CS 353 - Database Systems

Project Proposal System

Room / Flat Rental Application

Group 18

Mehmet Feyyaz Küçük - 22003550
Ender Utlu - 22001983
Ege Ayan - 22002478
Deniz Çelik - 22003271
Parsa Keihan - 22001422

1. Introduction

This report is a proposal for a Room / Flat Rental Web Application that comprises a database system. The application consists of different user types and the interaction between them. Homeowners can put their rental dwellings on the website and mention the related features such as available dates, location, etc. Travelers can rent a domicile by searching for a specific date in an area. Then, they can look for mentioned features and book the place using the balance system. Admins are the usual controllers of the general system. The system will provide a secure in-app balance payment system to book the places before. Also, each advert would be controlled by the system before the final release. In the following subsections, a detailed project description, functional and non-functional requirements, constraints, and limitations are cited. At last, there is an entity-relationship model to describe real-world objects.

2. Project Description

2.1 Why Is a Database a Part of the System

In this Room / Flat Rental Application, most of the features require many types of data. To keep all of the data, we will need an efficient, consistent, secure, scalable, and integrated system. A database will provide all of our needs and allow us to easily manage the data that will be used in different stages of the apps' functions. A database will also help us manage the data the users will interact with such as rental advert information. A database will let us keep all the information the users will interact with in a reliable and scalable way.

2.2 How Is a Database a Part of the System

The application will have multiple types of accounts that will interact with each other and be able to share certain information, such as a homeowner's flats location, features, available room count, and restrictions. User reviews will be held in the database so that users can have an idea about the rental flat. These reviews' average will be the reputation of the homeowner so travelers won't need reviews for a specific flat. There will be other interactions like these in the application to share the necessary information with our users.

3. Requirements

3.1. Functional Requirements

3.1.1. Traveler

- Traveler users can register and create a new account by providing a username and password.
- Traveler users can change their current password with a new one.
- Traveler users can reset their passwords if they forgot the current one.
- Traveler users can book a room.
- Traveler users can book flats.
- Traveler users can give ratings to rentals.
- Traveler users can give ratings to homeowners.
- Traveler users can write comments about the rentals.
- Traveler users can filter and search rentals by specifying their features. (flat type, location, etc.)
- Travelers can review the features of the rentals.
- Traveler users can review the comments they had written before.
- Traveler users can review their balance.
- Traveler users can delete their accounts.
- Traveler users can ask questions about the flat.

3.1.2. Home-owner

- Home-owner users can perform all the functionality that traveler users can.
- Home-owner users can put their flats up for rent.
- Home-owner users can put their rooms up for rent.
  Home-owner users can specify the details of the rental. (type of accommodation, available dates, location features, restrictions)
- Home-owner users can review their reputation status.
- Home-owner users can review the comments that are written about their rentals.
- Home-owner users can review their current rentals.
- Home-owner users can reply to the questions about flats traveler users ask.

3.1.3. Admin

- Admin users can create system reports. (like finding the most popular rental houses within a region)
- Admin users can review all of the users that are registered in the database system.

3.2. Non-functional Requirements

3.2.1. Security

- Since we will be storing sensitive user information such as a flat address, password and etc. in our database, authorization, and authentication of the user, also encryption and decryption of the data is important.

3.2.2. Performance

- Ideally, the user should not wait more than 1 second for each action. Queries should give fast responses.

3.2.3. Usability

- The application should have a user-friendly UI.
- Since the database will be quite large, the users should be able to find the flat they are looking for with the help of a search filter.

3.2.3. Reliability

- Unexpected crashes from the server or the client side should not result in data loss.

3.2.4. Scalability

- The performance requirement should be independent of database size. In other words, the database must grow without the application losing performance.
- The application should be manageable as it might be required to be updated. The E/R design should allow other entities and relationships to be easily added. This also affects the performance requirement as a significant number of entities and relationships might decrease response times.

3.3. Constraints (Pseudo-requirements)

- HTML, CSS, JavaScript, and React will be used for the front end.
- Java and Spring Boot will be used for the backend.

- PostgreSQL will be used for the database management system.

