PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

Department of Computer Science & Engineering

Session: Jan - May 2022

Object-Oriented Analysis and Design with Java - Laboratory UE19CS353

Mini Project

Report on

Airbnb Clone

By:

S Ritesh Sapata - PES1UG19CS403

Sachin Shenoy - PES1UG19CS408

Sahana Ramesh - PES1UG19CS413

6th Semester G

1. **Project Description**

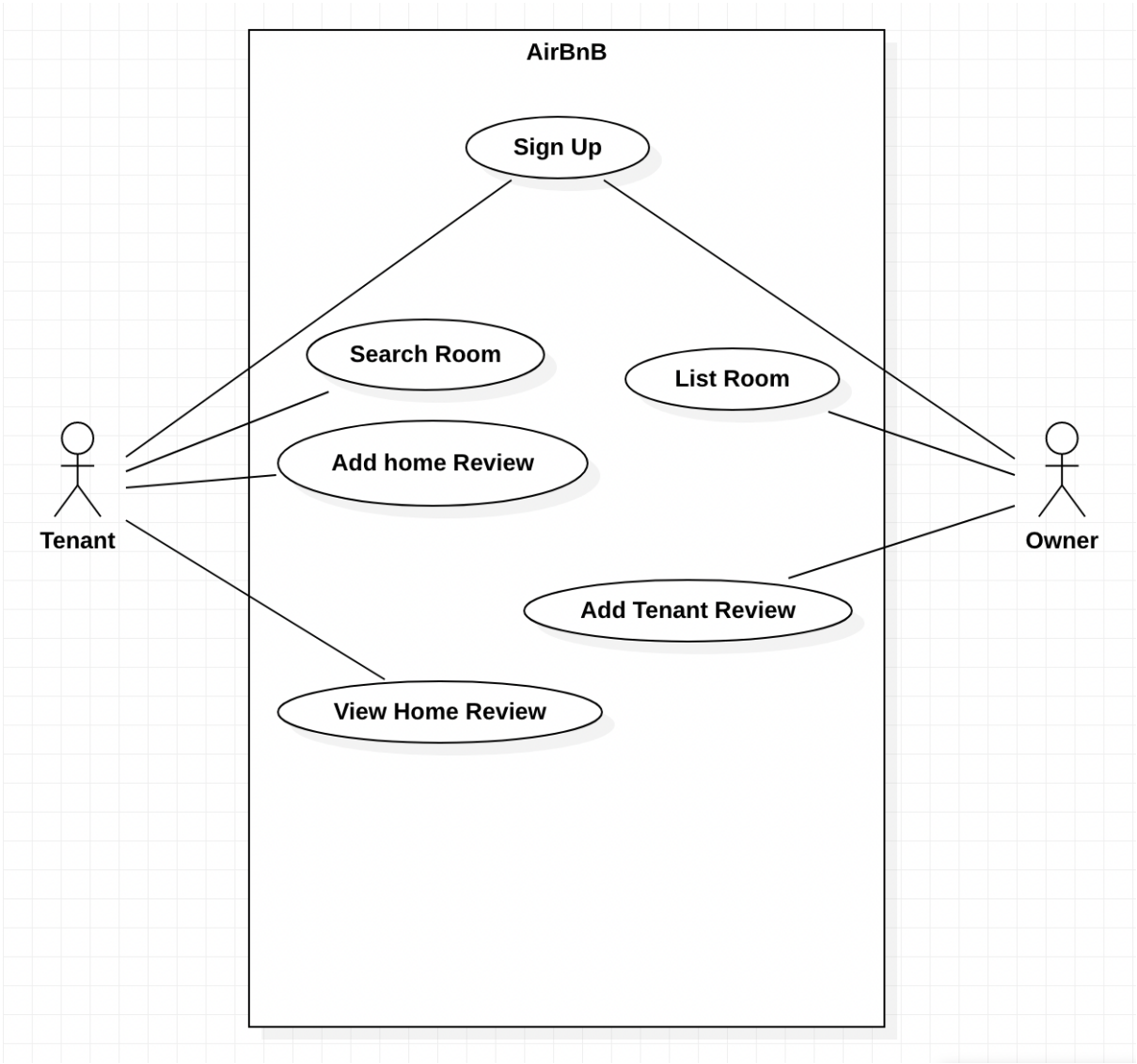
   Our project is a clone of the Airbnb application with the following functionalities:

   1. **Login for tenants (users) and owners (can rent out their homes for homestay)**
   2. **Add a room and review tenants for owners**
   3. **Search for rooms by city, review owners, and see room reviews for tenants**

