DATABASE MANAGEMENT SYSTEMS

PROJECT ABSTRACT

TITLE- PEER TO PEER BOOK RENTAL SYSTEM

Team Name: **Team KNK**

Team Members:

Kaushik Narayan R – 2019103536
Nanda Kumar R – 2019103545
Kiran George Gaurdian – 2019103029

B.E. COMPUTER SCIENCE AND ENGINEERING
Q Batch

The project, “P2P (Peer to Peer) Book Rental System” is developed to be a full stack web application, that maintains a record of users and their locations, books, transactions, requests, and offers, along with added features which will be discussed shortly.

The following is the stack that is implemented for this project.

Front-End:	ReactJS v17.0.2
Back-End:	Express v4.17.1
Database Tools:	PostgreSQL v13.2, pgAdmin4 v5.2, PostgreSQL CLI, plsql-supporting procedural language debugger
Database Server:	PostgreSQL Server 13

Functionalities and features of the System:

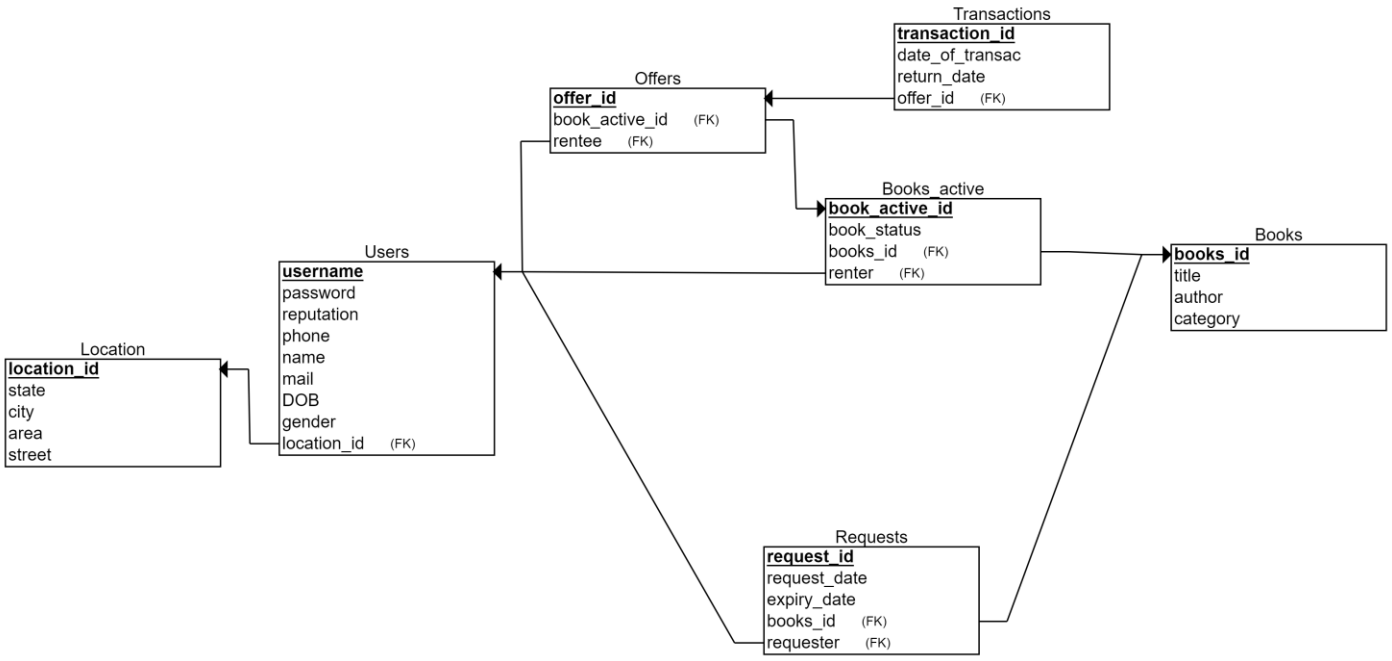
The system maintains:

- data of the users, who may be a buyer or seller or both
- data about the details of transactions that happen between the buyer and the seller
- static data of the location of the user
- data of the books that may or may not be put up for sale, including that regarding the status of the books that have been active in circulation
- a record of offers for books in circulation that have been made by buyers, which may or may not be accepted by the seller
- a record of requests made by buyers for a particular book

The application has a **reputation system**, which is a system that formalizes the process of gathering and distributing information about a user’s past behavior. With this system, depending on the number of transactions the user has participated in, the reputation of the user may increase or decrease.

Another feature is a **request system**, where a buyer can request for a particular book that may not be recorded as part of the database at that instance of time, but the request may be fulfilled at a later point in time by a seller.

Relational schema:



Database Relations:

1) **Users:**

Attributes:

- username - primary key that uniquely identifies a user
- password, reputation, phone, name, mail, DOB, gender
- **location_id** – foreign key that references **location_id** attribute in the relation, **"Location"**

2) **Location:**

Attributes:

- location_id – primary key that uniquely identifies the location for a user
- state, city, area, street – non-prime attributes

3) **Books:**

Attributes:

- books_id – primary key that uniquely identifies a particular book
- title, author, category – non-prime attributes

4) **Books_active:**

Attributes:

- book_active_id – primary key that uniquely identifies an active book in circulation
- book_status - non-prime attribute
- **books_id** - foreign key that references **books_id** attribute in the relation, **"Books"**
- **renter** – foreign key that references **username** attribute in the relation, **"Users"**

5) **Transactions:**

Attributes:

- transaction_id – primary key that uniquely identifies a transaction between a buyer and a seller
- date_of_transac, return_date – non-prime attributes
- **offer_id** – foreign key that references **offer_id** attribute in the relation, **"Offers"**

6) **Offers:**

Attributes:

- offer_id - primary key that uniquely identifies an offer made by a renter for a book in circulation
- **book_active_id** – foreign key that references **book_active_id** attribute in the relation, **"Books_active"**
- **rentee** – foreign key that references **requester** attribute in the relation, **"Requests"**

7) **Requests:**

Attributes:

- request_id – primary key that uniquely identifies a request made by a buyer
- request_date, expiry_date – non-prime attributes
- **books_id** – foreign key that references **books_id** attribute in the relation, **"Books"**
- **requester** – foreign key that references **username** attribute in the relation, **"Users"**

All the relations under the database are verified to be **Boyce-Codd Normalized**.

Entity – relationship diagram: