SOFTWARE REQUIREMENTS SPECIFICATION(SRS)

For

CAMPUS EVENT MANAGEMENT SYSTEM

Deshna Thunga - CS22B1020
Avanee Lakare - CS22B1080
Nimisha Thallapally - CS22B1082
Jyolsna Joji Isac - CS22B2034
Megha Hebbar - CS22B2039

Contents

1 Introduction

The Campus Event Management System Software Requirements Specification (SRS) outlines the essential functionalities and specifications for developing a centralized platform to manage and showcase events within a campus environment.

1.1 Purpose

The project seeks to establish a centralized hub where students and staff of the Institute can access information regarding events like cultural festivities, academic seminars, sports competitions, and alumni gatherings taking place on campus. By consolidating event details into a single platform, the aim is to provide easier access to event information and a wider reach to the intended audience. Ultimately, the project endeavors to enhance the overall campus experience by facilitating greater participation, connectivity, and enjoyment among students, faculty, staff, and alumni.

1.2 Intended Audience

The Campus Event Management System (CEMS) is designed to cater to two main user groups:

- **Event Organizers:** These users are responsible for creating and managing events within the system, handling attendee registrations, managing event details comprehensively, and effectively utilizing event management features.

- **Students of the Institute:** This group constitutes the primary user base who are primarily interested in viewing different events taking place on campus, receiving event notifications, and staying updated with event-related information.

1.3 Project Scope

The Campus Events Management System (CEMS) is envisioned as a dynamic platform that serves as the central hub for all campus events, ensuring widespread visibility and engagement among the student, teacher, and faculty communities. With a user-friendly interface, all students will have unrestricted access to both upcoming and past events that have transpired within the current academic year. Each event listing will encompass essential details including date, time, duration, hosting club, departmental affiliation, and venue. Events will be categorized into two main types: interactive events, which feature forms to facilitate students to register for events, and declarative events, which serve solely to inform students about event details. Users will have the ability to filter events based on departmental categorization or date, providing tailored event views for

specific interests. Furthermore, a dedicated section will spotlight events scheduled for the current day, ensuring effortless accessibility for students. Upon account authentication, students will be empowered to curate their event experience by selecting and saving favorite events for future reference, with timely notifications to remind them of upcoming favorites. Only designated departments and clubs will possess the authority to publish events pertinent to their organization.To optimize event coordination, the system will enforce proper venue allocation, preventing scheduling conflicts where multiple events cannot be assigned to the same venue simultaneously. Through these comprehensive functionalities, the CEMS aims to streamline event management processes, foster community engagement, and enhance the overall campus experience for all stakeholders.

1.4 Overview

The remainder of this document elaborates on the general description of the system, specific requirements, system models, data requirements, human interface requirements, system features, and appendices providing a comprehensive understanding of CEMS.

1.5 References

- Fundamentals of Database System, Shamkant Navathe and Ramez Elmasri.

2 Functional Requirements

2.1 Event Management

2.1.1 Event Creation:

Accounts with authorization can create new events with details like title, description, date, time, location, category, and organizer information. Optional features include attaching documents, adding event images, and specifying registration requirements.

2.1.2 Event Categorization:

Ability to categorize events based on types such as cultural, academic, sports, technical, alumni engagements, and hostel activities.

2.1.3 Event Details Management:

Comprehensive management of event details including speakers, agenda, and related resources. Options to edit, update, or cancel events as needed.

2.2 Participant Registration

The system should include the capability for event organizers to provide specific form links for event registration. This functionality allows organizers to utilize external registration forms if needed, enhancing flexibility and accommodating various registration processes.

2.3 Event Favoriting:

Event favoriting is a feature that allows users to mark events of interest, creating a personalized list within their account. This feature enhances user engagement by providing easy access to events they wish to attend or keep track of. Users can conveniently revisit their favorite events from their profile or dashboard, promoting continued interaction with the platform. Once the event is completed that particular event will be unfavorited from the user's account so that the user has an updated favorites list of events.

2.4 Venue Management and Reservation:

The system enables users to book and manage event venues seamlessly. It provides real-time updates on venue availability, allowing users to check the status instantly. Additionally, the system offers reservation options, empowering users to reserve venues efficiently based on their event requirements.

2.5 Notification Management:

The system features automated notifications to remind users about their favorite upcoming events and provide updates regarding any changes or additions. These notifications are sent based on user preferences, allowing individuals to customize their notification settings according to their preferences.