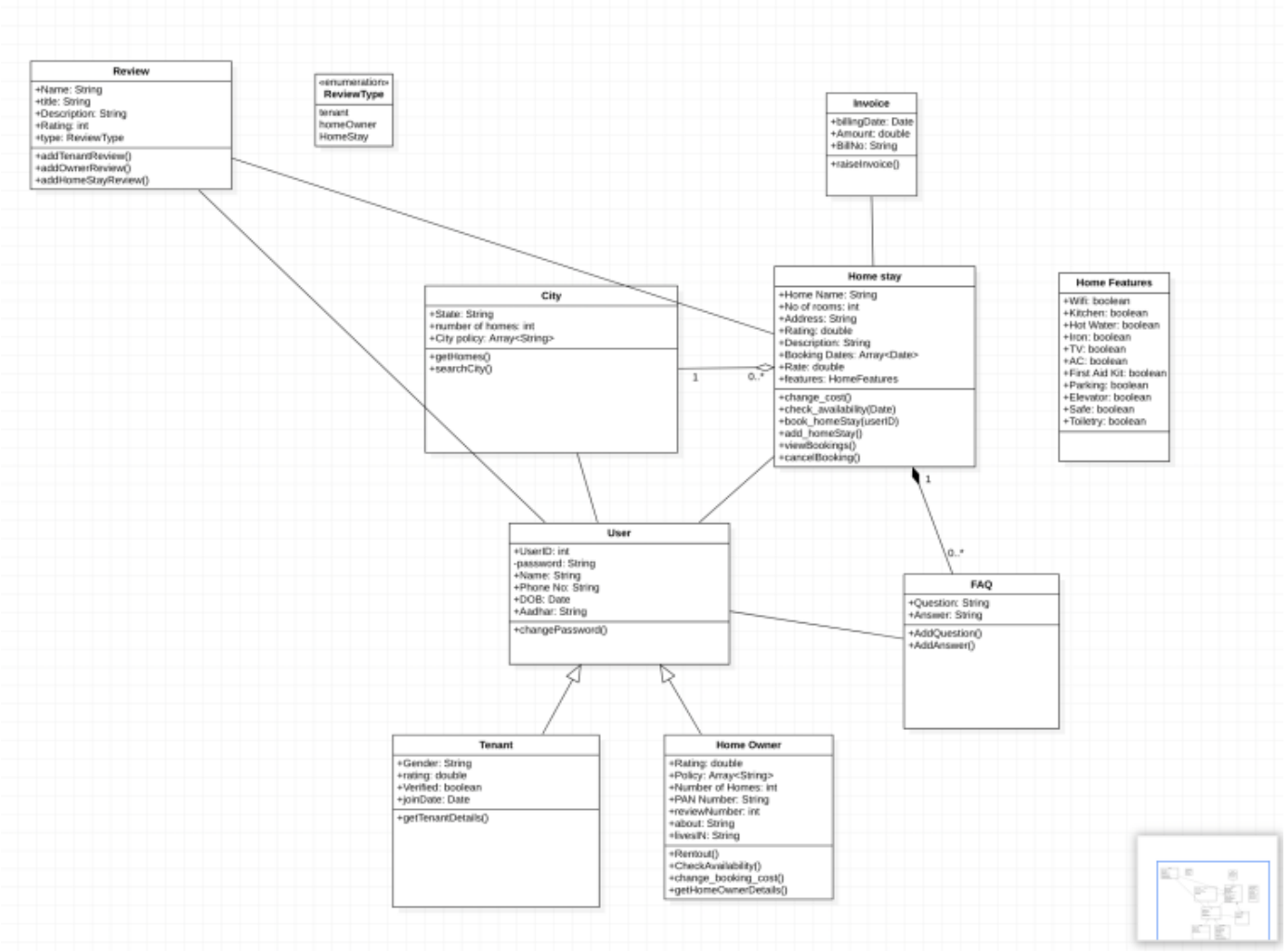
   https://github.com/RIT3shSapata/AirBnb

