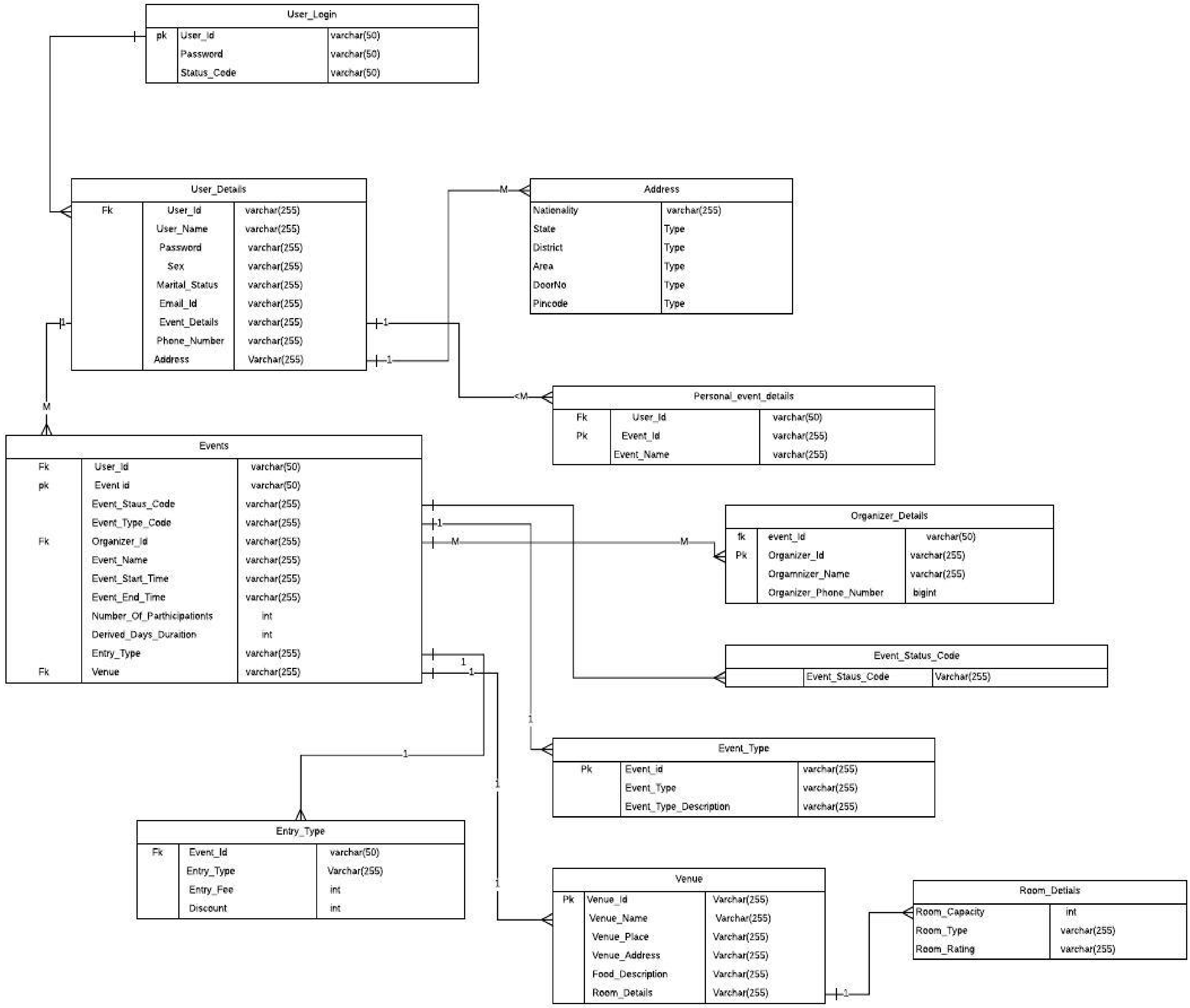
FLOWCHART:



User_Login

pk	User_Id	varchar(50)
	Password	varchar(50)
	Status_Code	varchar(50)

User_Details

Fk	User_Id	varchar(255)
	User_Name	varchar(255)
	Password	varchar(255)
	Sex	varchar(255)
	Marital_Status	varchar(255)
	Email_Id	varchar(255)
	Event_Details	varchar(255)
	Phone_Number	varchar(255)
	Address	Varchar(255)

Address

Nationality	varchar(255)
State	Type
District	Type
Area	Type
DoorNo	Type
Pincode	Type

Personal_event_details

Fk	User_Id	varchar(50)
Pk	Event_Id	varchar(255)
	Event_Name	varchar(255)

Events

Fk	User_Id	varchar(50)
pk	Event id	varchar(50)
	Event_Staus_Code	varchar(255)
	Event_Type_Code	varchar(255)
Fk	Organizer_Id	varchar(255)
	Event_Name	varchar(255)
	Event_Start_Time	varchar(255)
	Event_End_Time	varchar(255)
	Number_Of_Parthicipationts	int
	Derived_Days_Duraition	int
	Entry_Type	varchar(255)
Fk	Venue	varchar(255)

Organizer_Details

fk	event_Id	varchar(50)
Pk	Organizer_Id	varchar(255)
	Orgamnizer_Name	varchar(255)
	Organizer_Phone_Number	bigint

Event_Status_Code

	Event_Staus_Code	Varchar(255)

Event_Type

Pk	Event_id	varchar(255)
	Event_Type	varchar(255)
	Event_Type_Description	varchar(255)

Entry_Type

Fk	Event_Id	varchar(50)
	Entry_Type	Varchar(255)
	Entry_Fee	int
	Discount	int

Venue

Pk	Venue_Id	Varchar(255)
	Venue_Name	Varchar(255)
	Venue_Place	Varchar(255)
	Venue_Address	Varchar(255)
	Food_Description	Varchar(255)
	Room_Details	Varchar(255)

Room_Detials

Room_Capacity	int
Room_Type	varchar(255)
Room_Rating	varchar(255)

```
QUERIES:
-------------


** user_login **
create table if not exists user_login(
    user_id                varchar(20),
    password               varchar(255)    notnull,

    constraint      user_log_user_id       primary key(user_id)
);




** user details **

create table if not exists user_details(
    user_id           varchar(255),
    image             blob,
    first_name         varchar(20)         notnull,
    last_name          varchar(20)         notnull,
    gender                              varchar(1)          notnull,
    marital_status     varchar(1)          notnull,
    date_of_birth      date                notnull,
    email_id           varchar(50)         notnull,
    mobile_number      varchar(50)         notnull,
    address_id         varchar(10)         notnull,



    constraint     foreignkey(user_id)              referrenced(user_log_user_id);
    constraint     user_details_email_id             primary key(email_id),
    constraint     user_details_address_id           primary key(address_id),
    constraint     user_details_marital_status_chk     check(marital_status in("SINGLE","MARRIED"))
    constraint     user_details_gender_chk           check(gender in("MALE","FEMALE"))
);



** Address **
create table if not exist address(
    user_id           varchar(20),
    nationality        varchar(20)         notnull,
    state             varchar(20)         notnull,
    district           varchar(20)         notnull,
    area              varchar(20)         notnull,
    doorno          varchar(20)        notnull,
    pincode         int                notnull

    constraint  foreignkey(user_id)    referrenced(user_details_user_id);

                                   );
```

** Personal_events **

```
create table if not exists personal_events(
  user_id              varchar(20),
  event_id             varchar(20)        notnull,
  event_name           date               notnull,
  event_description    varchar(255)       notnull,
  event_date           date               notnull,


  constraint  foreignkey(user_id)        referrenced(user_login_user_id),
  constraint  personal_events_event_id     primary key

);
```

** organizer_details **

```
create table if not exists organizer_details(
  event_id                  varchar(20),
  organizer_id              varchar(20),
  organizer_name            varchar(20)      notnull,
  organizer_mobile_number        varchar(20)      notnull,

  constraint  foreignkey(event_id)           referrenced(personal_events_event_id),
  condtraint  organizer_details_organizer_id     primary key(organizer_id)


);
```

** event_type **

```
create table if not exist event_type(


    event_id              varchar(20),
    event_type             varchar(20)       notnull,


    constraint  foreignkey(event_id)            referrenced(personal_events_event_id),
    constraint  event_type_chk                  check(event_chk in("COMEDY","MUSIC","DANCE","PARTY","FAMILY_FUNCTION"))

);
```

** venue details **

```
create table if not exist venue_details(
    venue_id              varchar(20),
    event_id              varchar(20),
    venue_place            varchar(50),
    venue_address           varchar(100),
    food_description         varchar(100),
    room_details            varchar(100),

     constraint  foreignkey(event_id)            referrenced(personal_events_event_id)



);
```

** entry_type **

```
create table if not exist entry_type(
      event_id          varchar(20)          ,
      entry_type         varchar(20)          notnull,
      entry_fee         int                 notnull,
      discount          int                 notnull,

                                  constraint  entry_type_entry_type           check(entry_type in("FEES","FREE","DISCOUNT"))
      constraint  foreignkey(event_id)            referrenced(personal_events_event_id)
);
```

** Making-Events **

```
create table if not exist making_events(
    user_id                  varchar(20),
    event_id                 varchar(20),
    event_name               varchar(20)      notnull,
    event_status_code        varchar(20)      notnull,
    event_type               varchar(20)      notnull,
    organizer_id             varchar(20),
    event_start_time         varchar(20)      notnull,
    event_end_time           varchar(20)      notnull,
    number_of_participation      int          notnull,
    derived_duration             int          notnull,
    entry_type               varchar(20),
    venue_id                 varchar(20),

    constraint  foreignkey(user_id)           referrenced(user_login_user_id),
    constraint  foreignkey(event_id)          referrenced(personal_events_event_id),
    constraint  making_events_event_status_code    check in ("cancelled","approved","amount_paid","finished","processing")

);
```