2.6 Views:

Users will be able to view the list of events of that particular day on the website as default. They will be able to view a list of upcoming events, including event details such as title, description, date, time, location, and department. The system should allow users to filter events based on departments and other relevant criteria to find specific events of interest.

2.7 Filtering events:

Users will have the ability to filter events based on specific departments of their choice as well as the range of dates they wish to view the scheduled events. By selecting their preferred departments, users can narrow down their search to focus on events relevant to their interests or responsibilities. This targeted approach ensures user flexibility.

2.8 Accessibility and Security:

User authentication and authorization mechanisms to ensure data privacy and access control. The platform is open to everyone on a read-only view. But only authenticated users can access additional features such as event creation, registration, favoriting, and venue management.

2.9 Integration:

Integration with existing campus systems, including student databases, calendars, and communication platforms, to ensure smooth operation and a cohesive user experience. The Campus Events Management System (CEMS) should seamlessly sync with student databases for accurate user profiles, utilizing academic calendars to avoid scheduling conflicts, and leverage communication platforms for efficient event notifications and updates. This integration should streamline administrative tasks, enhance data accuracy, and optimize communication, ultimately enriching the user experience for all campus stakeholders.

3 Non-functional Requirements

3.1 Performance

- The system should be able to handle a high volume of users and event listings without significant performance degradation.
- Event listing pages should load quickly.

3.2 Availability

- The software should be available to users 24/7 with minimal downtime for maintenance.
- The system should have mechanisms for disaster recovery in case of outages.

3.3 Scalability

- The database and server infrastructure should be able to handle increased demand.

3.4 Security

- The system should protect user data (e.g., usernames, passwords, event details) from unauthorized access.

3.5 Usability

- The user interface should be intuitive and easy to navigate for all user types (students, faculty, staff).

3.6 Maintainability

- The system code should be well-documented and easy to understand for future maintenance and updates.

3.7 Monitoring

- Real-time monitoring should be implemented to identify and address any performance issues.

3.8 Flexibility

- Empowering users with flexibility to customize event filters according to their preferences and

4 Glossary

- **CEMS:** Campus Event Management System
- **SRS:** Software Requirement Specification
- **ER:** Entity Relationship

5 Constraints

5.1 Technology Stack Compatibility

The system must be compatible with the technology for development, including programming languages, frameworks, databases, and hosting platforms.

5.2 Data Privacy Regulations

CEMS must comply with data privacy laws and regulations, ensuring the protection of user data and sensitive information.

5.3 Performance Benchmarks

The system should meet specified performance benchmarks, such as response times, concurrent user handling, and system scalability, to ensure optimal performance under varying loads.

5.4 Security Measures

Implementation of robust security measures, including, secure user authentication, access control mechanisms,to protect the system from unauthorized access and data breaches.

6 Conclusion

This document has outlined SRS for a Campus Event Management System. The SRS defines the functionalities, user roles, and non-functional requirements for the system. Considering these requirements will ensure the CEMS effectively addresses the needs of the campus community by providing a central platform to discover, manage, and participate in events.

