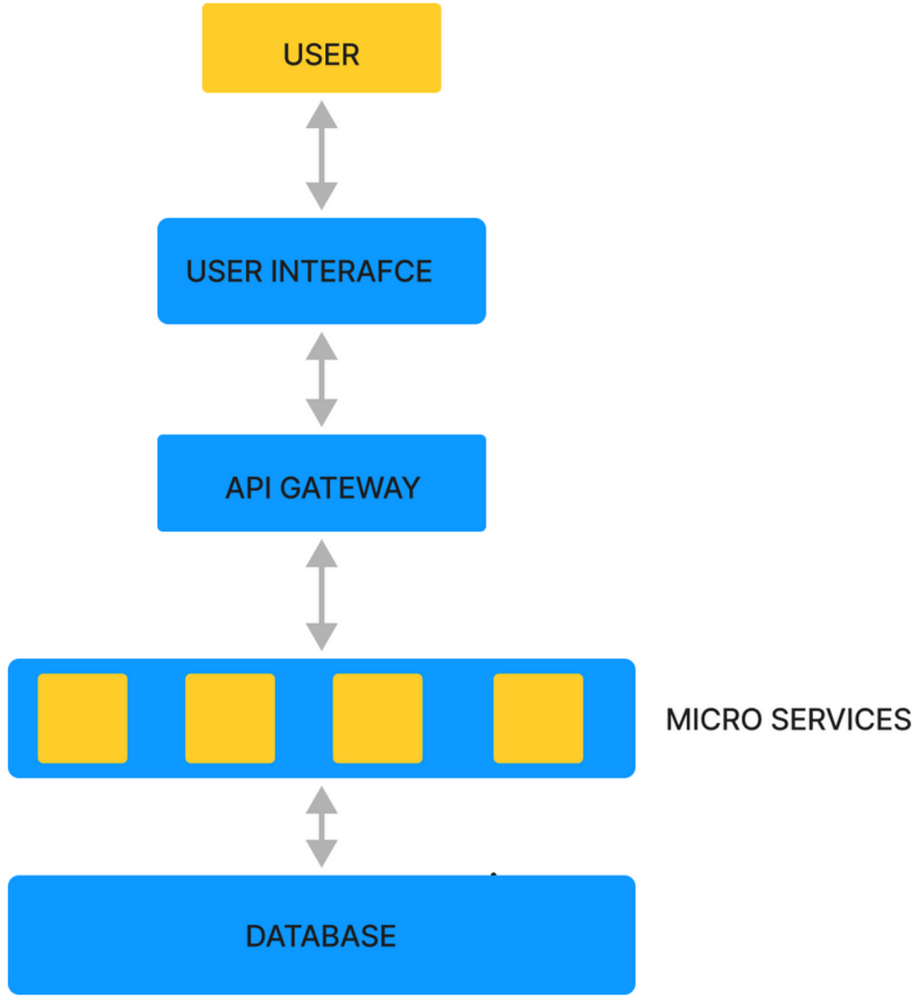
