1 Requirement Gathering and Analysis

Requirement analysis in this project involves identifying the data that needs to be stored in the database and understanding how it will be accessed by different users. The system is being developed after discussing needs with one of the campus event organizers. Ensuring clear and complete requirements before beginning is essential for creating a system that meets the needs of all stakeholders.

1.1 Process of Requirement Analysis

1.1.1 Discussion with an Event Organizer and students

Discussions with an event organizer form the core of the analysis. Through these conversations, the aim is to identify the problems encountered in organizing campus events, managing attendees, and allocating resources. Below are some key points from the discussion.

- **Question 1:** What are some challenges in managing events on campus?

  Event Organizer: We face issues coordinating with multiple departments, handling large attendance, and ensuring resource availability. With a few dozen events every month, things can get disorganized without a systematic approach.

- **Question 2:** Are there any other specific problems?

  Event Organizer: Yes. Scheduling conflicts in booking rooms and handling last-minute resource requests often arise. With multiple events happening simultaneously, managing these details without a proper system becomes overwhelming.

- **Question 3:** What issues do participants face?

  Student: Sometimes, event locations change at the last minute, and we don't get notified on time. We also find it challenging to track which events they've signed up for, especially during busy weeks with several overlapping activities.

  Event Organizer: We also don't have a clear way to notify attendees if an event is canceled or rescheduled. It's frustrating when notifications aren't timely.

1.1.2 Understanding the Requirement

After discussing these points, it was concluded that the system must handle all event-related data automatically, minimizing errors and delays. The system should allow organizers and participants to manage and access information seamlessly and ensure reliable, timely communication of updates.

Basic information requirements identified are:

- **Event Information**: Event name, date, time, location, and organizer.
- **Participant Information**: Name, ID, email, and role in the event (e.g., attendee, speaker).
- **Resource Information**: Type, availability, and quantity of resources.
- **Location Information**: Room details, capacity, and availability for booking.

1.2 Identifying Stakeholders

Since the system is being developed to address event management challenges faced on campus, the key stakeholders are:

- **Event Organizers** (teachers, students, and staff involved in planning and managing events).
- **Participants** (students, faculty, and guests attending events).
- **Campus Administration** (managing location bookings and maintaining event records).

As the Question , I aim to create a solution that is user-friendly and effective for all stakeholders.

1.3 System Specification

After analyzing requirements and identifying stakeholders, a **web-based system** has been selected for development. The next step is to design a conceptual model, focusing on a high-level view to confirm that it aligns with the specified requirements.

The system will include features for event creation, participant registration, resource management, and real-time notifications, ensuring that all necessary information is accessible and manageable for everyone involved.

2 Conceptual Modeling

The next step after requirement analysis is conceptual modeling. Conceptual modeling is the process of developing an abstract model or graphical representation using real-world concepts or ideas. It is a high-level diagram that defines, describes, organizes, and presents data elements and their relationships with relatively few details.

For this project, the Entity Relationship Model (ERM) was chosen as it is widely used and was emphasized in our course. ERM provides a clear and organized way to represent data structures, which is essential for the campus event management system.

2.1 Entity-Relationship Model

An Entity-Relationship Diagram (ERD) shows the relationships of entities stored in a database. ER diagrams explain the logical structure of a database and illustrate how entities interact. In ER diagrams, three basic concepts are used:

1. **Entity**: In database management systems, an entity is a data object that represents a real-world object or concept, such as an event or a participant. Entities are often represented as tables, with each row representing a specific instance. Entities are shown in rectangular shapes in the ER model.

2. **Attribute**: An attribute represents a characteristic or property of an entity. For example, attributes for an event entity might include *event name*, *date*, and *location*. Attributes are represented by oval shapes in the ER model.

