concept
event
management

EVENT MANAGEMENT SYSTEM

Mukund W.N.
Sai Kaushik S.
Sathya Swaruban B.S.
Kogul Karna K.V.

ABSTRACT

The project focuses on automating the tasks involved in organising an event. The application is implemented as a website which allows user to organise events once after they register. The server side script will be written in such a way that it authenticates the user user using a confirmation mail to avoid bots. The implementation is done in a modular way using MVC architecture which would provide a high level abstraction to the database handler. The development will ensure that the input strings are sanitized to avoid injection of database.

ENTITY LIST

- USER - This entity includes all the people who have registered with the community and their login details.
- ORGANISER - This entity holds information about the event organiser who is also a user of the community. This entity holds details about the event and cost of organisation.
- PARTICIPANT -This entity holds information about the user who registered with an event.
- EVENT - This entity contains the detail of events that are being organised and also the details about location.
- VOLUNTEERS - This entity contains information about the users who offered to assist the event manager in organising an event.
- LOCATION - This table contain information about various locations that are available to conduct an event.
- PHONE - This entity holds all the phone numbers that are used to register with the community.
- RENT&PHONE - This entity holds information regarding the rent and the contact number of particular location.

ATTRIBUTE LIST

- USER(user_id,name,password,email,city,state,zipcode)
- ORGANISER(o_id,name,email,bill,user_id,e_id)
- PARTICIPANT(p_id,name,email,entry_charge,e_id,v_id,user_id)
- VOLUNTEERS(v_id,name,email,e_id,user_id)
- LOCATION(Location_id,name,city,state,zipcode,availability,owner,)
- PHONE(p_ident,code,number,user_id)
- RENT_PH(R_id,amt,location_id,ph_num)

RELATIONSHIPS - MAPPING CARDINALITY

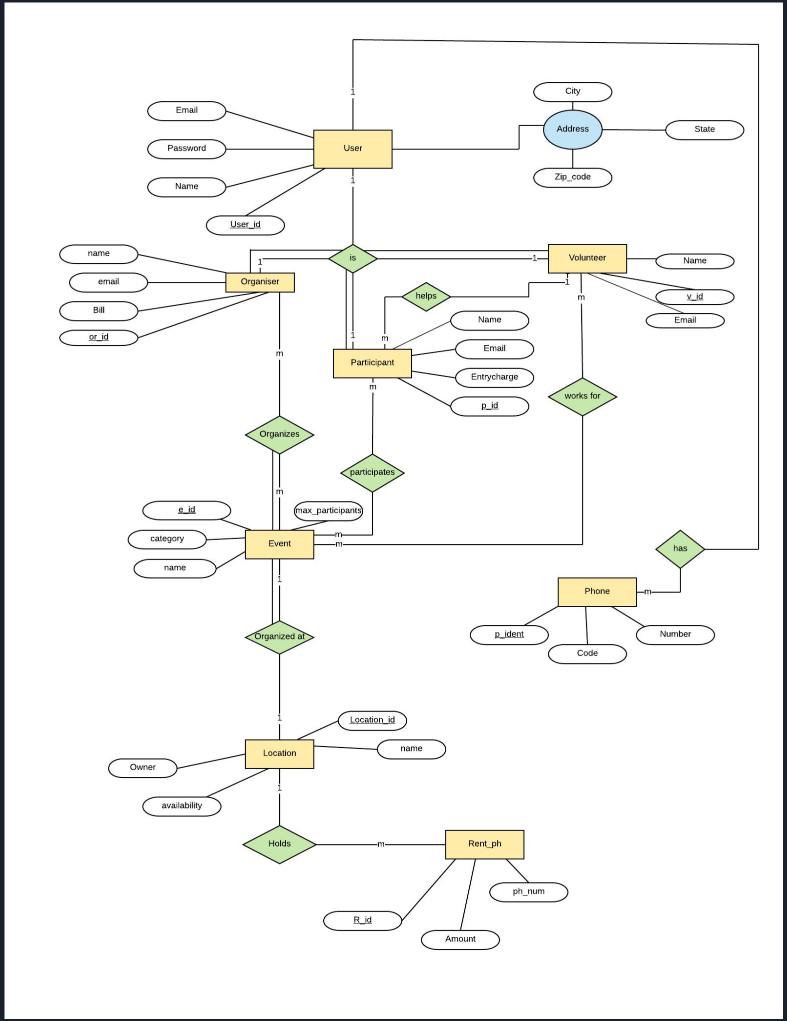
- A user can have more than one phone number and many phone numbers can belong to one user. (1:M)
- A user will be allowed to have only one organizer profile and one organizer profile can belong to only one user. (1:1)
- A user can be a volunteer and that volunteer can be mapped to only that user. (1:1)
- A user can be a participant and that participant can be mapped to one user only. (1:1)
- A volunteer can assist many participants; Many participants may have same volunteer to help them. (1:M)
- A set of volunteers may involve themselves in more than one event or none. It is not necessary that every volunteer needs to have an event associated with them. (M:M)
- Many participants can participate in many events and an event can have many participants. (M:M)
- Many organizers can host many events and an event can be organized by more than one organizer. (M:M)

RELATIONSHIPS - MAPPING CARDINALITY

- An event can be organized only at a particular location provided by the community and that particular location can have only one event hosted at a time. (1:1)
- Each location can have different rent values and contact numbers; Different contact numbers and rent values can pertain to the same location based on demands. (1:M)