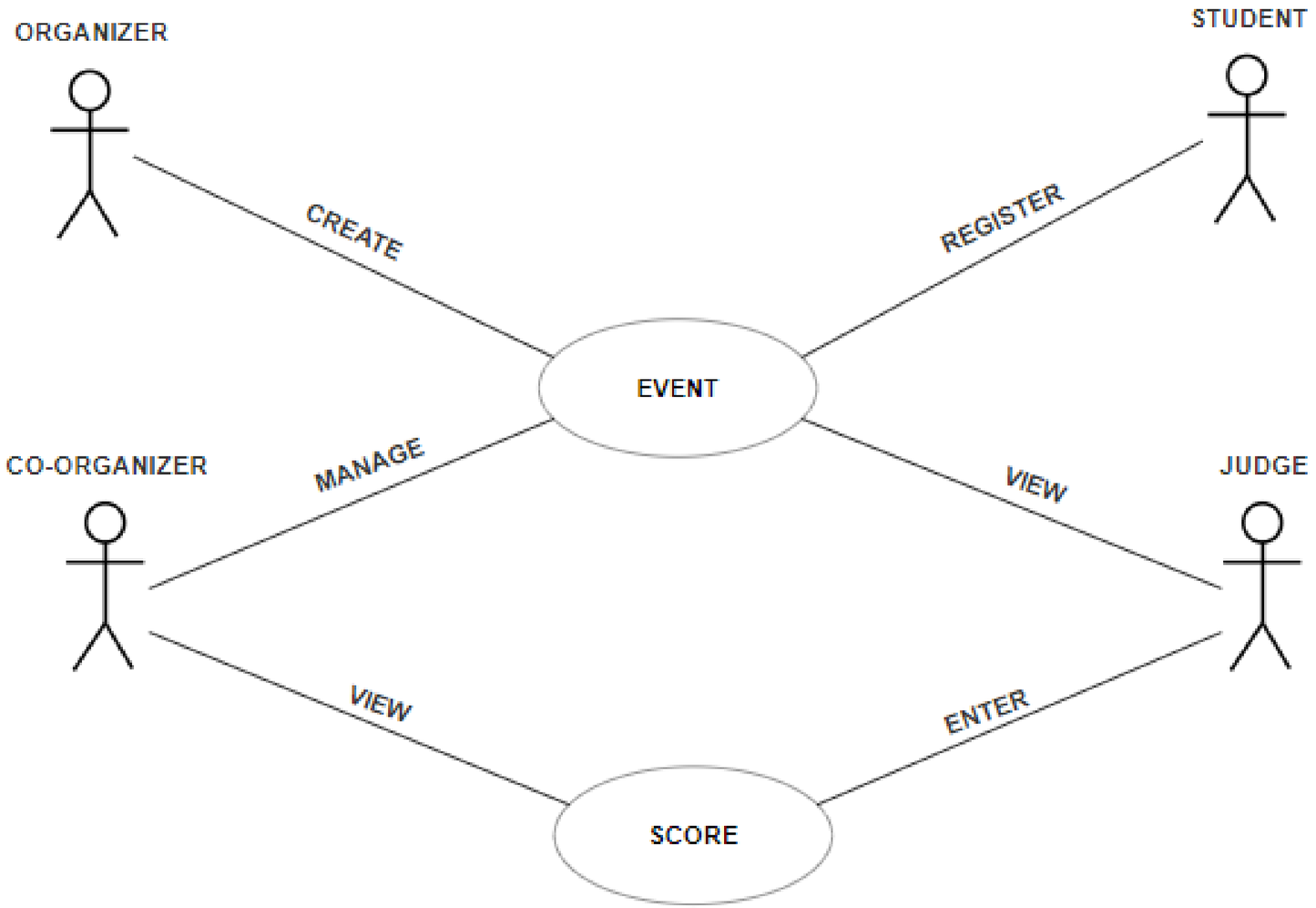
END TO END
EVENT MANAGEMENT

