UE15CS256: Database Management Systems Lab

Mini Project

“Cinema Booking Application”

**Team Members:**

1. Ravi Shreyas Anupindi 01FB15ECS234

2. Rakesh K 01FB15ECS230

3. Aditya Singh 01F15ECS218

**Problem Statement:** Using an RDBMS to implement a database for a single multiplex, providing functionality for booking tickets.

**Project Description:**

The database is modeled using an ER diagram, where the entities are:

Movie, Actor, Director, Show, Auditorium, Seat, Booking, Customer and Picture format.

Each movie is uniquely identified by its title. Actors of the movie and its director are fetched from the corresponding relationships between movies and the the other entities.

A movie also has a specific format, whether it is 2D,3D or IMAX, as well as the frame rate of the picture.

Each show is modeled as such: the show is uniquely identified by show id, date and time. Each show screens a movie and is hosted in an particular auditorium.

The auditorium entity is uniquely identified by the auditorium number.

Each auditorium can contain many seats, which is an entity itself. Each seat is identified by its row number and column number. Each seat also holds information as to its price as well as availability(booked or free).

To identify all seats for a show, the auditorium is referenced followed by all seats within that auditorium. The availability attribute of the seat entity helps to query all available seats.

Each customer is an object of the Customer entity set and is uniquely identified by their email id. Customers’ names as well as phone numbers are held. To complete transactions, each customer has a e-wallet as an attribute, denoting their current balance. Funds can be added via simple update queries.

Booking entity is a weak entity whose owner is the Customer entity. Each booking is related to a show and to seats as well. This way to display all the details for that booking, we query the relationship tables.

**Possible Front End:** The RDBMS selected is PostgreSQL. Using a Flask micro-framework for the back-end, the application will feature a web interface. The framework uses Python for its backend. SQLAlchemy extension for flask acts as a Object Relation Mapper(ORM) which will interface PostgreSQL with the web-app. The ORM is necessary as Flask implements an Object oriented DBMS. The show timings and other front end functionality will be implemented by querying the database.

**Project Outcomes:** At the end of the mini-project, we expect a minimal, clean front end interface for the cinema app featuring a robust database, capable of complex queries.