

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**BELAGAVI, KARNATAKA - 590018**



Design Phase Report of  
**“End to End Event Management System”**

*Submitted in partial fulfillment of the requirements for the award of the degree*

**BACHELOR OF ENGINEERING**  
**in**  
**COMPUTER SCIENCE AND ENGINEERING**  
By

<b>Nireekshith</b>	<b>4SO18CS081</b>
<b>Novin Misquith</b>	<b>4SO18CS086</b>
<b>Pranith Rao</b>	<b>4SO18CS088</b>
<b>Vaibhav K Suvarna</b>	<b>4SO18CS120</b>

Under the guidance of  
**Ms. Gayana M N**  
Asst. Prof, CSE Dept  
Department of Computer Science and Engineering



**ST JOSEPH ENGINEERING COLLEGE**  
**MANGALURU 575028**  
**2021-22**

# CONTENTS

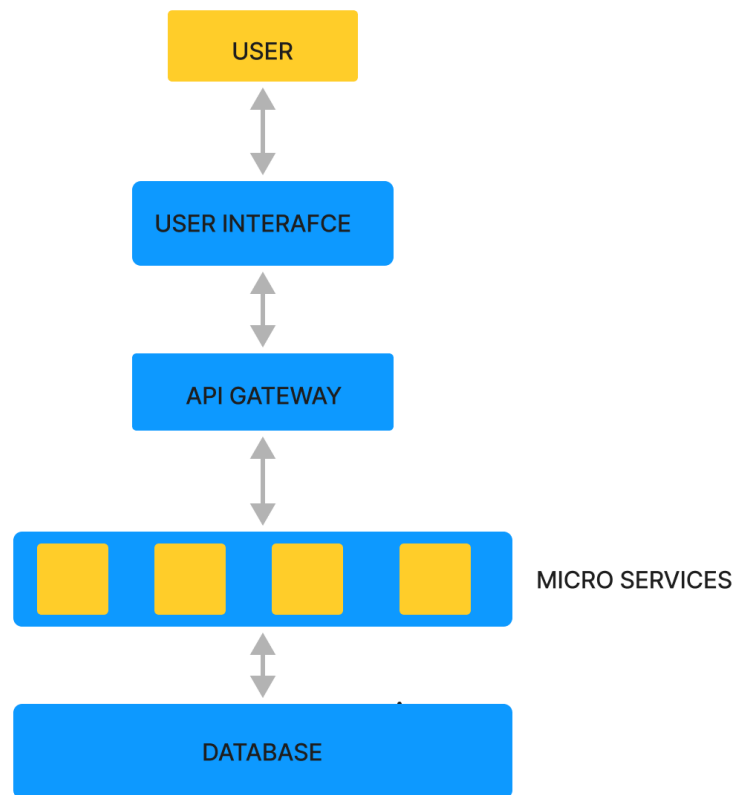
1. Abstract Design.....	04
1.1 Architectural diagram.....	04
1.2 Use case diagram.....	05
2. Functional Design.....	06
2.1 Data flow diagram.....	06
2.2 Sequence diagram.....	07
3. Control flow Design.....	08
3.1 Complete system flow diagram.....	08
4. Access layer Design .....	10
4.1 Database Schema.....	10
4.2 ER Model.....	10

## **LIST OF FIGURES**

1.1.1 Architectural diagram.....	04
1.1.2 Use Case Diagram.....	05
2.1.1 Data Flow Diagram.....	06
2.2.1 Organizer Sequence Diagram.....	07
2.2.2 Judge Sequence Diagram.....	07
2.2.3 Student Sequence Diagram.....	08
3.1.1 System Flow Diagram.....	09
4.1.1 Database Schema.....	10
4.2.1 ER Model.....	11

