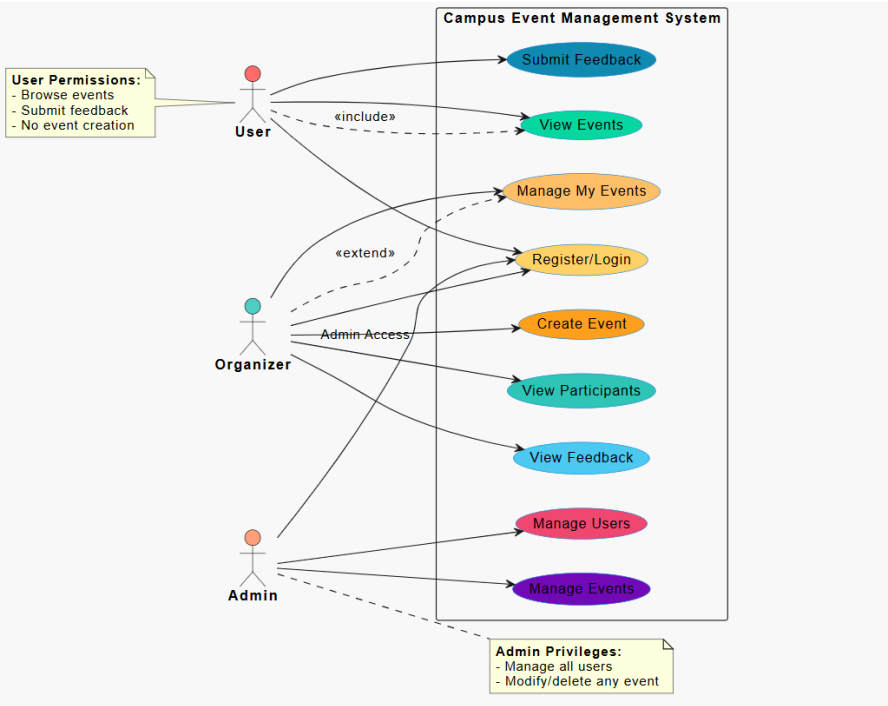
3.3 Feedback Submission

- ➢ After attending an event, users can submit feedback.
- ➢ Feedback is visible to event organizers and admins.

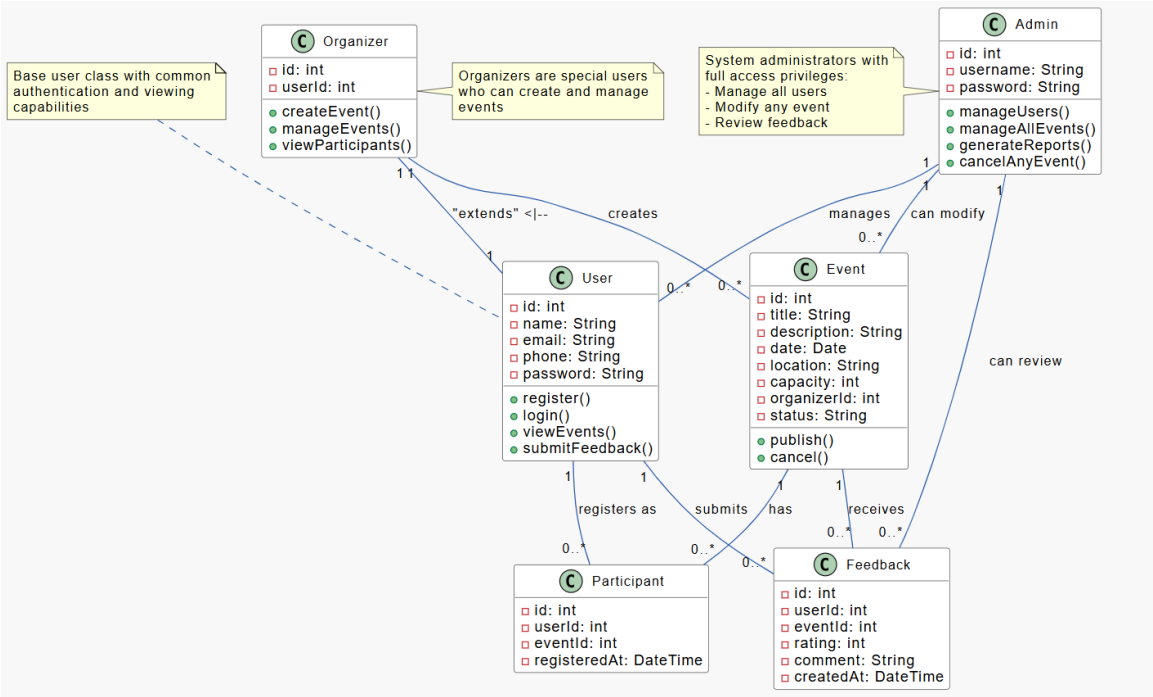
3.4 Role-Based Access Control

- ❖ Only organizers can create events.
- ❖ Only admins can manage users and events not created by them.