Architectural diagram

- End to End Event management deals with 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 the events 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.



Use case diagram



Use Case

Organizer

The organizer can create events involved in the program that can be participated by the participants and managed by the co-organizers.

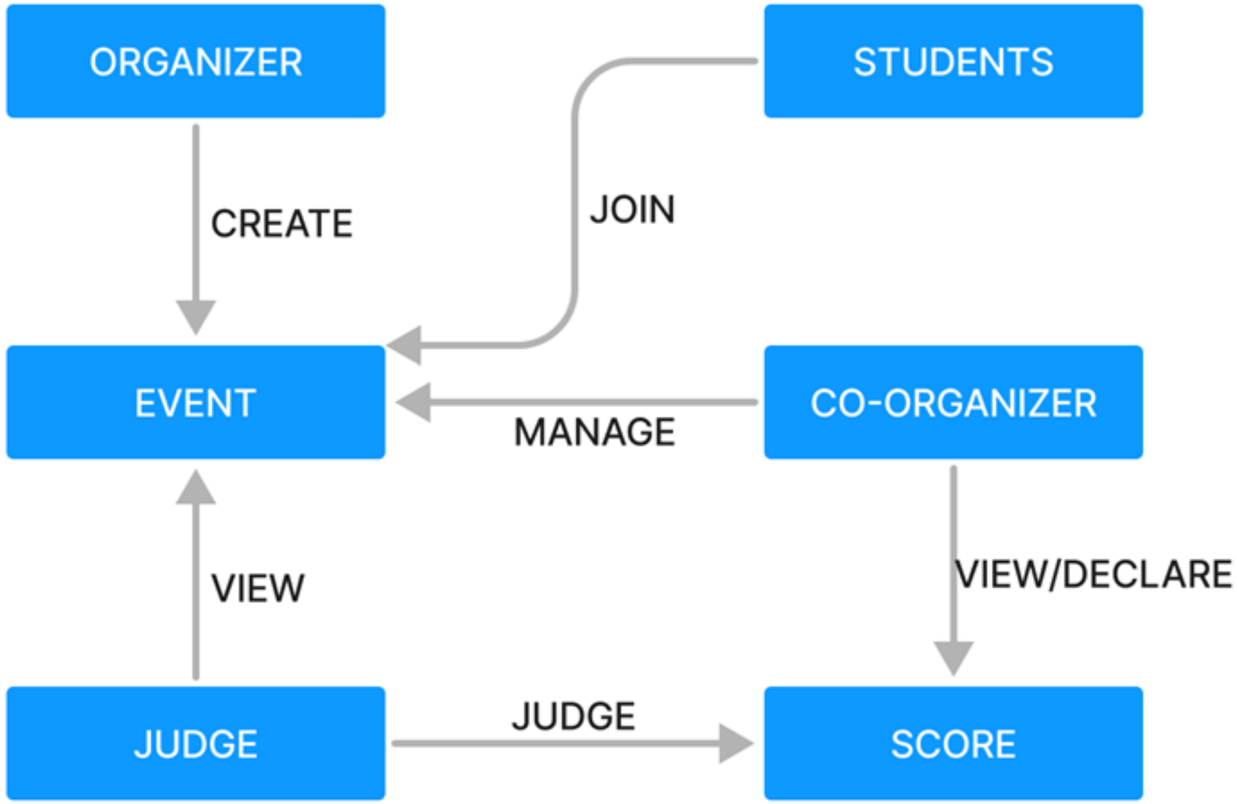
Student

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.

Judge

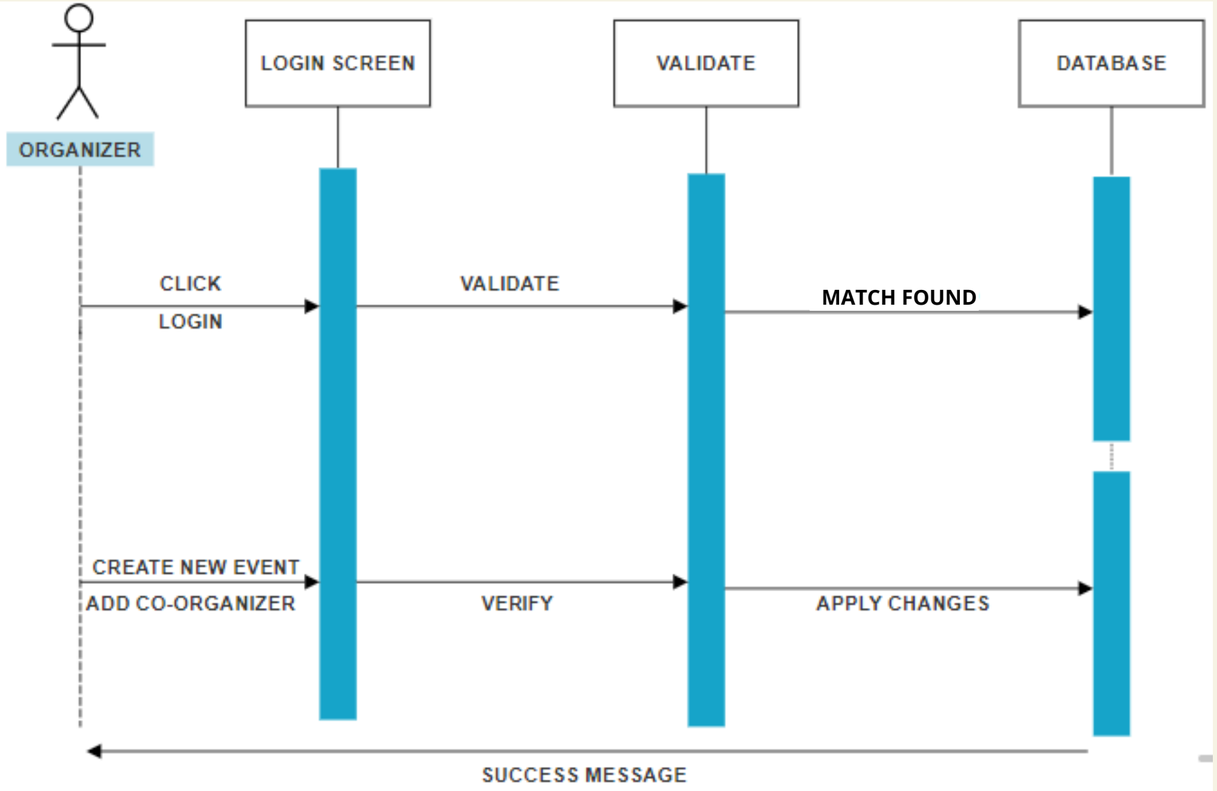
The judge can view the details of the event he is assigned to judge and also can submit the scores after judging the performance of the participants.

Data flow diagram



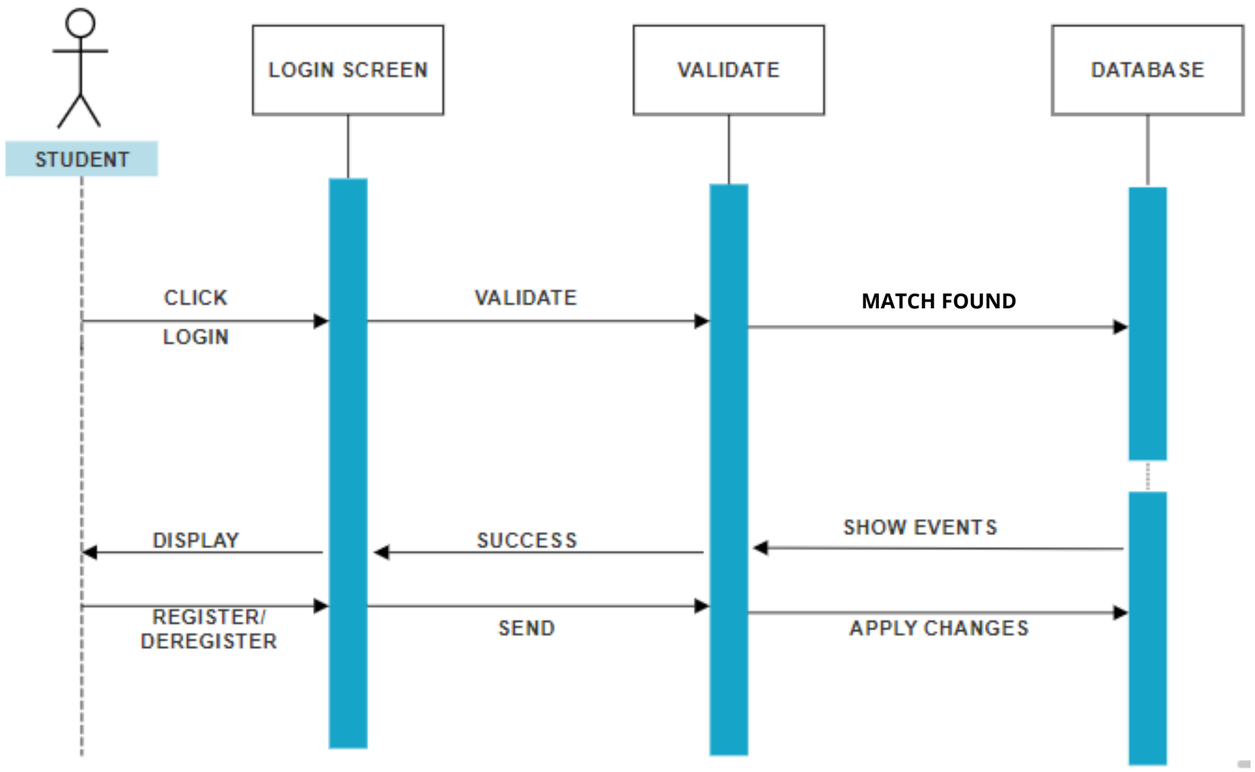
- A data flow diagram (DFD) maps out the flow of information for any process or system.
- This data flow diagram shows how the organizer adds events and assigns coorganizers to manage them, students join the events of their interest, and how judges judge the performance of each participant in various events and assign scores as per their performance that help in issuing certificates to the participants.