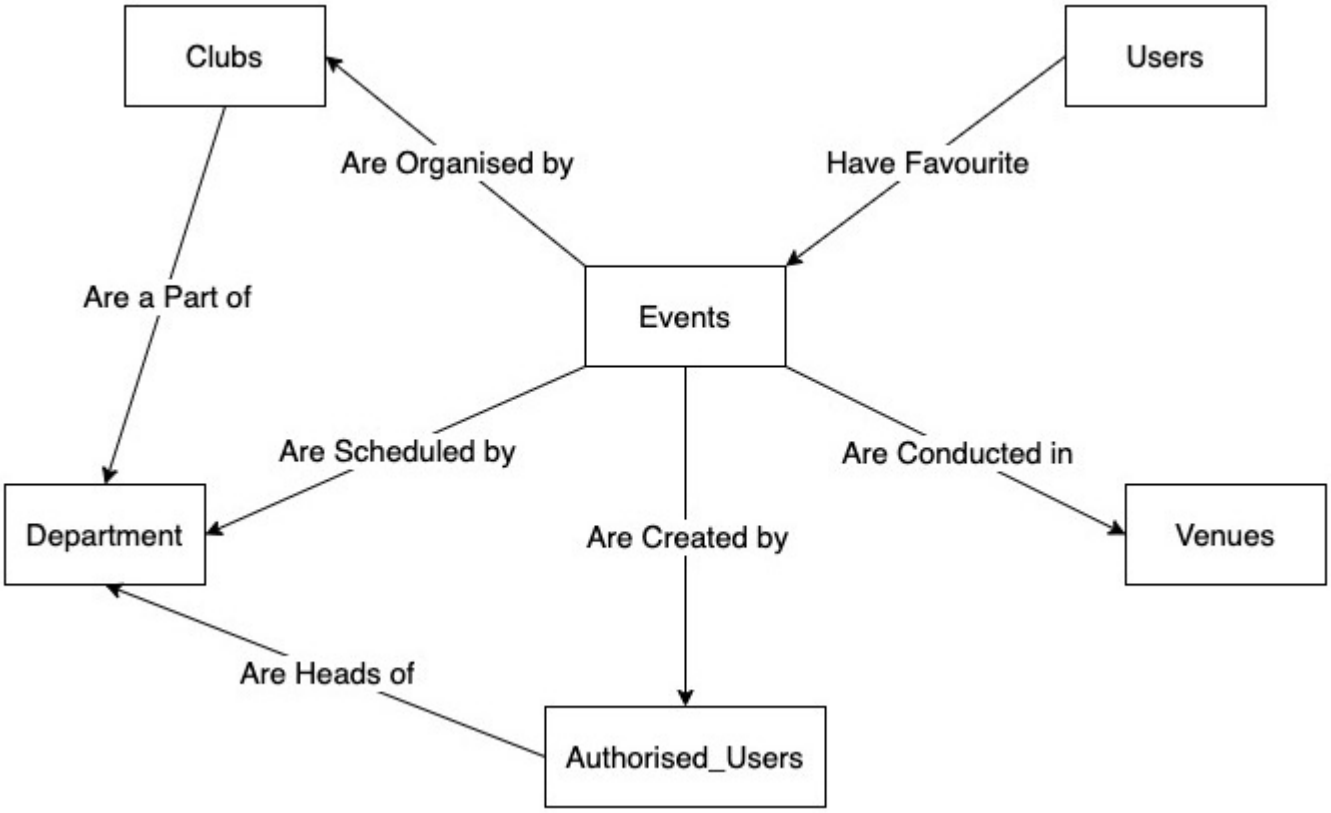

Figure 1: Conceptual Data Model

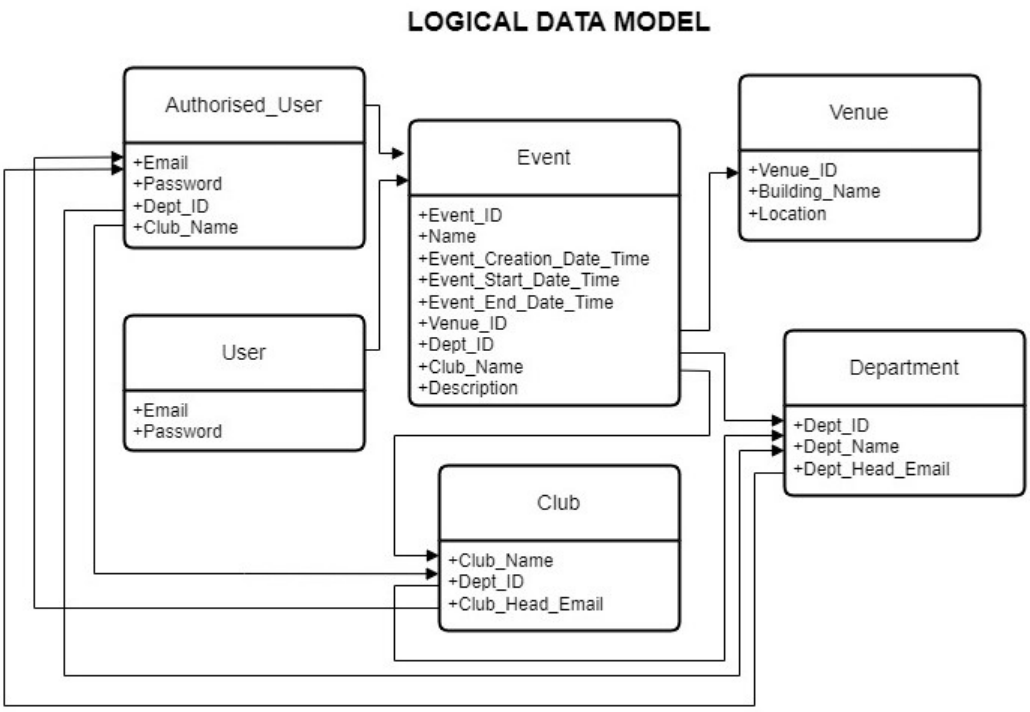