4. Limitations

- Only homeowners can put a property on rent.
- Homeowners cannot reach travelers' information besides their names and contact information.
- Homeowners cannot put reviews about their adverts.
- Travelers cannot put reviews about properties that they didn’t rent before.
- Only homeowners can add or change the features of a property.
- Homeowners cannot rent their own property.
- Admins cannot rent or advert.
- Password length needs to be between 8-15 characters.
- Passwords must contain at least one special, upper-letter, and lower-letter character.
- Users need to be at least 18 years old.

5. Entity Relationship Model

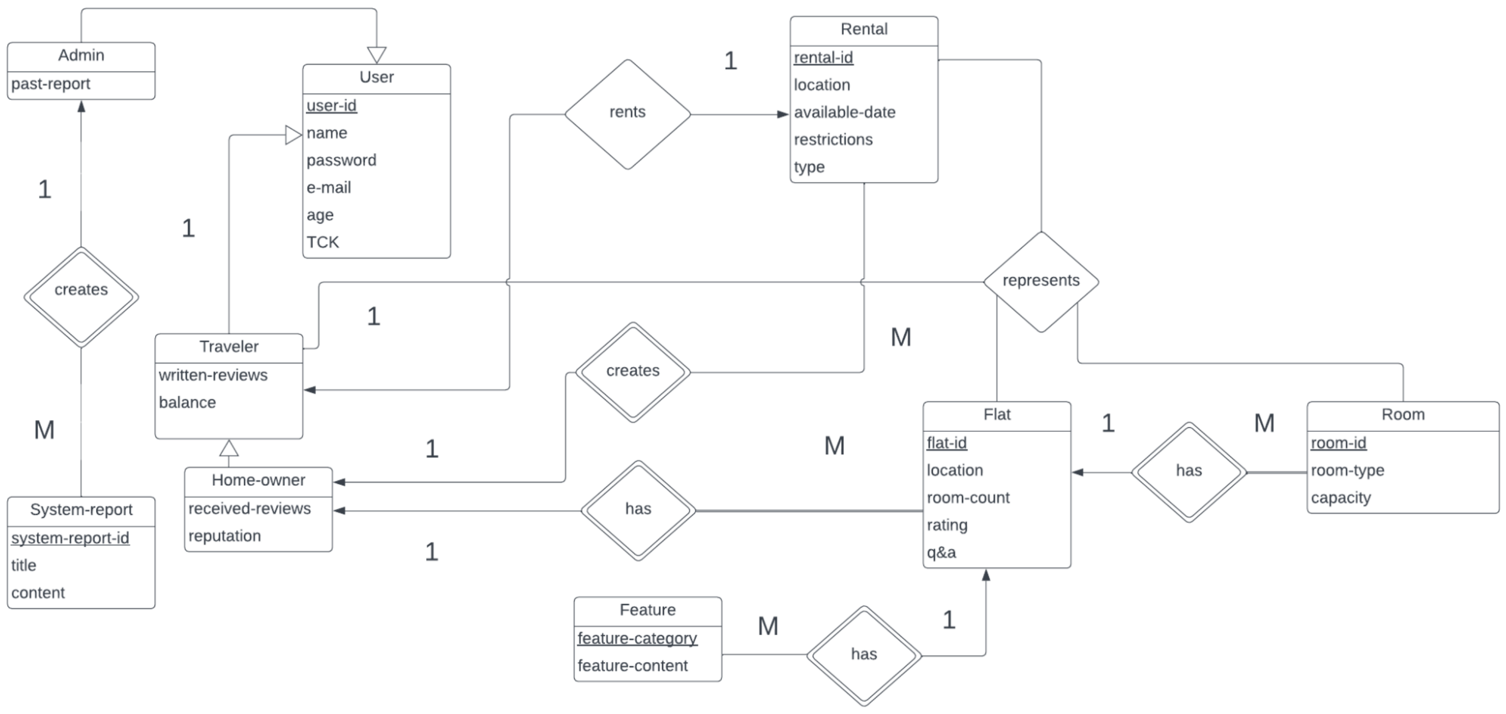

Fig 1: Flat / Room rental application E/R model