Sequence diagram – Organizer



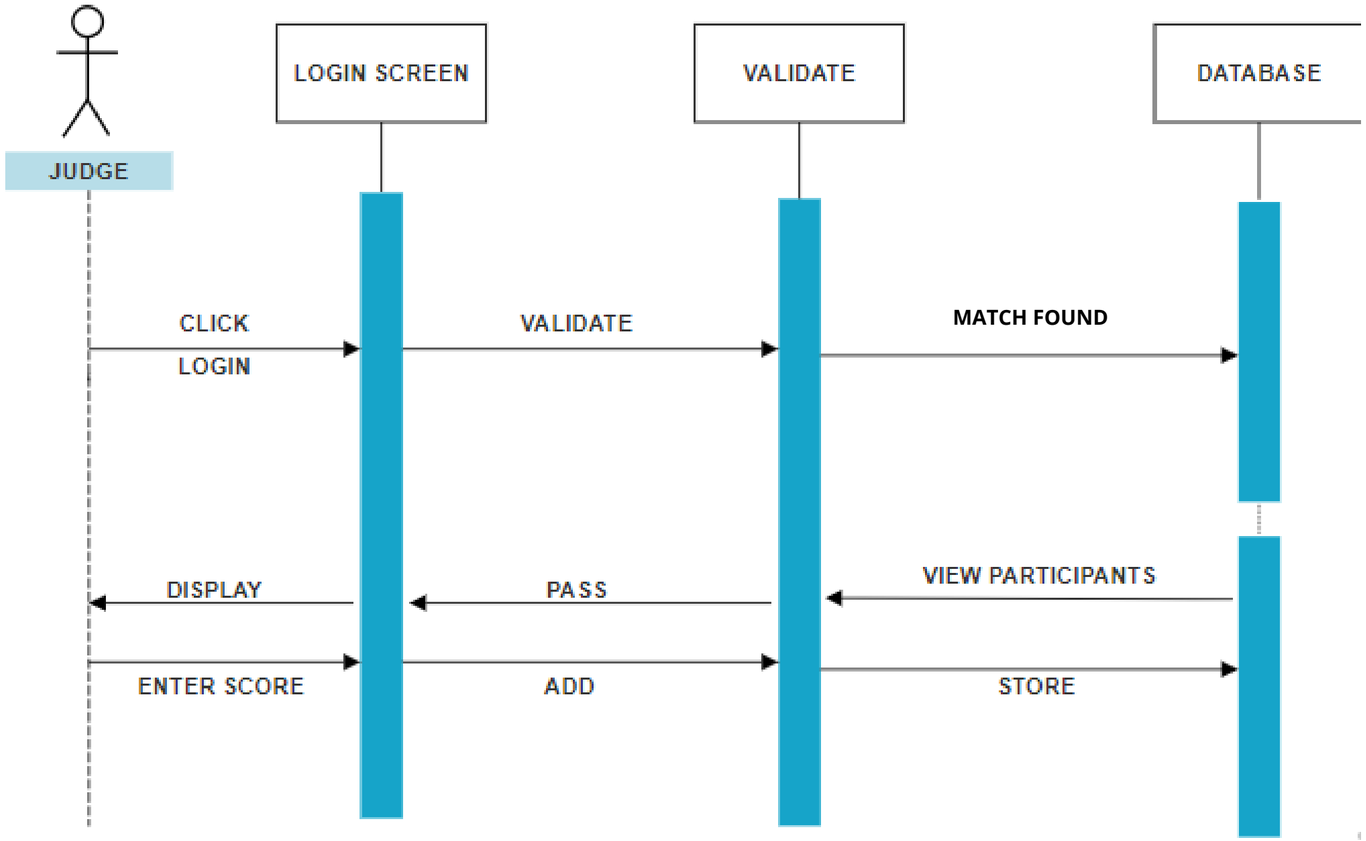
- The organizer first clicks the login button that redirects him to the login page where the organizer enters his credentials.
- Once the credentials are validated, he gets access to the dashboard where he has the option to create and view events and add co-organizers to manage the events.
- After all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