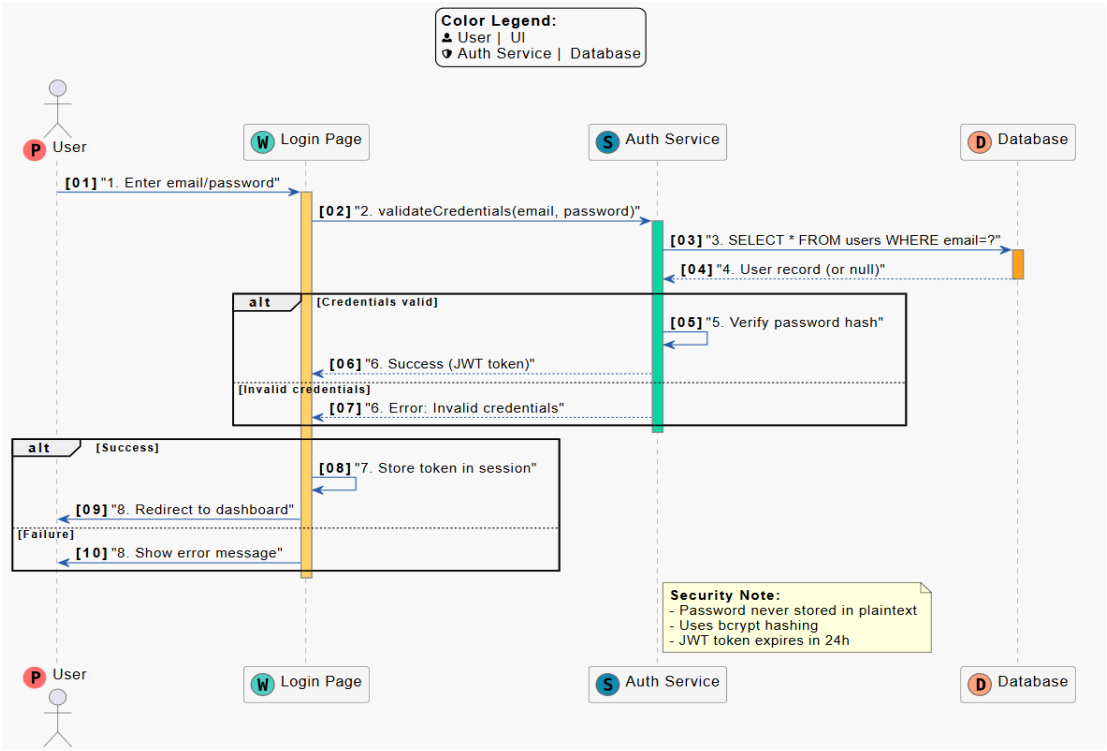
4. UML Use Case Diagram



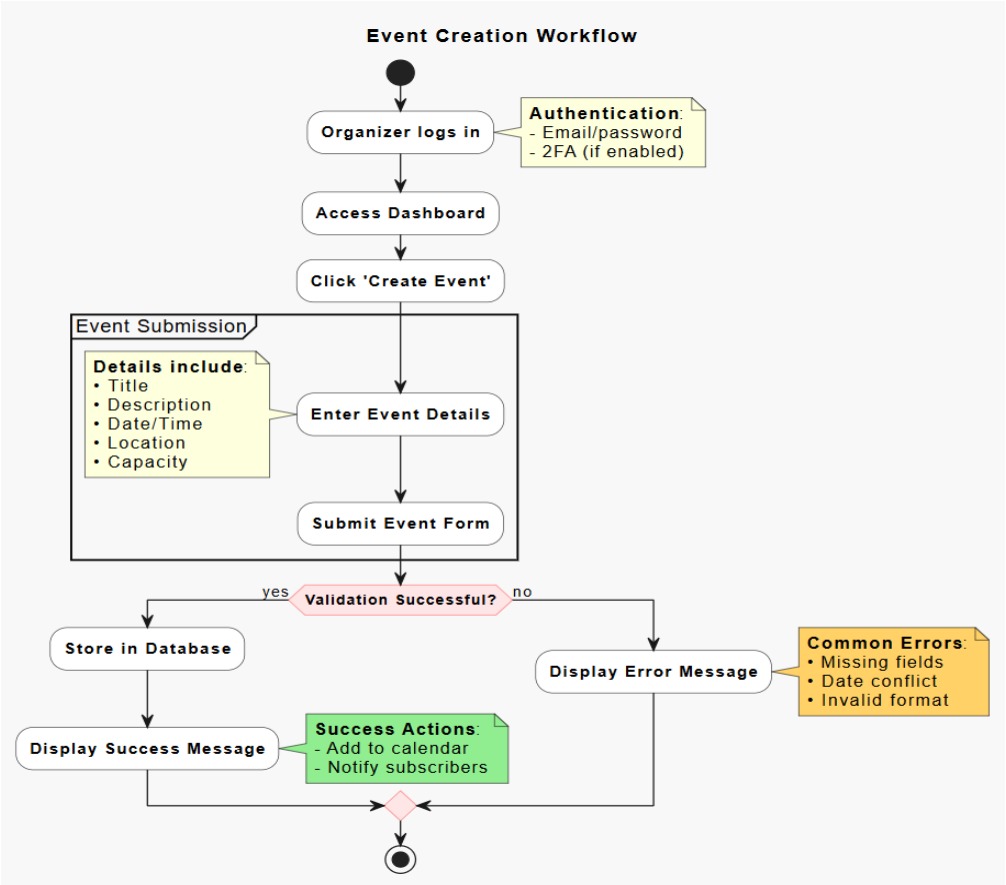
5. UML Class Diagram



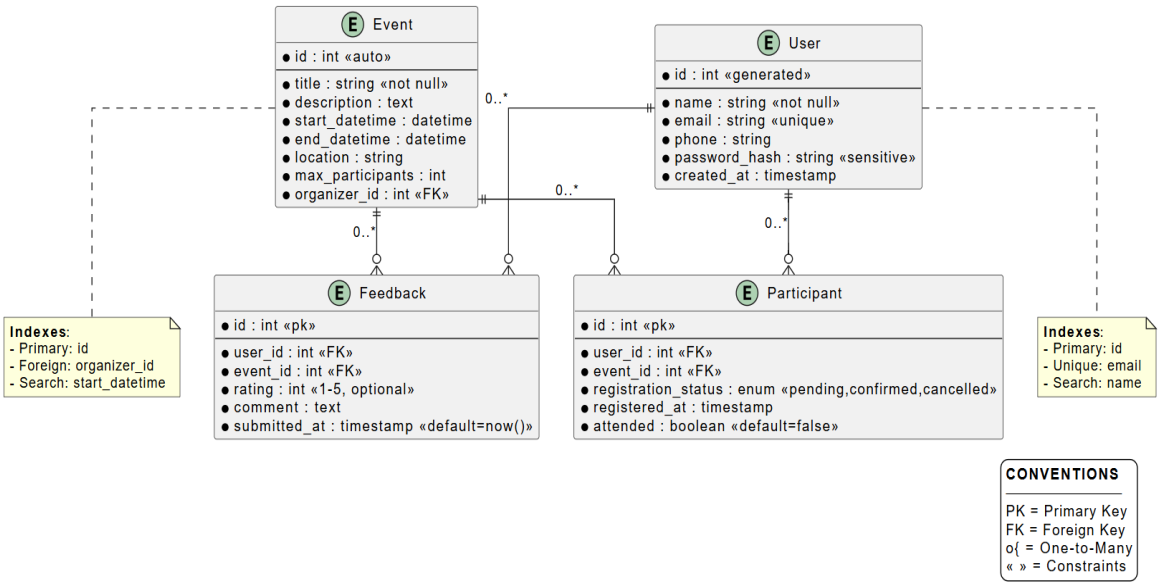
6. Sequence Diagram: Login Flow



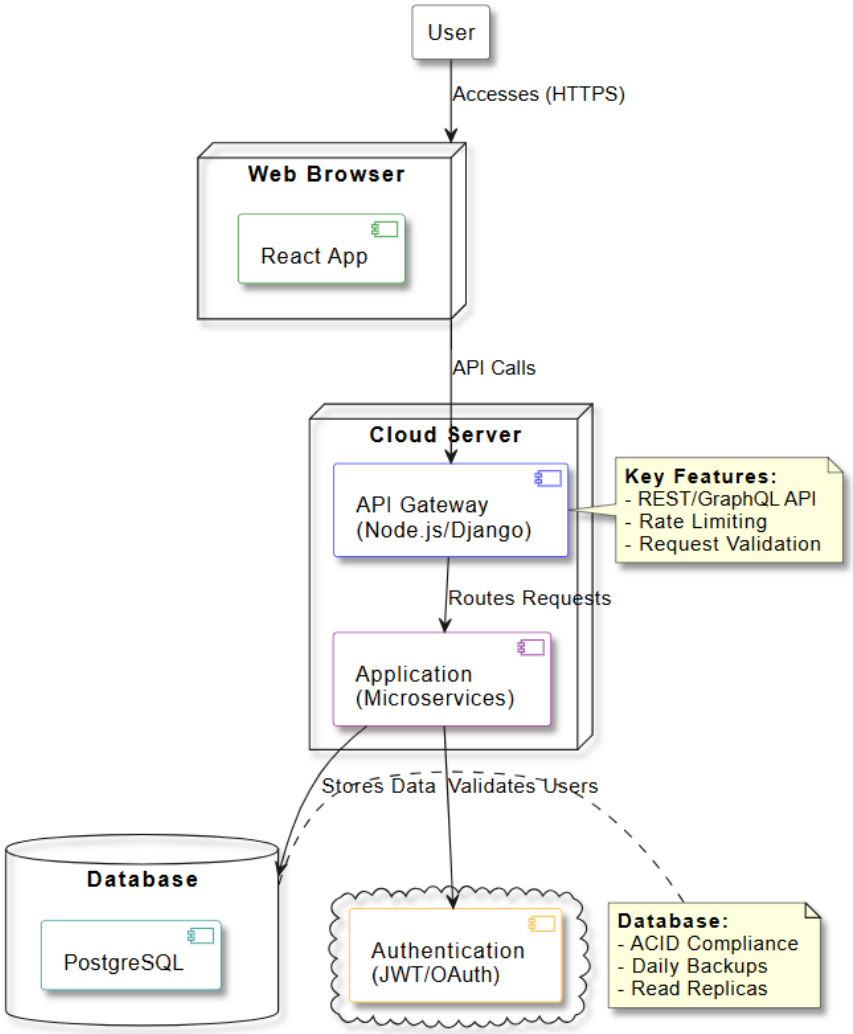
7. Activity Diagram: Event Creation



8. ER Diagram (Entity Relationship Diagram)



9. Deployment Diagram



---

10. Non-Functional Requirements

10.1 Performance Requirements

- ❖ System must support at least 100 concurrent users.
- ❖ Response time must not exceed 2 seconds under normal load.

10.2 Security Requirements

- ✓ Use HTTPS to secure communications.
- ✓ Store passwords using a secure hashing algorithm like bcrypt.
- ✓ Implement proper session management and timeout mechanisms.

10.3 Usability Requirements

- Simple and intuitive UI layout.
- Accessible from all standard web browsers.

10.4 Availability and Reliability

- ✓ System must have an uptime of 99.5% monthly.
- ✓ Automatic daily backups should be configured.

10.5 Maintainability

- ➢ Code should follow standard naming conventions and design principles.
- ➢ Detailed documentation should be available for future maintenance.

---

11. External Interface Requirements

11.1 User Interfaces

- Registration, login, dashboard, and event browsing pages.
- Admin panel with full control features.

11.2 Hardware Interfaces

- ❖ Compatible with standard desktops and mobile devices.
- ❖ Requires stable internet connection and a modern browser.

11.3 Software Interfaces

- ✓ **Frontend:** HTML, CSS, JavaScript (React or similar framework)
- ✓ **Backend:** Node.js, Django, or equivalent REST API
- ✓ **Database:** MySQL/PostgreSQL

---

12. Future Enhancements

- **Event Ticketing Integration:** Enable ticket generation and scanning.
- **Email Notifications:** Send reminders and confirmations via email.
- **Search and Filter:** Implement dynamic filtering and search features.
- **Mobile App:** Native mobile support for iOS and Android.