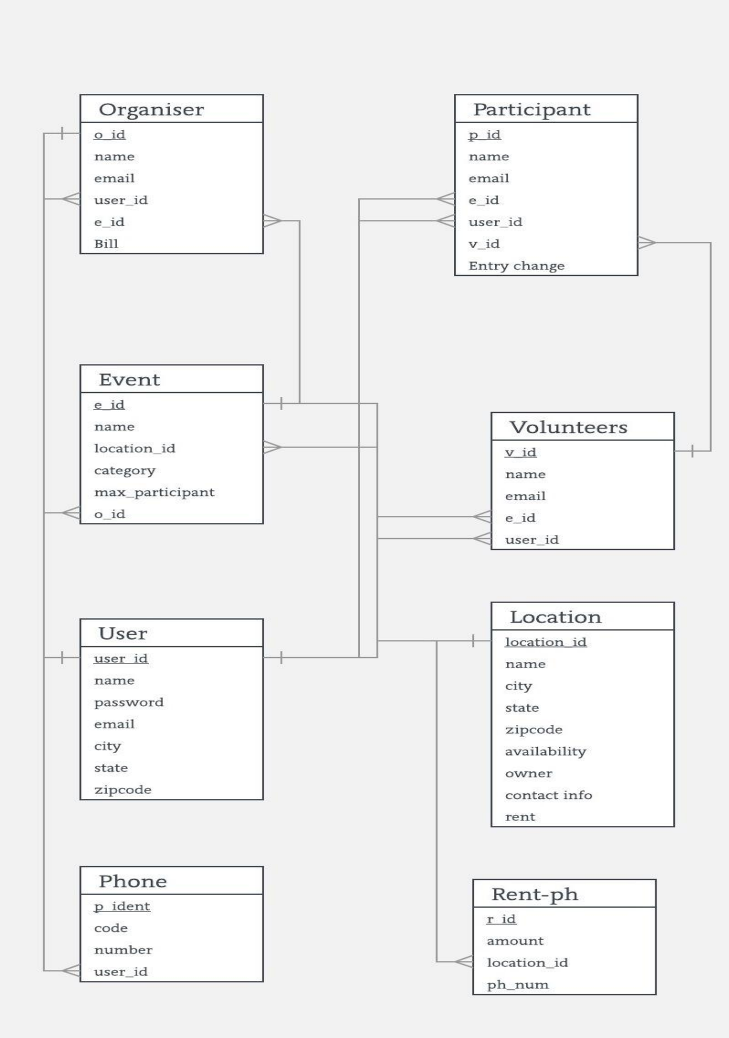
RELATIONSHIPS - PARTICIPATION CONSTRAINTS

- Every event must have at least one organizer; but an organizer need not host an event. (partial - total)
- Every organizer, participant, volunteer must be registered users to interact with the system. But every user need not be an organizer,participant, volunteer; Can be any one of them . (partial - total)
- Every event must have a location, but every location need not necessarily have an event happening. (total - partial)


City
Email
Address
State
User
Password
Name
Zip_code
User_id
name
is
Volunteer
Name
email
Organiser
Bill
helps
v_id
Email
or_id
Name
Email
Partiicipant
Entrycharge
works for
p_id
Organizes
participates
e_id
max_participants
category
Event
name
has
Phone
Organized at
p_ident
Number
Code
Location_id
name
Location
Owner
availability
Holds
Rent_ph
ph_num
R_id
Amount

ER DIAGRAM

SCHEMA DIAGRAM



Organiser
o_id
name
email
user_id
e_id
Bill

Participant
p_id
name
email
e_id
user_id
v_id
Entry change

Event
e_id
name
location_id
category
max_participant
o_id

Volunteers
v_id
name
email
e_id
user_id

User
user_id
name
password
email
city
state
zipcode

Location
location_id
name
city
state
zipcode
availability
owner
contact info
rent

Phone
p_ident
code
number
user_id

Rent-ph
r_id
amount
location_id
ph_num

RELATION 1: USER

ATTRIBUTE LIST: USER (user_id, name, password,email,city,state,zipcode)

PRIMARY KEY: user_id

FOREIGN KEY: none

RELATION 2: ORGANISER

ATTRIBUTE LIST: ORGANISER(o_id,name,email,bill,user_id,e_id)

PRIMARY KEY: o_id

FOREIGN KEY: user_id, e_id

RELATION 3: PARTICIPANT

ATTRIBUTE LIST: PARTICIPANT(p_id,name,email,entry_charge,e_id,v_id,user_id)

PRIMARY KEY: p_id

FOREIGN KEYS: e_id, v_id, user_id

RELATION 4:VOLUNTEERS

ATTRIBUTE LIST: VOLUNTEERS(v_id,name,email,e_id,user_id)

PRIMARY KEY: v_id

FOREIGN KEYS: e_id, user_id

RELATION 5: LOCATION

ATTRIBUTE LIST : LOCATION(Location_id,name,city,state,zipcode,availability,owner,)

PRIMARY KEY: Location_id

FOREIGN KEYS: none

RELATION 7:LOCATION

ATTRIBUTE LIST: PHONE(p_ident,code,number,user_id)

PRIMARY KEY: p_ident

FOREIGN KEY: user_id

RELATION 8: RENT_PH

ATTRIBUTE LIST: RENT_PH(R_id,amt,location_id,ph_num)

PRIMARY KEY: R_id

FOREIGN KEY: location_id

IMPLEMENTATION

FRONT END : HTML, CSS , JAVASCRIPT.

BACK END: PHP, MySQL.