## 1. Abstract Design

### 1.1 Architectural diagram



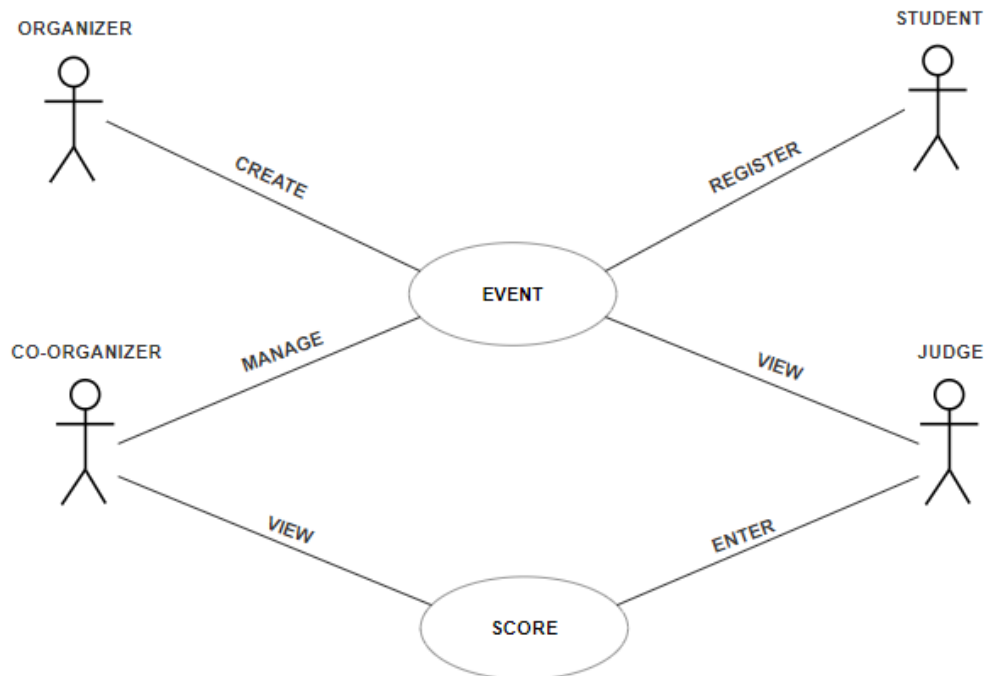
**Fig 1.1.1 Architectural diagram**

End to End Event management is the management of events which consists of the creation, development and updating of small and/or large-scale personal or corporate events. Events are organized by almost all universities, but all work from registration to certification is done by hand and is not digitalized. Our project demonstrates on managing an event using a web-based interface. The main objective in developing an event management interface is providing the institution a single application which will help them to organize and manage all the events.

Organizer adds the co-organizers who add the events which are visible to the participants and they registered for the events of their interest and judges assigned to each events evaluate the participants' performance and give scores according to it which will help in issuing certificates to the participants. An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. The diagram explains about the system software in perception of overview of the system.

## 1.2 Use case diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. This project includes four main actors: organizer, co-organizer, judge and participant/student.



**Fig 1.1.2 Use Case Diagram**

Detailed Description of the Use cases:

- Use Case:** Create Event  
**Summary:** The interface allows the organizer to create multiple events.  
**Actor:** Organizer  
**Description:** The organizer can create events involved in the program that can be participated by the participants and managed by the co-organizers.
- Use Case:** Register Event  
**Summary:** The interface allows the participants to register and participate in the event.  
**Actor:** Student  
**Description:** The participants can register for the events of their interest. The number of events in which they can participate can be limited based on the restrictions.
- Use Case:** Manage event and view scores  
**Summary:** The interface allows the co-organizer to manage event and view the scores submitted by the judges.

**Actor:** Co-Organizer

**Description:** The co-organizer manages the event he is assigned to and can also view scores that are submitted by the judges on basis of the performance of the participants in that particular event

- **Use Case:** View event and submit scores.