2. **Analysis and Design Models**

   Use case diagram

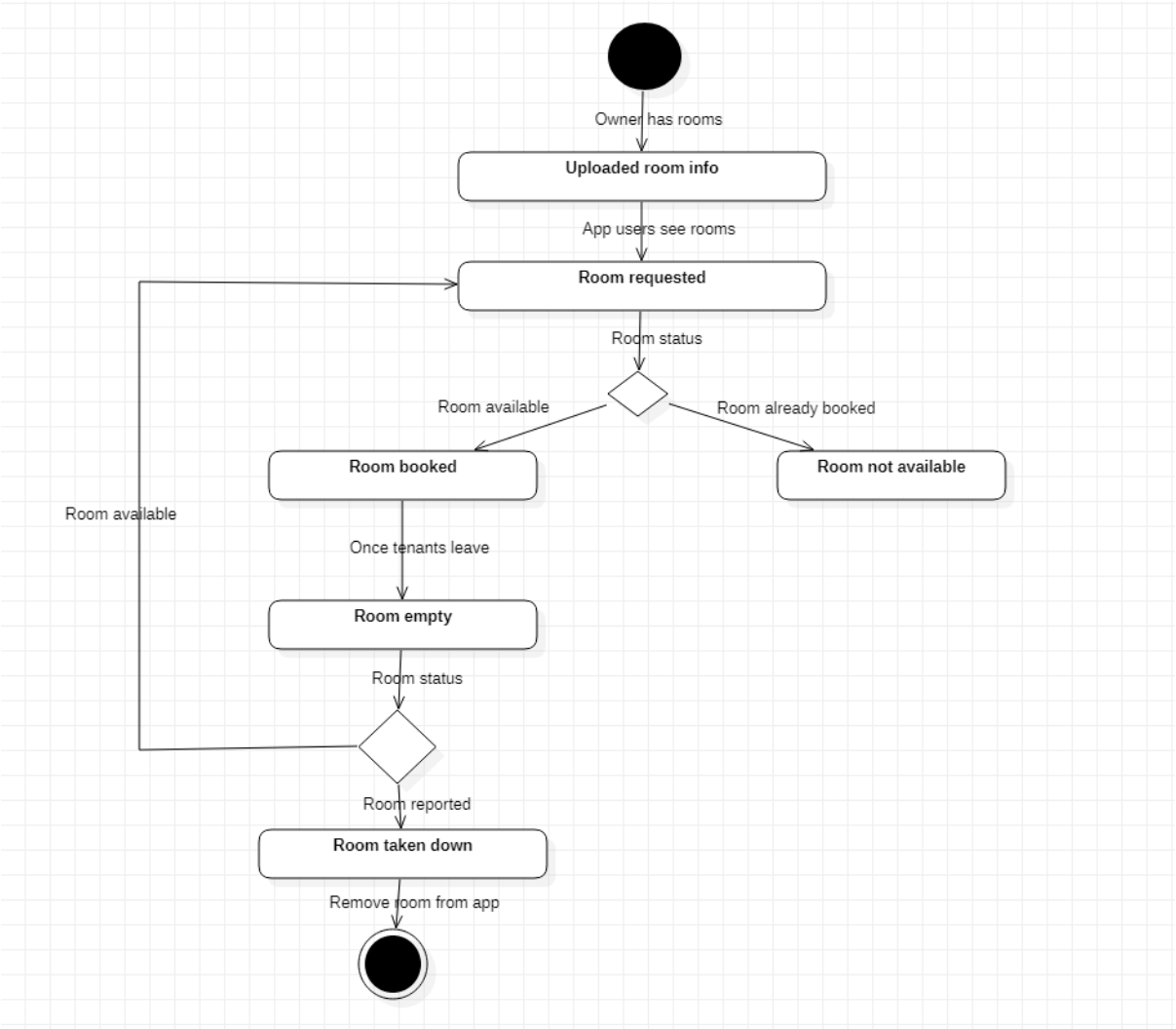


Class Diagram

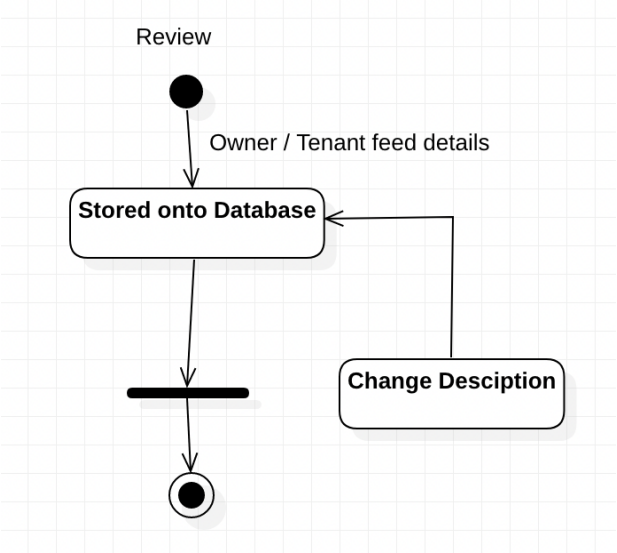


State diagram for the classes with temporal behavior

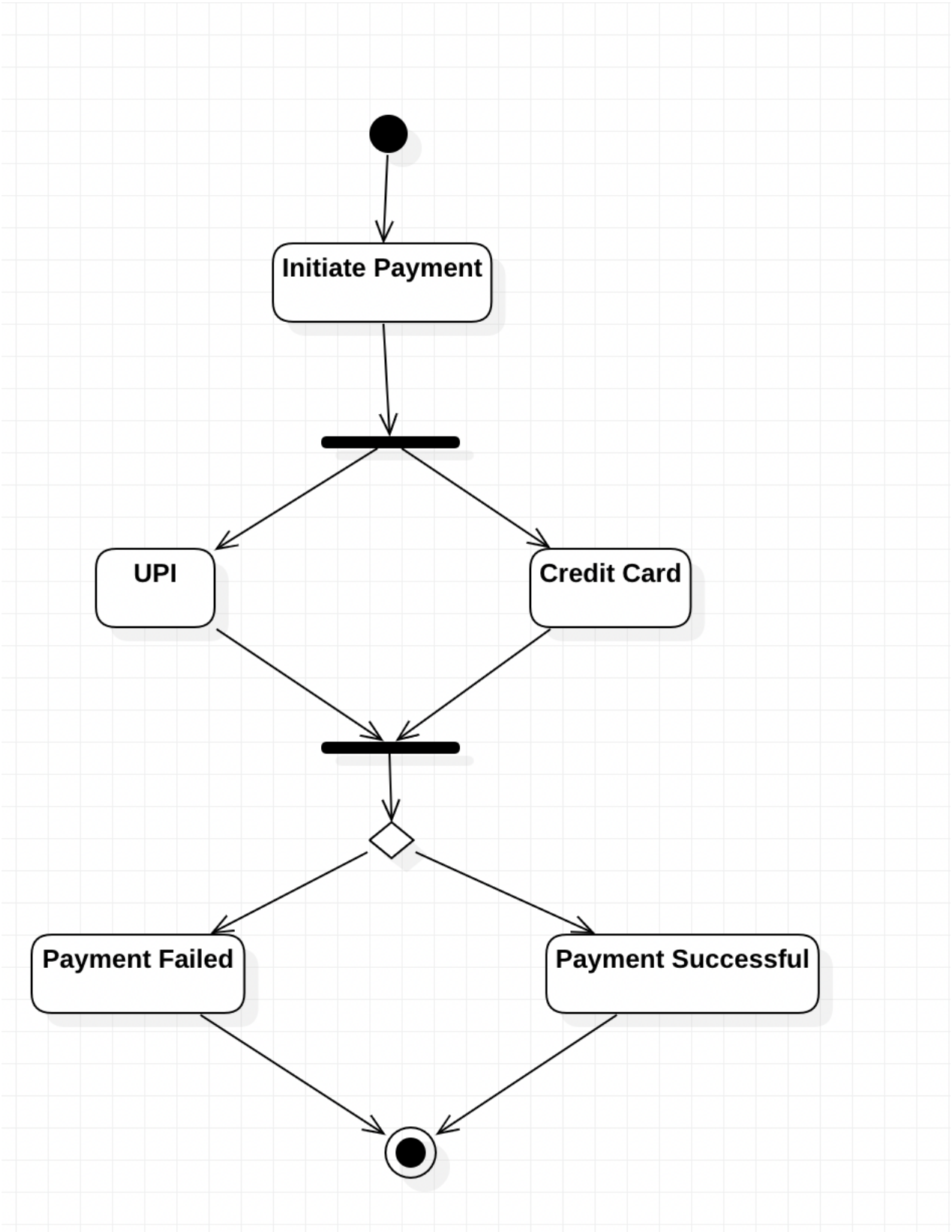
a. Room



b.Review

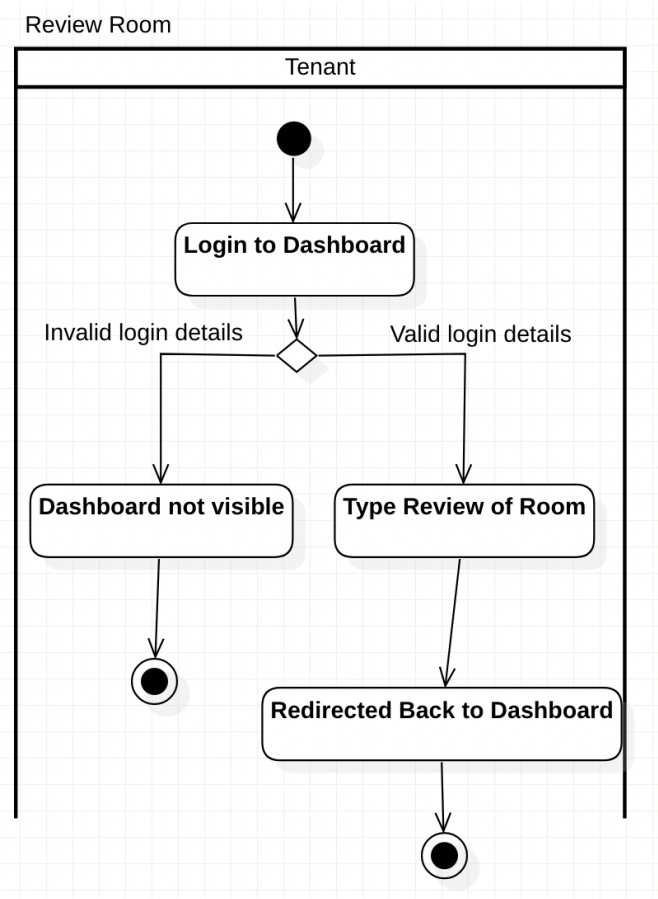


c. Payment

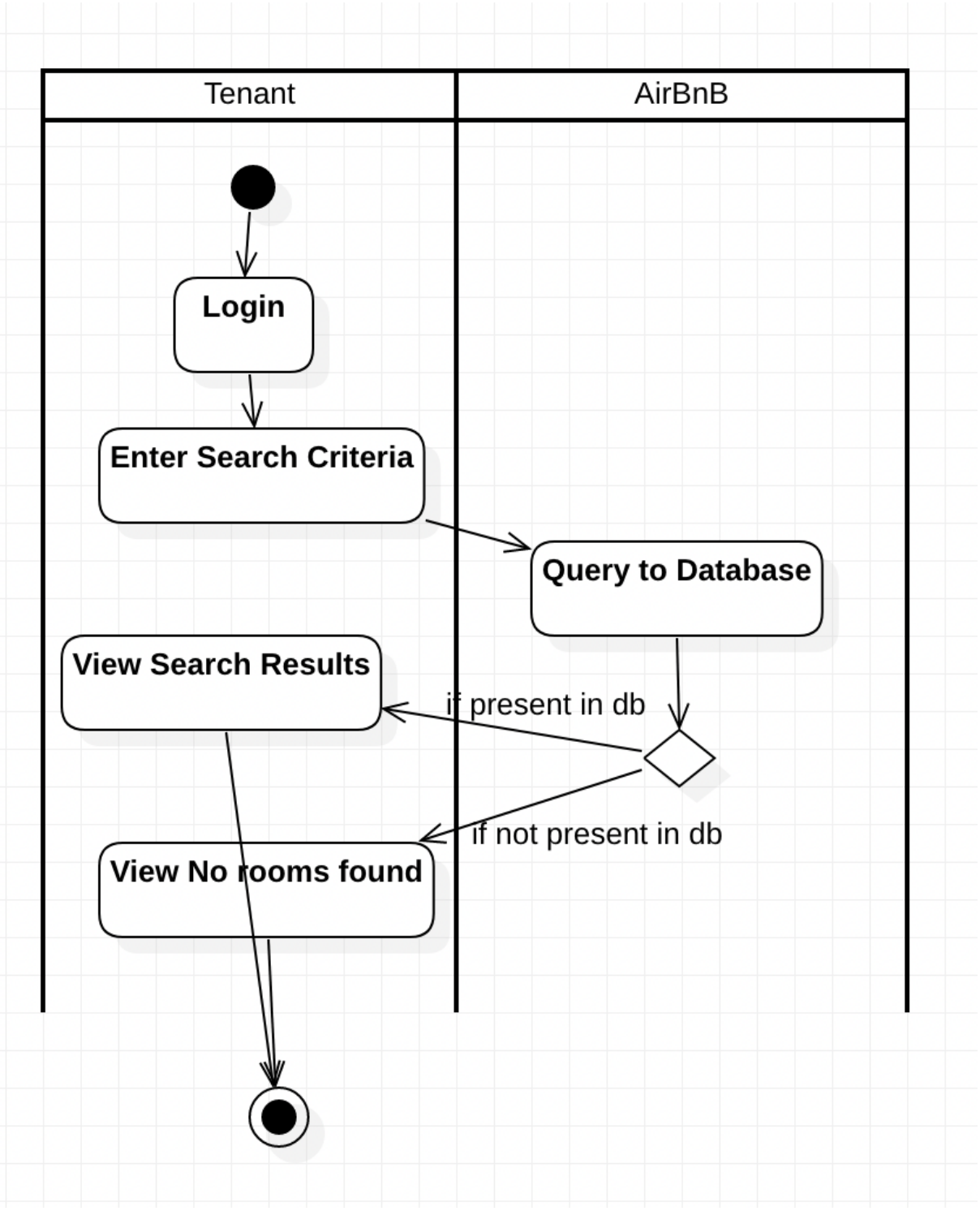