3. **Relationship**: Relationships are connections between entities that define how data in one entity relates to data in another. In the ER diagram, relationships between entities are represented by diamonds and are labeled to describe the interaction. Common types of relationships include:

   (a) One-to-One

   (b) One-to-Many

   (c) Many-to-One

   (d) Many-to-Many

2.2 Detailing the Entity-Relationship Model

From the analysis, the following entities and relationships have been identified for the Campus Event Management System:

2.2.1 Entity Types

1. **User**: Represents users of the system with attributes such as *user_id*, *username*, *email*, *password*, *user_type*, *created_at*, and *last_login*.

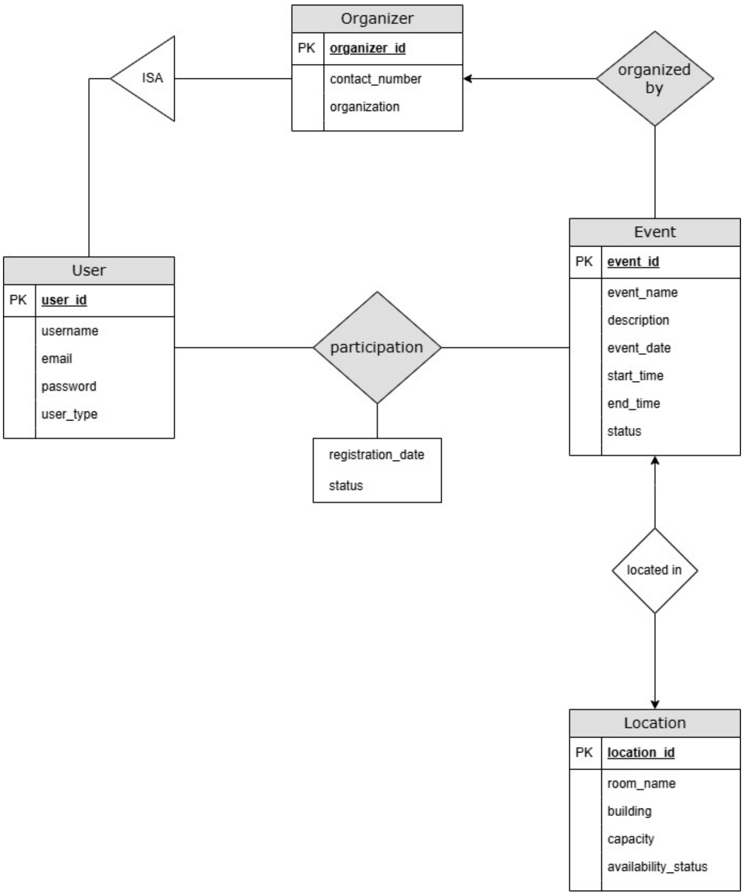
2. **Event**: Represents each campus event, including attributes like *event_id*, *event_name*, *description*, *event_date*, *start_time*, *end_time*, *location_id*, *organizer_id*, *status*, and *max_attendees*.

3. **Organizer**: Represents individuals organizing events with attributes such as *organizer_id*, *name*, *contact_number*, *email*, *user_id*, and *organization*.
4. **Participant**: Represents people attending events, including attributes like *participant_id*, *event_id*, *user_id*, *registration_date*, and *status*.
5. **Location**: Represents locations where events are held, with attributes like *location_id*, *room_name*, *building*, *capacity*, and *availability_status*.

2.2.2 Relationship Types

The relationships between the entities identified in the ER model are as follows:

- **Organizes**: Connects *Organizer* and *Event* entities, where each organizer can manage multiple events.
- **Registers**: Connects *Participant* and *Event* entities, where a participant can register for multiple events, and each event can have multiple participants.
- **Requires**: Connects *Event* and *Resource* entities, where an event may require multiple resources, and each resource can be allocated to multiple events.
- **Located At**: Connects *Event* and *Location* entities, where each event is held in a specified location.



In the figure the ER diagram displays these entity types and their relationships. For instance, the relationship between **Event** and **Participant** indicates that one participant can register for multiple events. The **Organizer** manages one or more events, and resources are allocated to events as needed.

Entities are represented in rectangular shapes, relationships in diamond shapes, and attributes in oval shapes.