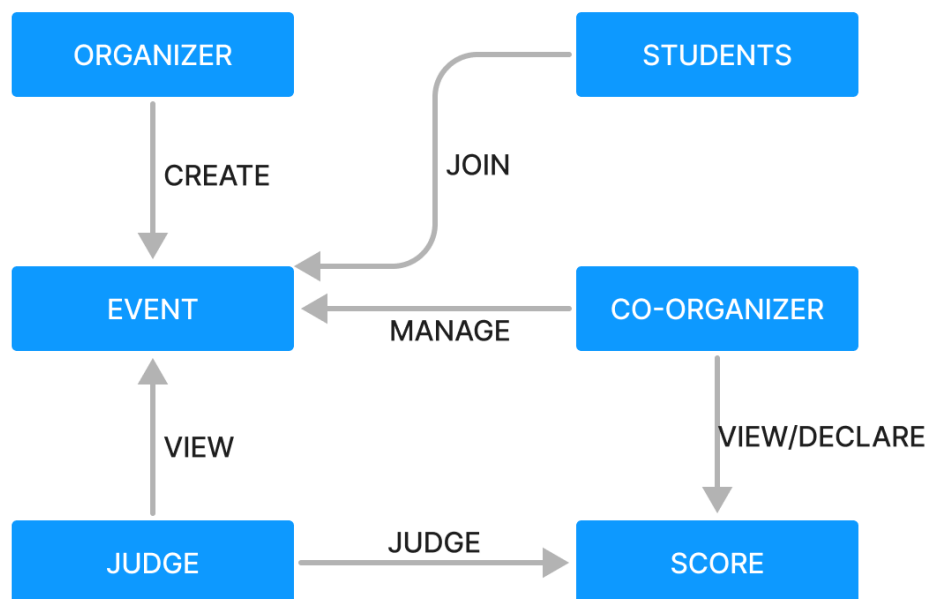
**Summary:** The interface allows the judges to view the events and submit scores on basis of the performance of the participants.

**Actor:** Judge

**Description:** The judge can view the details of the event he is assigned to judge and also can submit the scores after judging the performance of the participants.

## 2. Functional Design

### 2.1 Data flow diagram

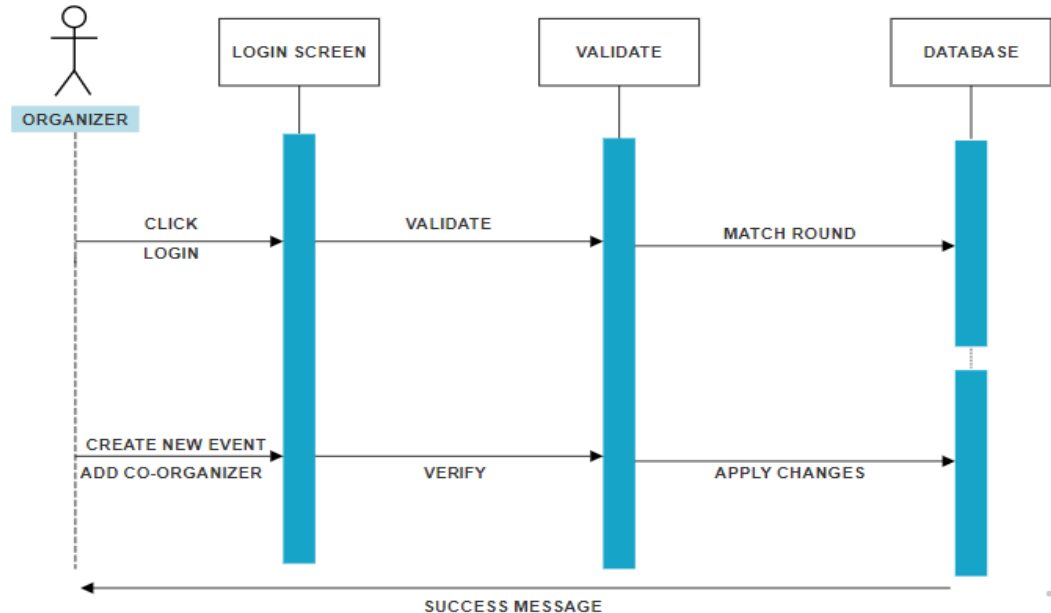


**Fig 2.1.1 Data Flow Diagram**

The dataflow diagram of this project consists of all the various aspects a normal flow diagram requires. This dataflow diagram shows how organizer adds events and assigns co-organizers to manage them, how students join the events of their interest and how judges judge the performance of each participant in various events and assign scores as per their performance that helps in issuing certificates to the participants.