Initiate Payment
UPI
Credit Card
Payment Failed
Payment Successful

Activity diagram for the use cases implemented

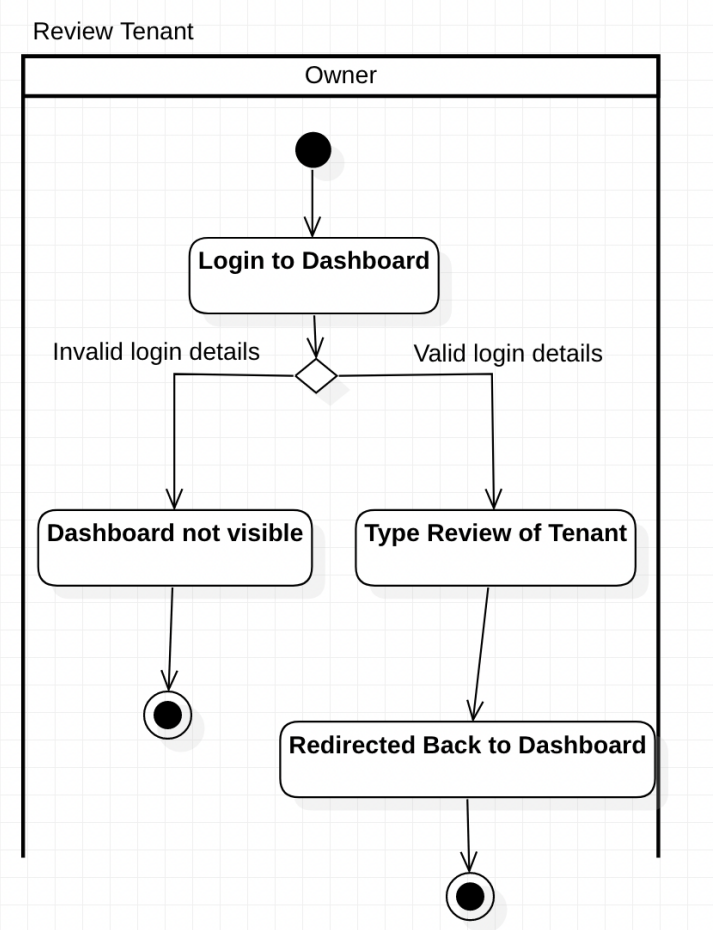
a. Add Room Review



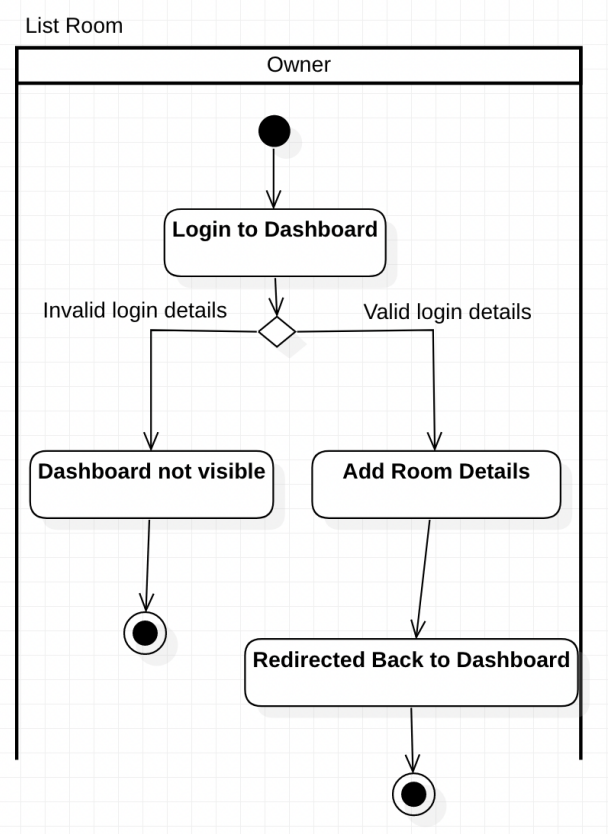
b. Search Room



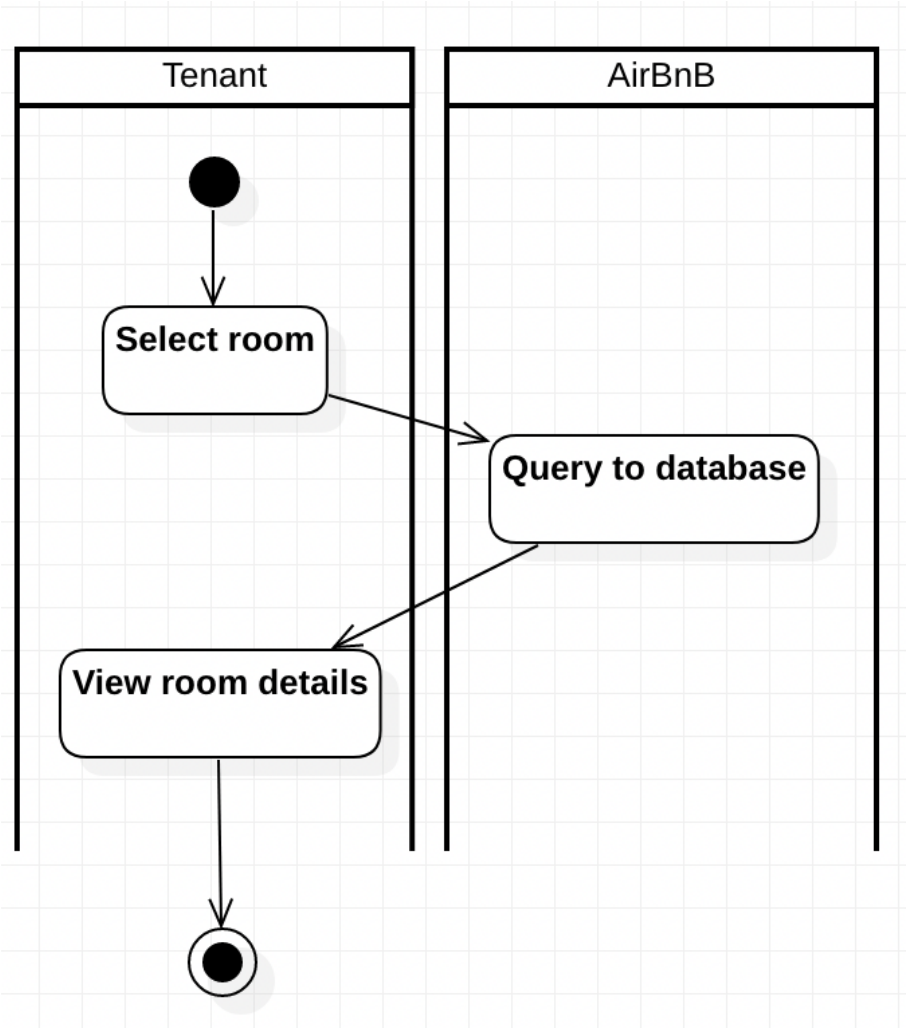
c. Add Tenant Review



d. List Room



c. View Rooms



3. **Tools and Frameworks Used:**

<u>Spring-Boot</u> - Spring Boot was the Backend framework which we chose to build this Web application, which follows MVC architecture. Spring boot has default configurations that allow faster bootstrapping and also the added benefit of dependency injection and out-of-the-box modules.

<u>Thymeleaf</u> - Thymeleaf is a template engine that enabled us to use static template files in our application. At runtime, the template engine replaces variables in a template file with actual values and transforms the template into an HTML file sent to the client. This approach makes it easier to design an HTML page.

PostgreSQL - PostgreSQL is the database management system we used for the application. We used Spring JPA and Hibernate which understand the mappings between the tables and the objects to connect and push data to the database.

Spring-Sessions - Used to store the client data sessions on the browser, helps in storing the state of the application.

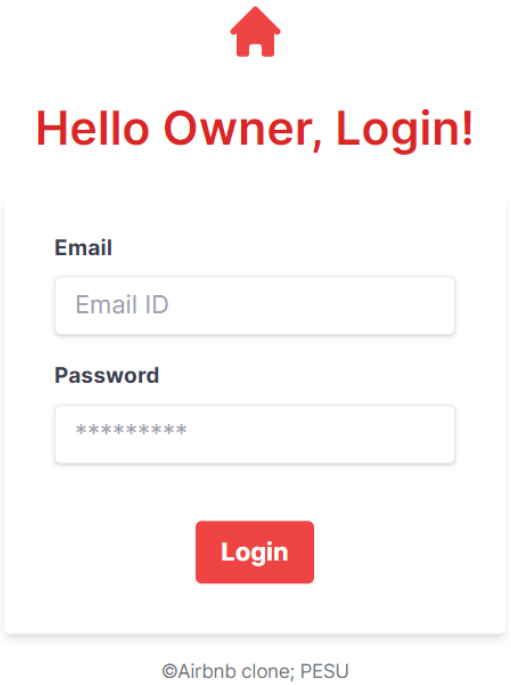
HTML - HTML was used for the front-end forms and display pages.

TailwindCSS - The CSS framework we used to design the components and the pages along with HTML.

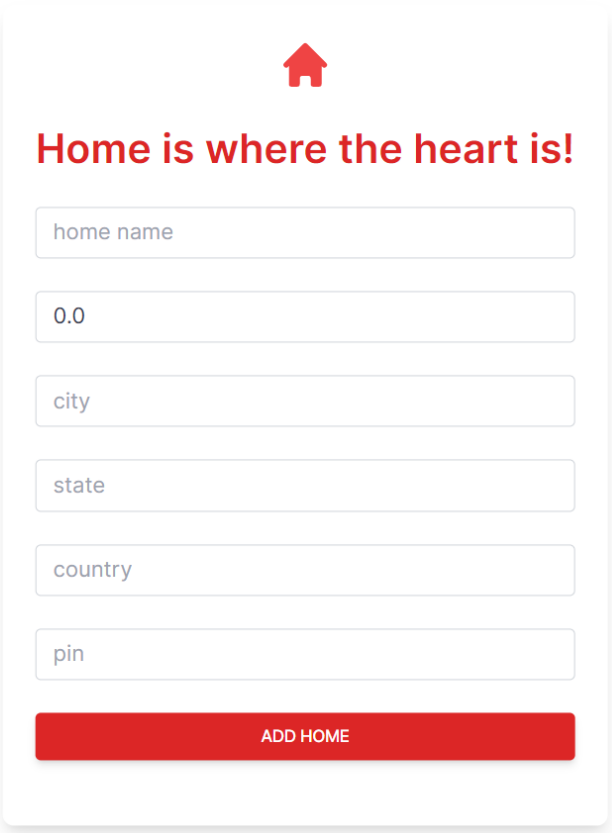
4. Design Principles and Design Patterns Applied

- Singleton - Springboot automatically creates a single object of a class and injects it into the required classes
- Chain of Responsibility - The request is first handled by the controller, it is then passed on to the respective service which is then passed on to the Repositories for the respective database operation
- Single Responsibility Principle - Tenant and the owner responsibility has been separated in Controller, services, and repositories
- Liskov substitution principle - Owner and tenant have few similar functionalities but they aren't inherited from the same parent class
- Interface segregation principle - The interfaces for the owner and tenant service are segregated

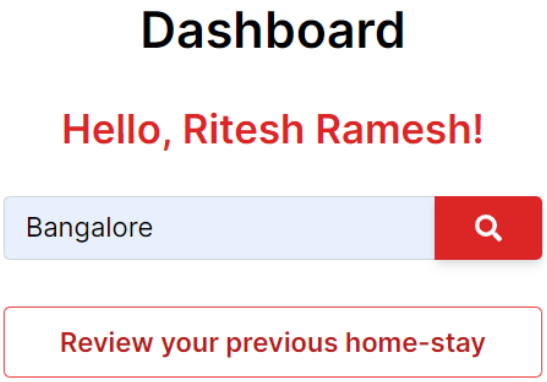
5. Application Screenshots (3-4 important pages)



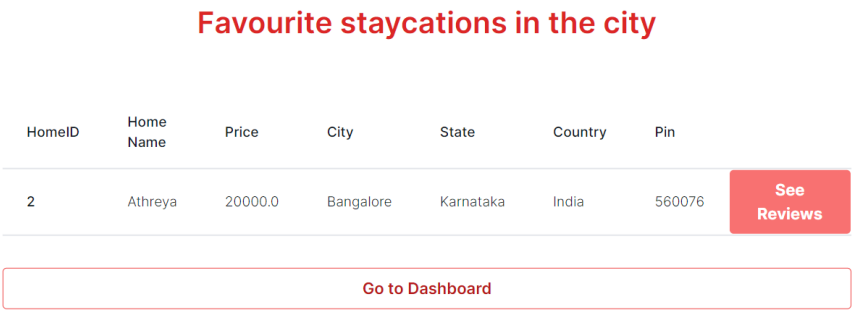
Login for tenant and owner



Forms to add home, review owner, and review tenant



Tenant dashboard with a search bar for searching for homestays by city



Results for searching for homestays by city

6. Team member contributions

S Ritesh Kumar - Springboot Setup, Database setup, Tenant login/signup, Owner login/signup

Sachin Shenoy - Add a Home review, Add tenant Review

Sahana Ramesh - Add Home, Search rooms, View reviews, CSS designs