Sequence diagram – Student

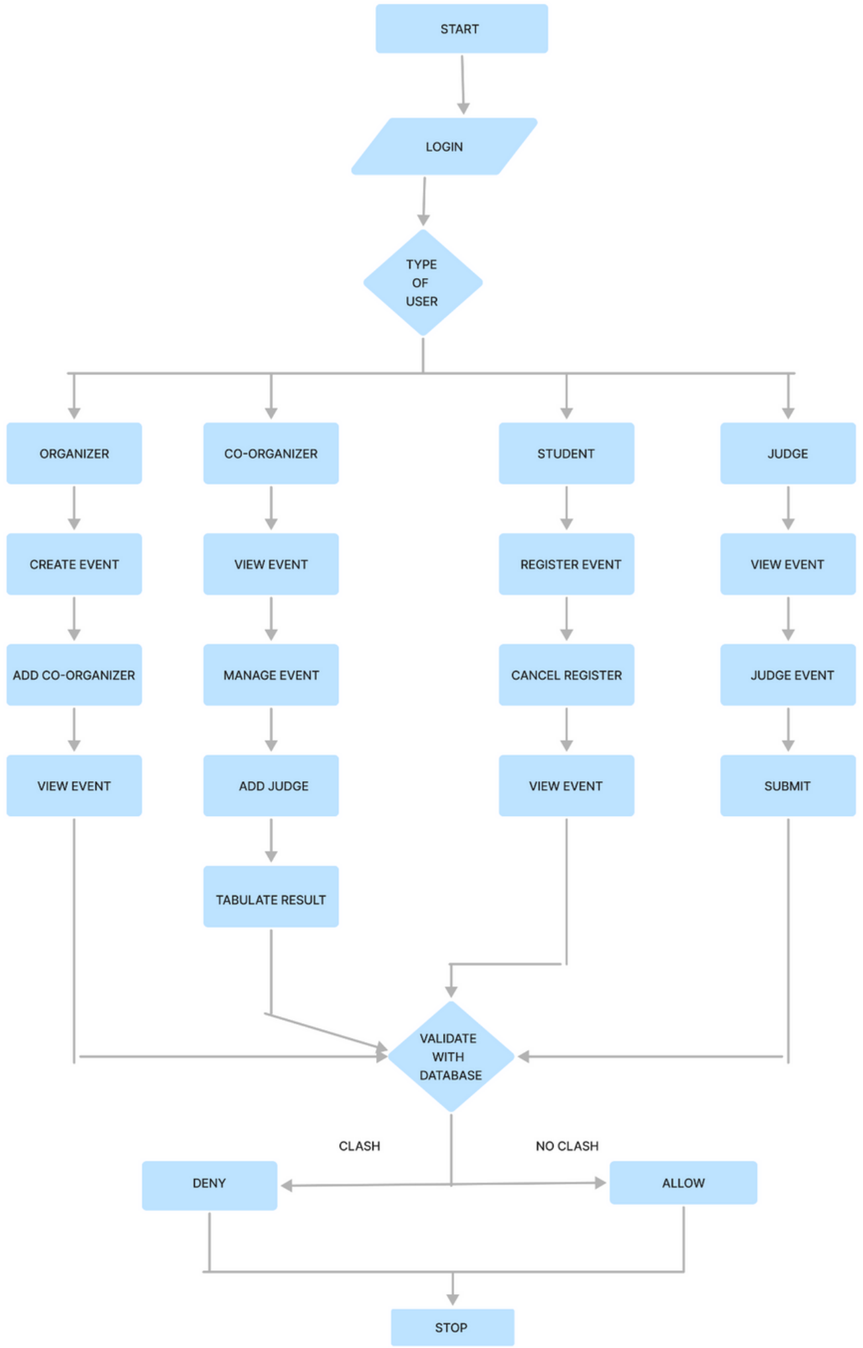


- The student first clicks the login button that redirects them to the login page where they can enter credentials.
- Once the credentials are validated, they get access to the dashboard where they have the option to view all the events and register for the events of their interest.
- Once it's done the registration of each participant is stored in the database.