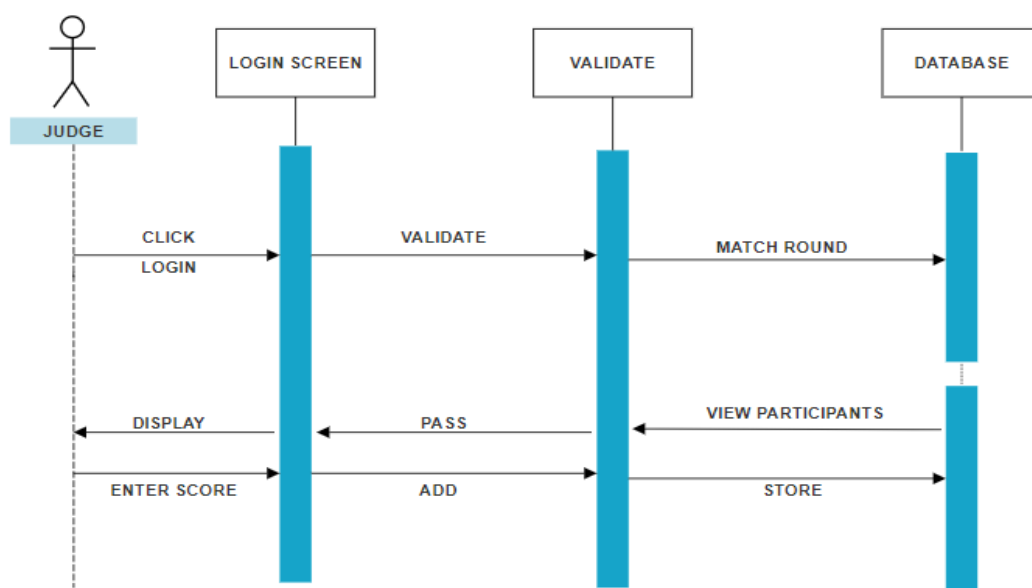
## 2.2 Sequence diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates how a group of objects interact and operate with each other sequentially. Here all the sequence diagrams show how each actor interacts with the interface and explains the outcome of each action.



**Fig 2.2.1 Organizer Sequence Diagram**

The organizer first clicks the login button that redirects him to the login page where the organizer enters his credentials. Once the credentials are validated, he gets access to the dashboard where he has the option to create and view events and add co-organizers to manage the events. Once all the details are verified and submitted a success message will be issued.



**Fig 2.2.2 Judge Sequence Diagram**

The judge first clicks the login button that redirects him to the login page where he enters his credentials. Once the credentials are validated, he gets access to the dashboard where he has the option to view all the details of the event and once the event begins, he has the option to enter the scores of the participants on basis of their performance that are stored in the database and used to issue certificates to the participants accordingly.

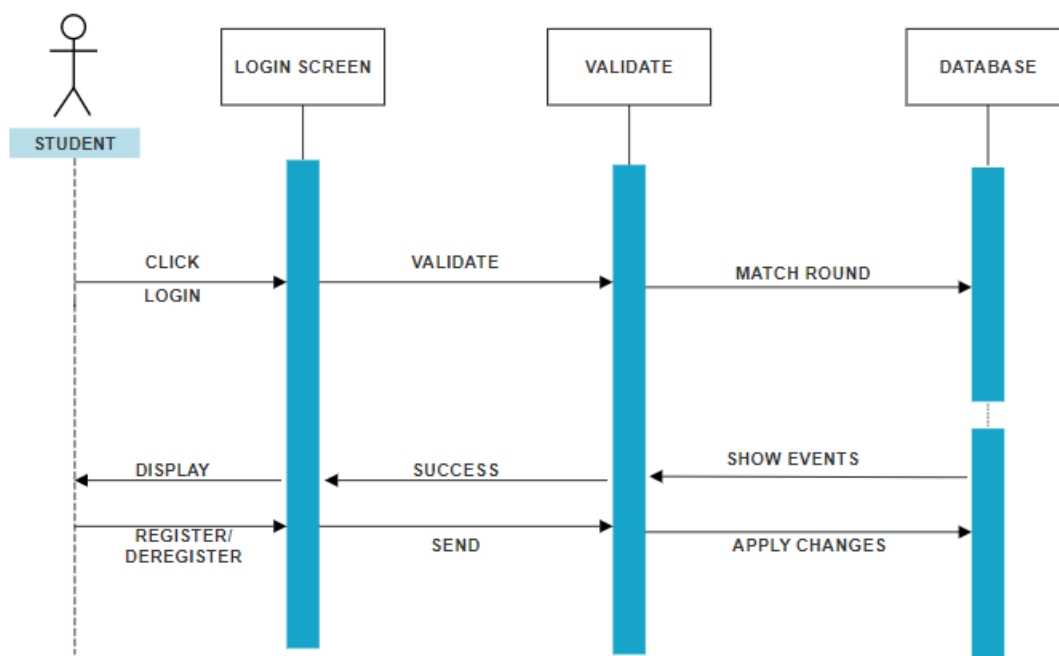


Fig 2.2.3 Student Sequence Diagram

The student first clicks the login button that redirects him to the login page where he/she enters his/her credentials. Once the credentials are validated, he/she gets access to the dashboard where they have the option to view all the events and register for the events of their interest. Once it's done the registration of each participant is stored in the database.

### 3. Control flow Design

#### 3.1 Complete system flow diagram

The system flow diagram shows the activities carried out by each actor using the interface. Organizer adds all the events and assigns one or more Co-Organizers to each event to handle them. Co-Organizer will be managing a particular event and will have access to only that event. He will be responsible for the event he is conducting and will be able to publish results of the same. Students will basically be able to register for an event and view the details of that event. He/she will also be even given an option to cancel the event if necessary. Judge will judge the particular event or game and allot the scores that will be used to declare the winners.