Figure 2: Logical Data Model

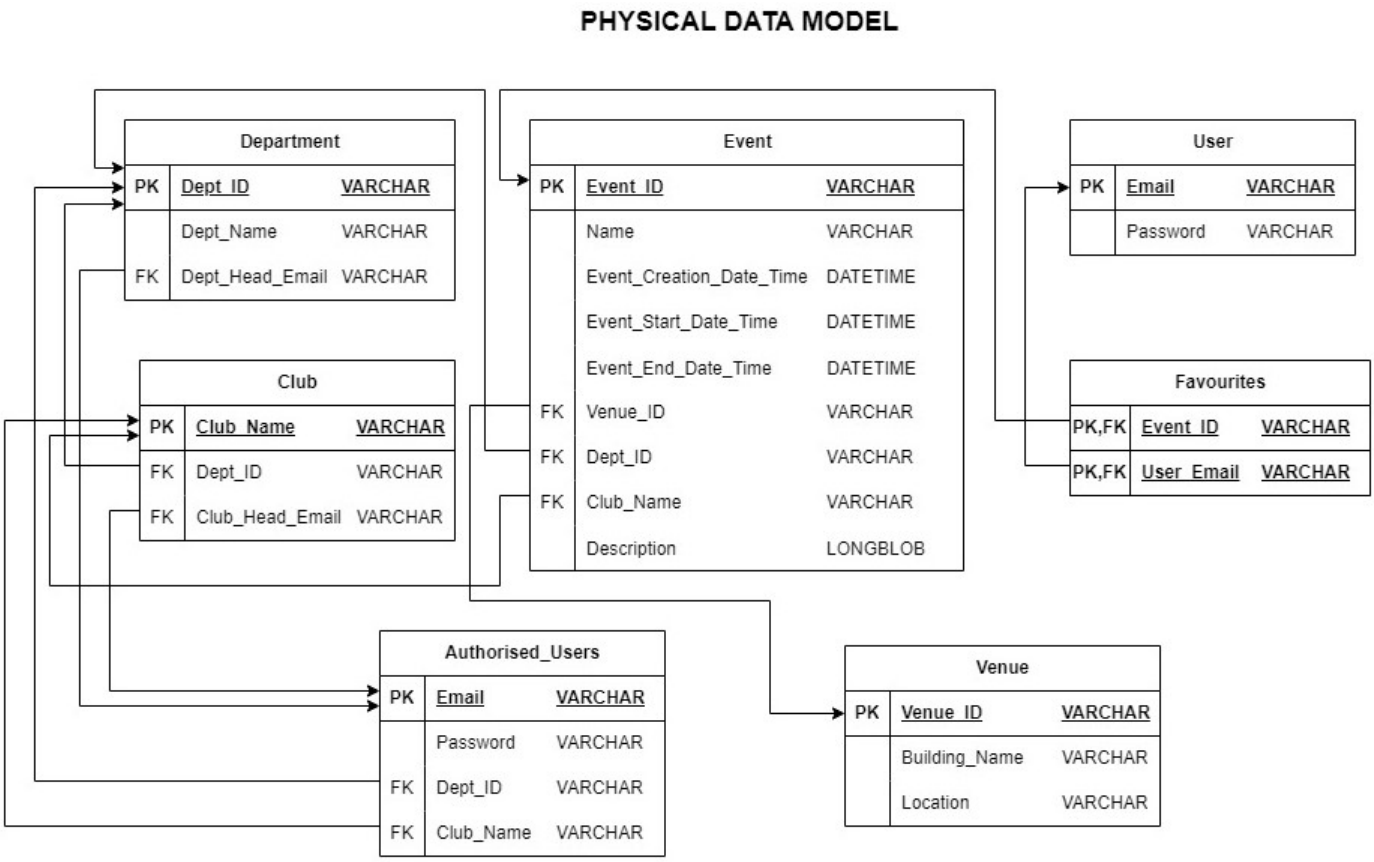

Figure 3: Physical Data Model

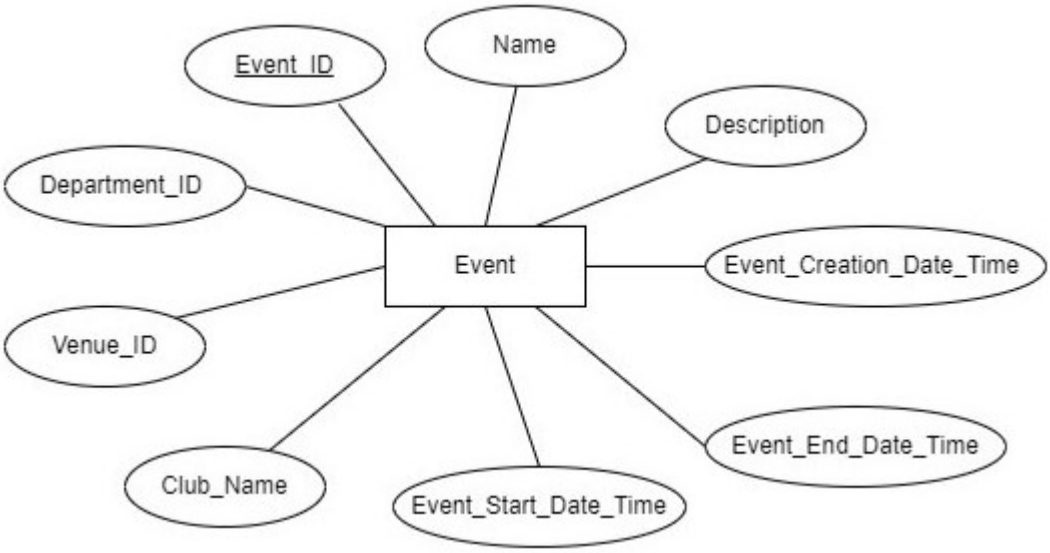