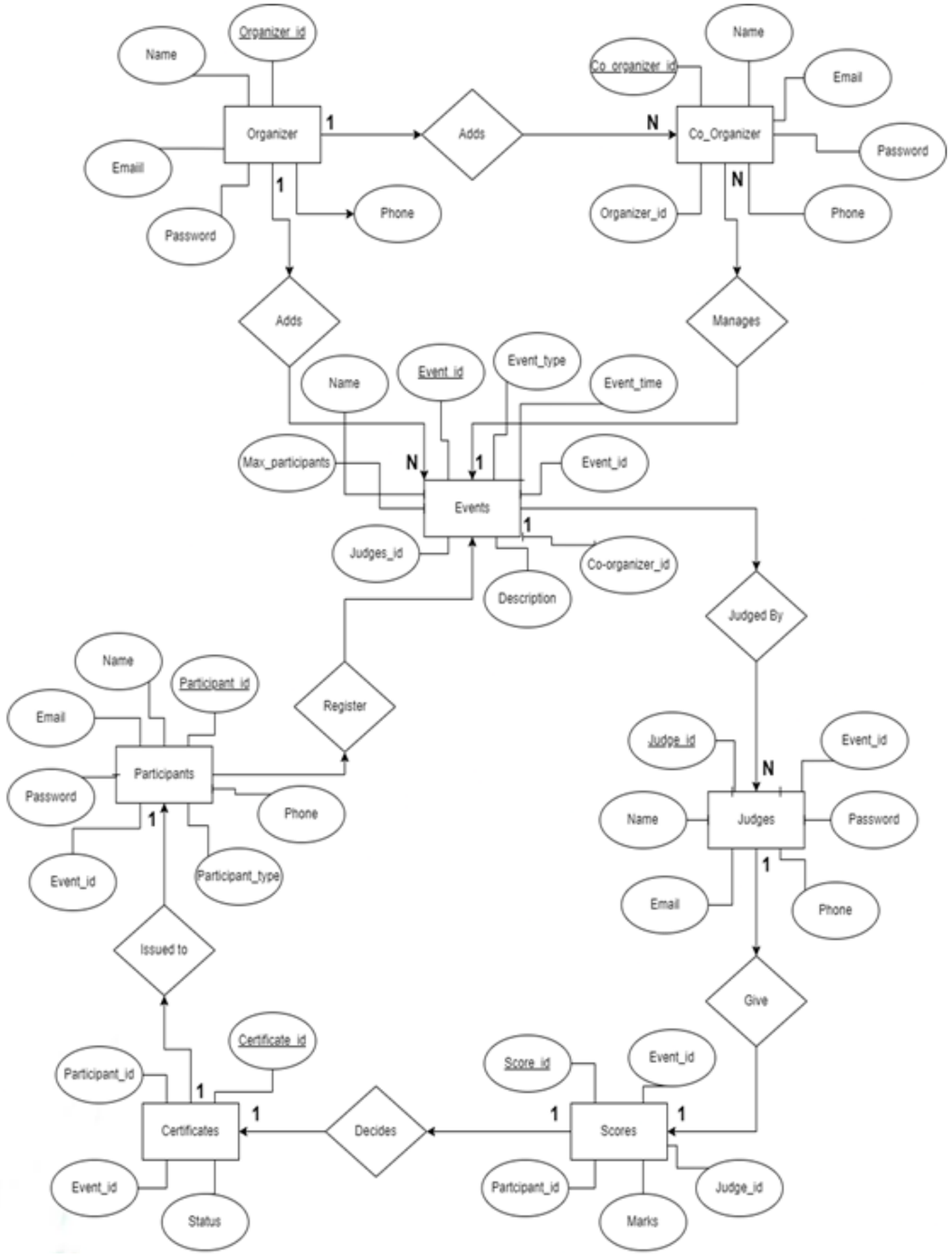
Sequence diagram – Judge

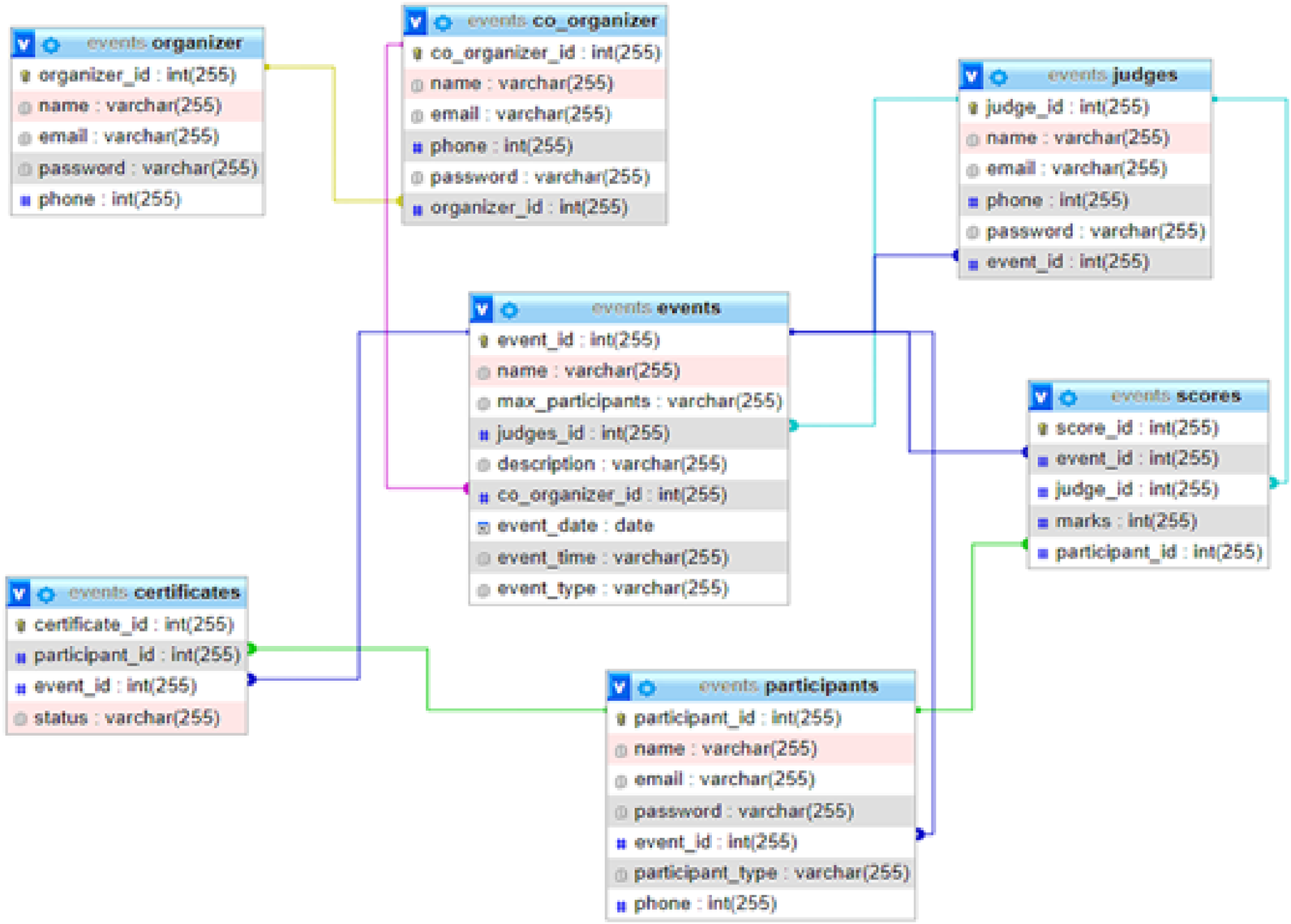


- The judge first clicks the login button that redirects them to the login page where they enter their credentials.
- Once the credentials are validated, they get access to the dashboard where they have the option to view all the details of the events and once the event begins, they have the option to enter the scores of the participants which will be stored in the database and used to issue certificates to the participants accordingly.


START
LOGIN
TYPE OF USER
ORGANIZER
CO-ORGANIZER
STUDENT
JUDGE
CREATE EVENT
VIEW EVENT
REGISTER EVENT
VIEW EVENT
ADD CO-ORGANIZER
MANAGE EVENT
CANCEL REGISTER
JUDGE EVENT
VIEW EVENT
ADD JUDGE
VIEW EVENT
SUBMIT
TABULATE RESULT
VALIDATE WITH DATABASE
CLASH
NO CLASH
DENY
ALLOW
STOP

- 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.


Organizer_id
Name
Organizer
1
Adds
N
Co_organizer_id
Name
Email
Co_Organizer
Password
Emaiil
1
Phone
Password
Organizer_id
N
Phone
Adds
Manages
Name
Event_id
Event_type
Event_time
Max_participants
N
1
Event_id
Events
1
Judges_id
Co-organizer_id
Description
Judged By
Name
Participant_id
Email
Register
Judge_id
Event_id
N
Participants
Password
Phone
Name
Judges
Password
1
Event_id
Participant_type