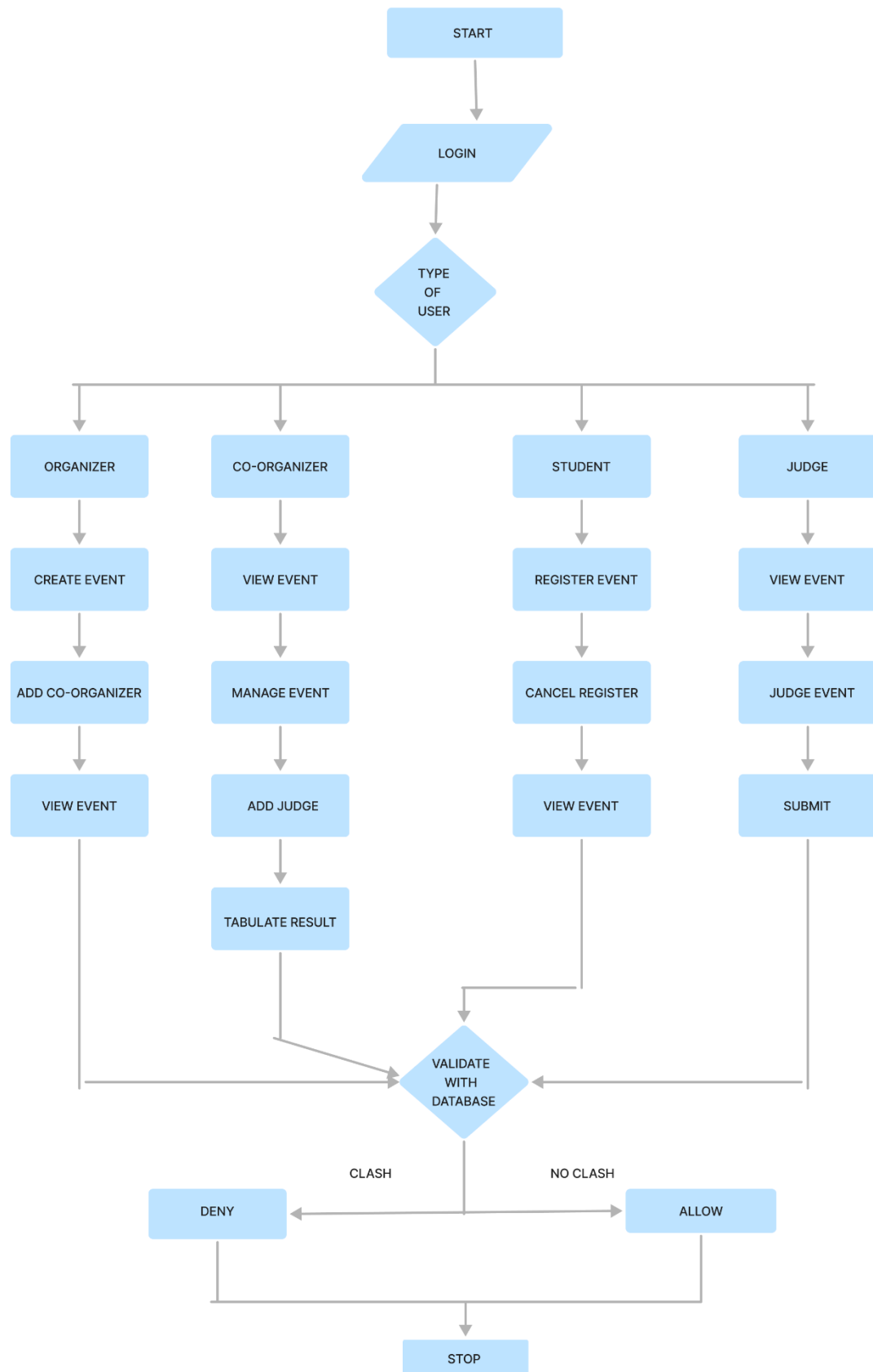


Fig 3.1.1 System Flow Diagram

## 4. Access layer Design

### 4.1 Database Schema

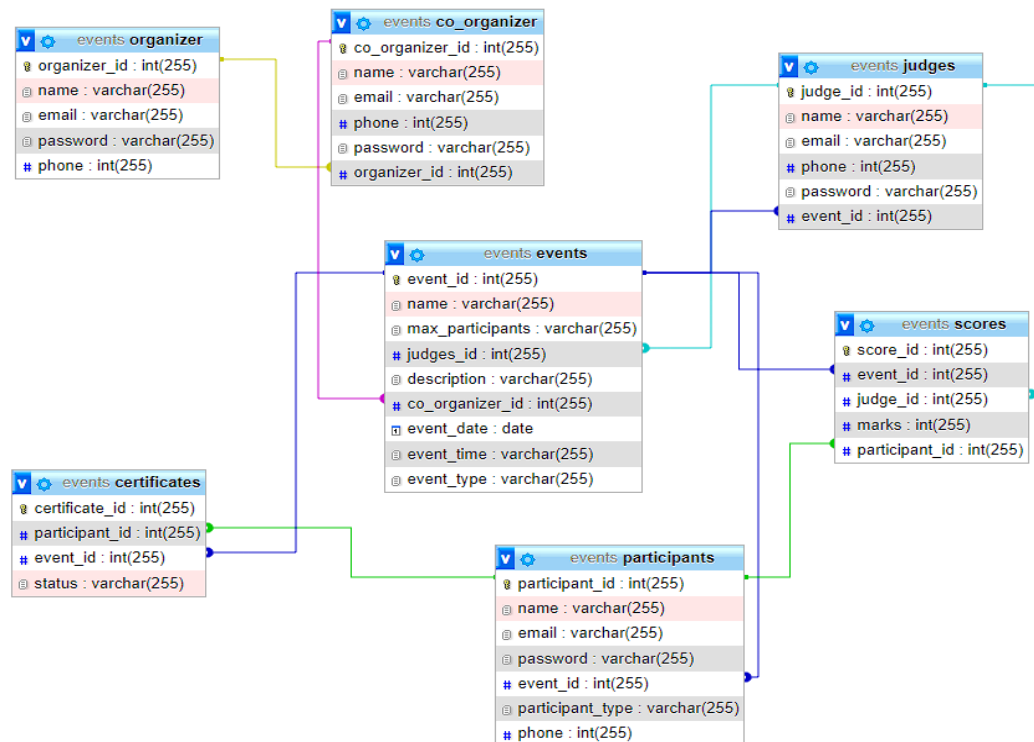


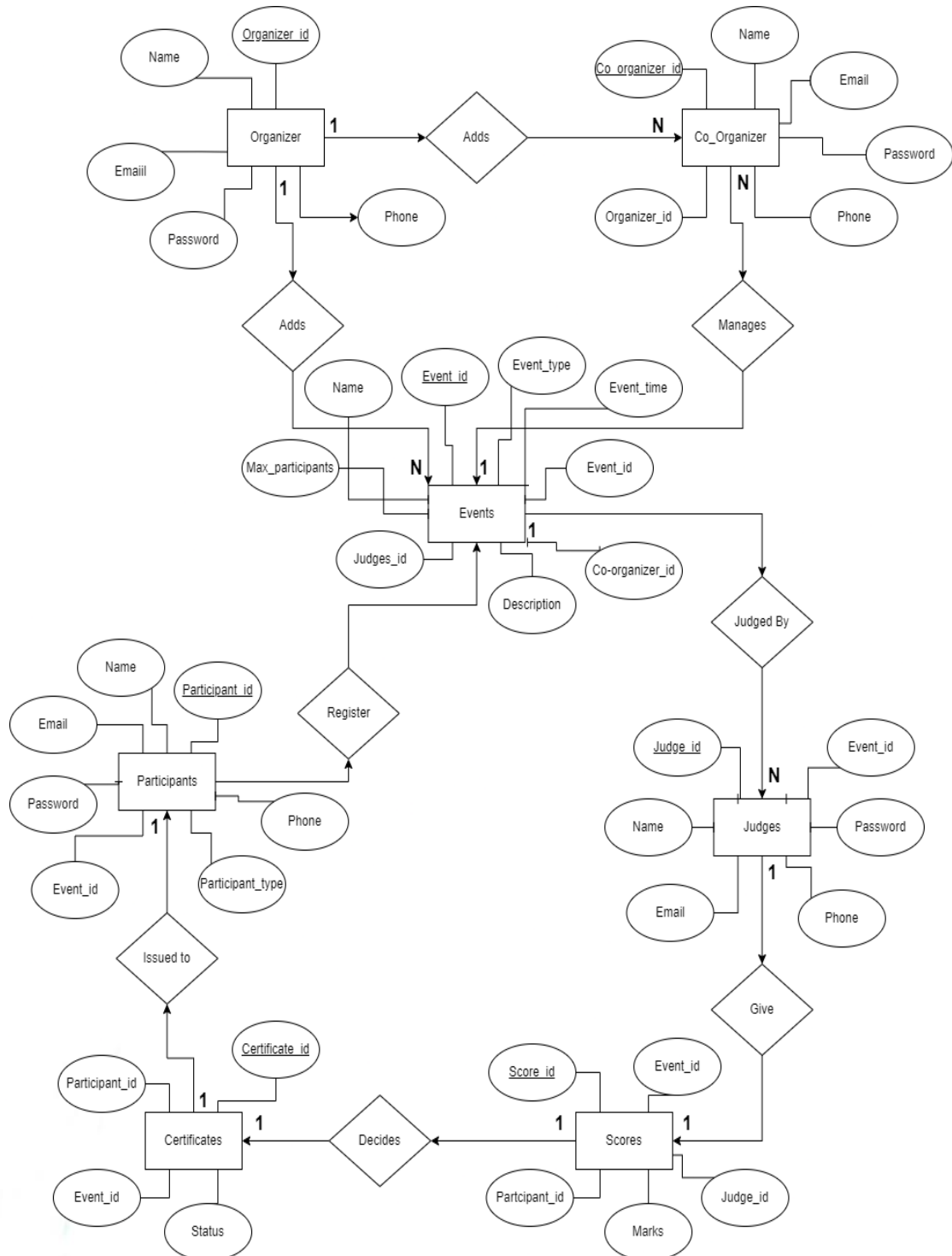
Fig 4.1.1 Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. The diagram shows how data is stored in a relational database structured with seven different tables namely Organizer, Co-Organizer, Judges, Events, Participants, Scores and Certificates and all the fields associated with each table. It also depicts the relation between tables and constraints imposed on each field.

### 4.2 ER Model

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. The ER diagram below shows how organizers add co-organizers and events that are managed by co-organizers and participated by students and judged by judges and scores allotted by judges used to issue to certificates to participants. The Cardinality ratio of organizer and co-organizer is 1:N as one organizer can add one or more co-organizers. The Cardinality ratio of organizer and events is 1:N as one organizer can add multiple events. The Cardinality ratio of co-organizer and events is N:1 as multiple co-organizers can manage only one event. The Cardinality ratio of events and judges is 1:N as one event is judged by multiple judges. The

Cardinality ratio of judges and scores is 1:1 as each judge allots score to each participant. The Cardinality ratio of score and certificates is 1:1 as score of each participant is used to issue to certificate to him/her. The Cardinality ratio of certificates and participants is 1:1 as certificate of each event is issued to that particular participant only.



**Fig 4.2.1 ER Model**