DOCUMENTATION

1.) System Requirement specification (SRS)

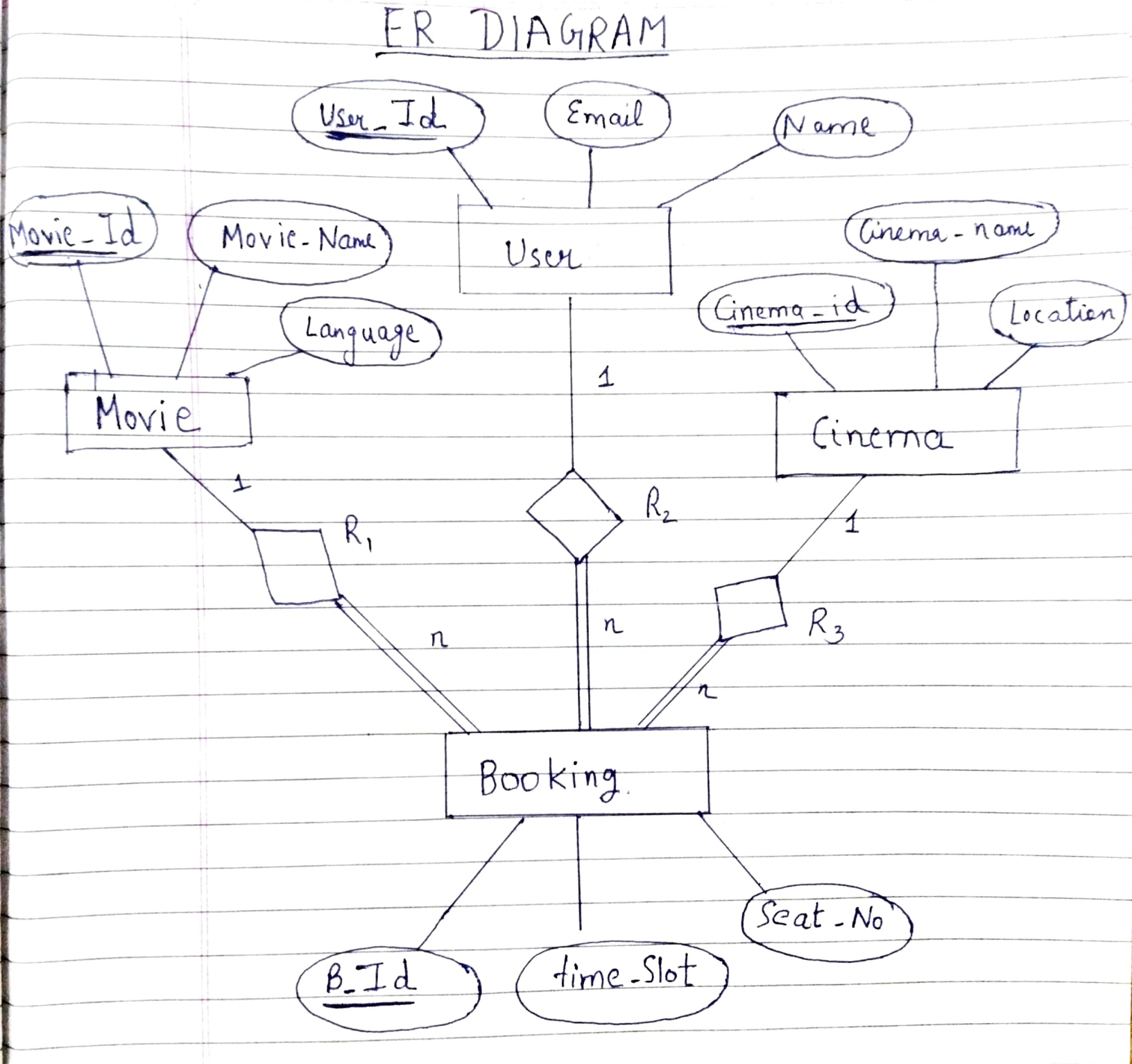
The online movie ticket web application will include various functionalities for the users that are required for booking a movie seat in a cinema hall.

The application will have a user page .Once the user enters the user page, he will see the options of booking and cancelling the movie ticket.

When the user clicks on the booking option, he will see the next page where he has to enter his name, email address, movie name, movie language, cinema name, cinema location, time slot in which he wants to watch the movie and his preferred seat number. Once he clicks on the Book Ticket button, his details will be added in our database and hence his ticket will be booked.

User can also delete his booking if he wants to, by going to the cancel booking option in the user page. Then, he has to give some details on that page which includes his name, email address, movie name, movie language, cinema name, cinema location, time slot, seat number. Once he clicks on the Cancel Ticket button, his details will be removed from our database and hence his ticket is cancelled.

ER DIAGRAM


User_Id
Email
Name
User
Movie_Id
Movie_Name
Language
Movie
Cinema_name
Cinema_id
Location
Cinema
1
1
1
R_1
R_2
R_3
n
n
n
Booking
B_Id
time_Slot
Seat_No

System modeling: Schema design

1) Movie(Movie_Id, Movie_Name, Language)
2) User(User_id, Email, Name)
3) Cinema(Cinema_Id, Cinema_Name, Location)
4) Booking(B_Id,Movie_Id, User_Id,Cinema_Id, Time_Slot, Seat_No)

System modeling: Data Normalization

We have created relations which are in BCNF. Due to this, there is no update enamoly, no insert enamoly ,no delete enamoly and lesser data redundancy.

Insert enamoly: This means that if we add a new entry, we do not have to make the unknown entities as null. For eg, if admin adds a movie and if we had a single table for all entries, then we had to keep the other entries as null. This enamoly is insert enamoly and is removed by data normalization. Also due to data normalization, our database becomes less redundant.

Update Enamoly: for eg, if I want to update a movie name because it got changed due to some reason, then data normalization will help to change the data in a particular table and not change everywhere if it had been a single table.

Delete Enamoly: for eg, if I want to delete a movie name, then data normalization will help to delete the data in a particular table and not delete everywhere if it had been a single table.

System modeling: List of Tables Required

1) Movie Table: which lists the movie Id, movie name and movie language.
2) User Table: which lists the user Id, user email, user name.
3) Cinema Table: which lists the cinema Id, cinema name, cinema Location.
4) Booking Table: which lists the booking Id, Movie Id, User Id, Cinema Id, Time Slot, Seat No.

System modelling: Additional Components

The system requires installation of Nodejs.