Figure 4: Event Entity

Figure 5: User Entity

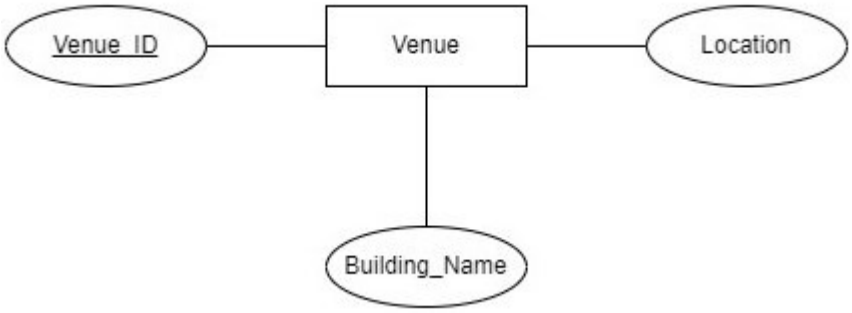

Figure 6: Venue Entity

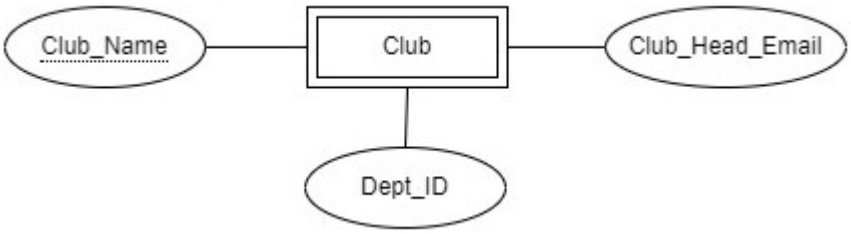

Figure 7: Club Entity

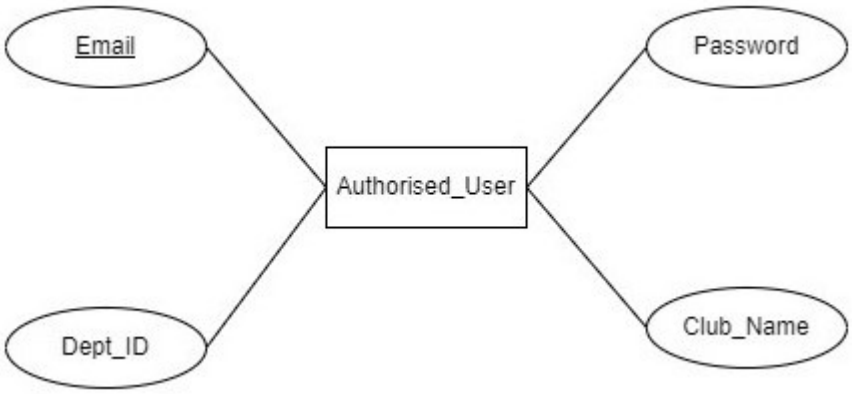

Figure 8: Authorised User Entity