Email
1
Phone
Issued to
Give
Certificate_id
Participant_id
Score_id
Event_id
1
1
Certificates
Decides
1
Scores
1
Event_id
Status
Partcipant_id
Judge_id
Marks


events organizer
organizer_id : int(255)
name : varchar(255)
email : varchar(255)
password : varchar(255)
phone : int(255)
events co_organizer
co_organizer_id : int(255)
name : varchar(255)
email : varchar(255)
phone : int(255)
password : varchar(255)
organizer_id : int(255)
events judges
judge_id : int(255)
name : varchar(255)
email : varchar(255)
phone : int(255)
password : varchar(255)
event_id : int(255)
events events
event_id : int(255)
name : varchar(255)
max_participants : varchar(255)
judges_id : int(255)
description : varchar(255)
co_organizer_id : int(255)
event_date : date
event_time : varchar(255)
event_type : varchar(255)
events scores
score_id : int(255)
event_id : int(255)
judge_id : int(255)
marks : int(255)
participant_id : int(255)
events certificates
certificate_id : int(255)
participant_id : int(255)
event_id : int(255)
status : varchar(255)
events participants
participant_id : int(255)
name : varchar(255)
email : varchar(255)
password : varchar(255)
event_id : int(255)
participant_type : varchar(255)
phone : int(255)

- 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.


Thank you!