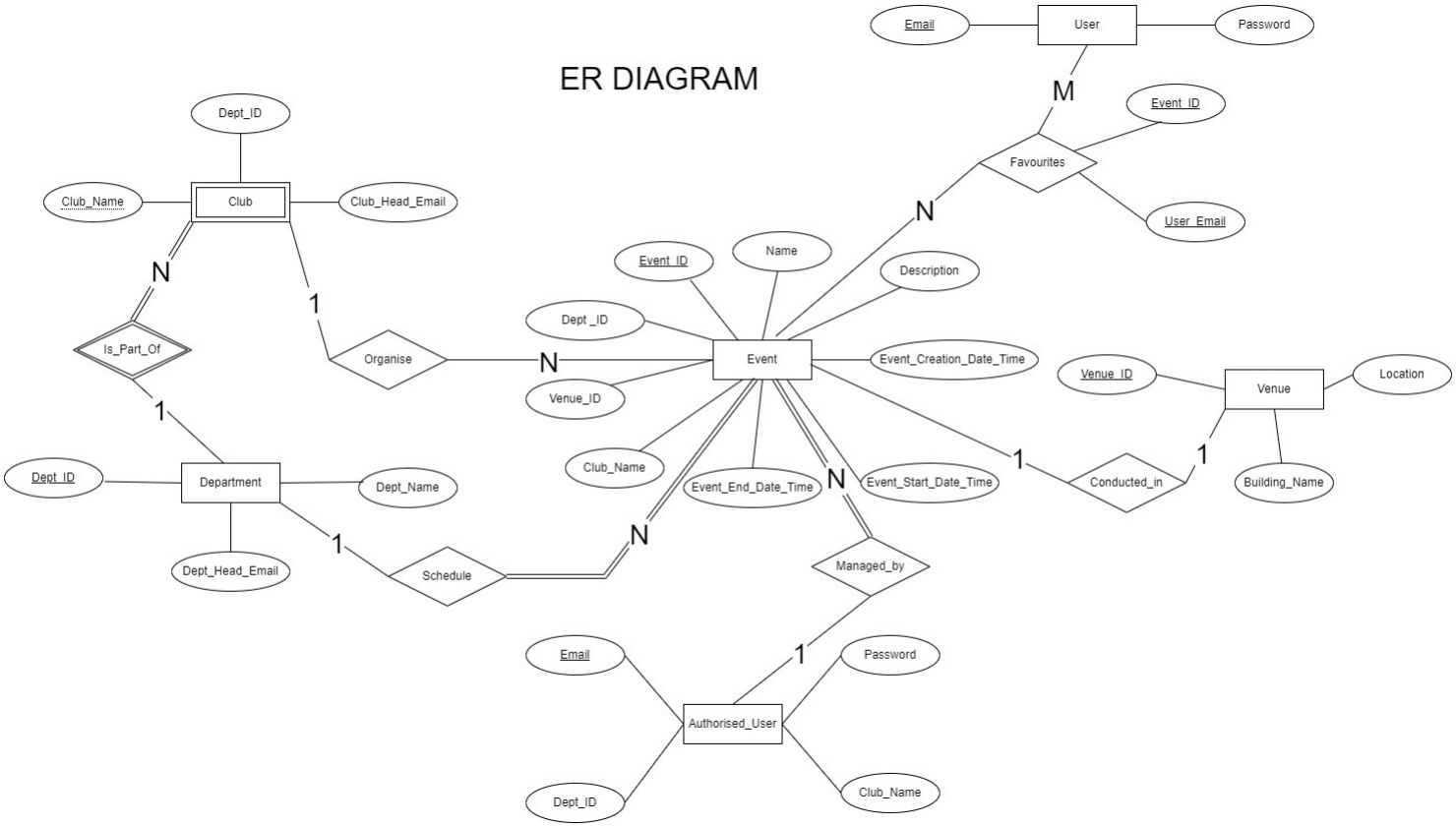

Figure 9: ER Diagram

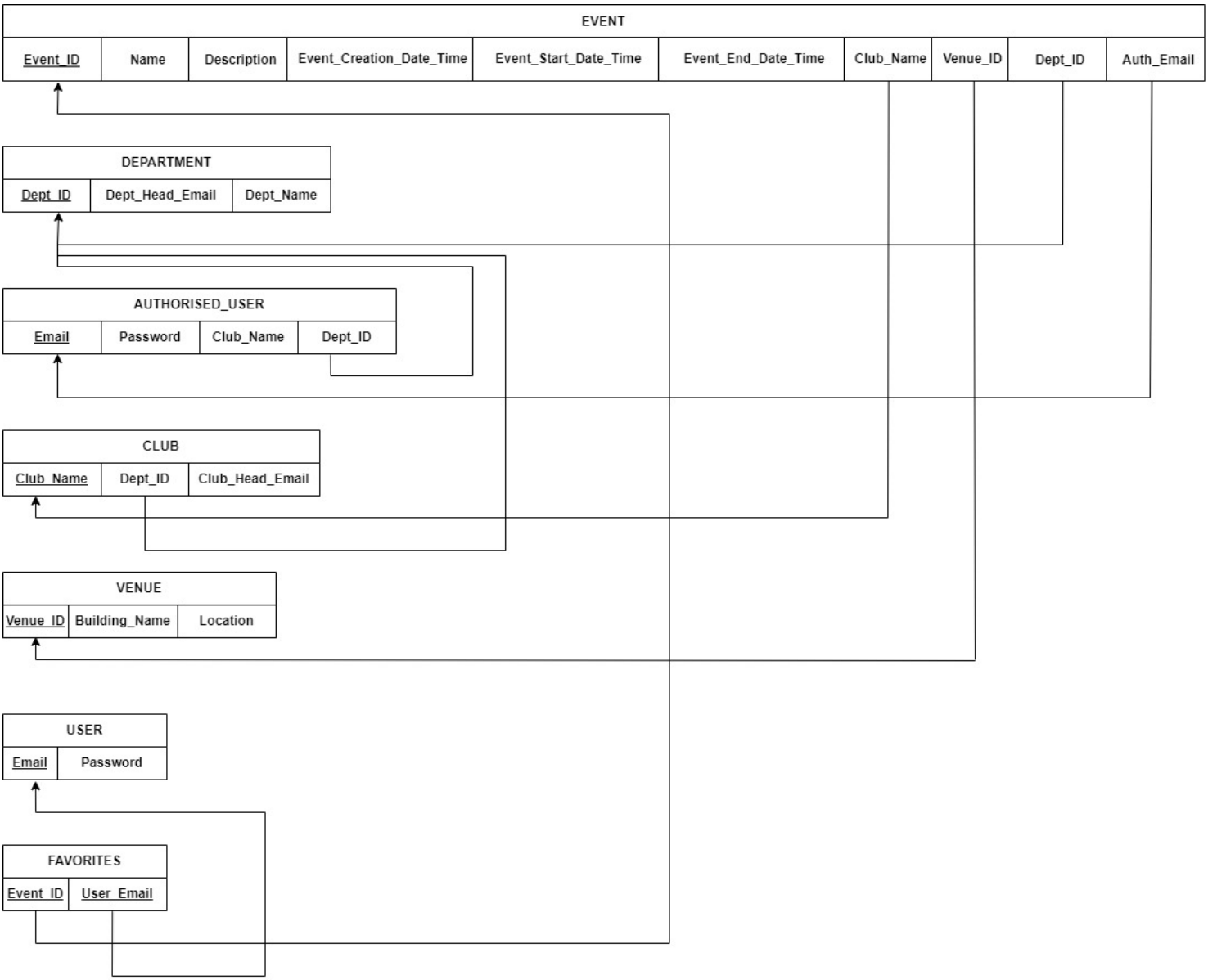

Figure 10: Schema