**Project Name: Roomies**

**Project Member:**

**Ajinkya Bhosale 220343120005**

**Satnam Singh Veerak 220343120113**

**Kishor Janjal 220343120040**

**Snehal Deshmukh 220343120104**

**Abstract:**

As the name specifies “Roomie” is a software developed for Finding Flats, Rooms, and Hostels online. In terms of every service, there are two parts, One provides the service and one receives it. Likewise in our Service, there are also two sides. In various countries, it's very expensive to have a flat alone. In this case, the age group 18-30 has faced the most problems because of its expense.

So this is a service where anyone can pick and share their preferred place with someone you like within your budget. The plus part is to find someone trustworthy for this accommodation service. This project tries to explore how we can help the user with easy navigation with clean interfaces and experiences as usual.

This project deals with developing a Web App for online room search. It provides the user with an option of different categories of users available for registering on the website. The system is implemented using a 3-tier approach, with a backend database, a middle tier of Express Node.js, and React as the front-end.

In order to develop a WebApp, a number of Technologies must be studied and understood. These include multi-tiered architecture, server, and client-side scripting techniques, and implementation technologies such as Express Node.js, React, and relational databases (such as MySQL).

This is a project with the objective to develop a website where the user will be given the advantage to chat with the owner. Thus, resulting in better communication and trust between them.

**Implementation Technologies:**

1. **React:**

A JavaScript library for building user interfaces.

* Developed by Facebook
* React is a view layer library, not a framework like Backbone, Angular, etc.
* You can't use React to build a fully-functional web app

**1.1 Features of React:**

**1. Easy to understand what a component will render**

* Declarative code → predictable code
* You don't really need to study JS in the view file in order to understand what the file does

**2. Easy to mix HTML and JS**

* You do it already with template libraries (e.g. Handlebars, Underscore, etc.)

**3. Uses the full power of JS**

* Decoupling templates from logic does not rely on the templates’ primitive abstractions but uses the full power of JavaScript in displaying views

**4. No complex two-way data flow**

* Uses simple one-way reactive data flow
* Easier to understand than two-way binding
* Uses less code

**5. React is fast!**

* Real DOM is slow
* JavaScript is fast
* Using virtual DOM objects enables fast batch updates to real DOM, with great productivity gains over frequent cascading updates of DOM tree

**6. Flexible**

* React approaches differently by breaking them into components while building user interfaces. This is incredibly important in large applications.

**7. Reusable UI components**

React improves development and debugging processes.

**1.2 Advantages of React:**

* React.js creates its own virtual DOM. This will improve app’s performance since JavaScript virtual DOM is faster than the regular DOM.
* ReactJS helps to create awesome UI.
* ReactJS is SEO-friendly.
* Component and Data patterns improve readability which helps to maintain larger apps.
* React can be used with other frameworks.
* React simplifies the complete process of the scripting environment.
* It facilitates advanced maintenance and increases productivity.
* Ensures faster rendering
* The best thing about React is the provision of a script for mobile app development.
* A strong community backs ReactJS.
* React JS comes with a helpful developer toolset
* Both start-ups and fortune 500 companies use React.

1. **Express**

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for the web.

**2.1 Features of Express:**

1. **Fast Server-Side Development**

The [features of node js](https://www.simplilearn.com/nodejs-for-beginners-article) help express saving a lot of time.

1. **Middleware**

Middleware is a request handler that has access to the application's request-response cycle.

1. **Routing**

It refers to how an application's endpoint's URLs respond to client requests**.**

1. **Templating**

It provides templating engines to build dynamic content on the web pages by [creating HTML templates](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html) on the server.

1. **Debugging**

Express makes it easier as it identifies the exact part where bugs are.

**2.2 Advantages of React:**

* Express is Unopinionated, and we can customize it.
* For request handling, we can use Middleware.
* A single language is used for [frontend and backend development.](https://www.simplilearn.com/full-stack-vs-front-end-vs-back-end-developers-article)
* Express is fast to link it with databases like MySQL, MongoDB, etc.
* ​​Express allows dynamic rendering of HTML Pages based on passing arguments to templates.

**3.Node Js:**

Node.js is a JavaScript runtime environment that is majorly used for the backend development of web projects. It allows the developer to create robust and efficient server-side as well as networking web applications.

**3.1 Features of Node Js:**

* **Asynchronous and Event Driven**

All APIs of the Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

* **Very Fast**

Being built on Google Chrome's V8 JavaScript Engine, the Node.js library is very fast in code execution.Node.js makes use of the V8 JavaScript Runtime motor, which is also used by Google Chrome. Hub provides a wrapper for the JavaScript motor, which makes the runtime motor faster. As a result, the preparation of requests inside Node.js becomes faster as well.

* **Single-Threaded**

Node.js employs a single-threaded architecture with event looping, making it very scalable. In contrast to typical servers, which create limited threads to process requests, the event mechanism allows the node.js server to reply in a non-blocking manner and makes it more scalable. When compared to traditional servers like Apache HTTP Server, Node.js uses a single-threaded program that can handle a considerably larger number of requests.

**3.2 Advantages of Node js:**

* High-performance for Real-time Applications
* Easy Scalability for Modern Applications
* Cost-effective with Fullstack JS
* Community Support to Simplify Development
* Easy to Learn and Quick to Adapt
* Helps in building Cross-functional Teams
* Improves App Response Time and Boosts Performance
* Reduces the Time-to-Market of your applications
* Extensibility to Meet Customized Requirements
* Reduces Loading Time by Quick Caching
* Helps in Building Cross-Platform Applications

**4.** **MySQL**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**4.1 Features of MySQL:**

* **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

* **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

* **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

* **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

* **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

1. **Hardware and Software Requirements (Minimum):**

**Hardware:**   
 1. Intel i3 processor 3rd generation or later / AMD Ryzen 200 2nd generation or later

2. 2 GB ddr3 ram.

3. Windows 8 Home edition or later.

4. 200 GB Sata HDD Space

5. Data Connection 200 kbps

**Software:**

1. Node.js
2. Express
3. React.js
4. MySQL
5. Windows 10
6. **ER Diagram:**

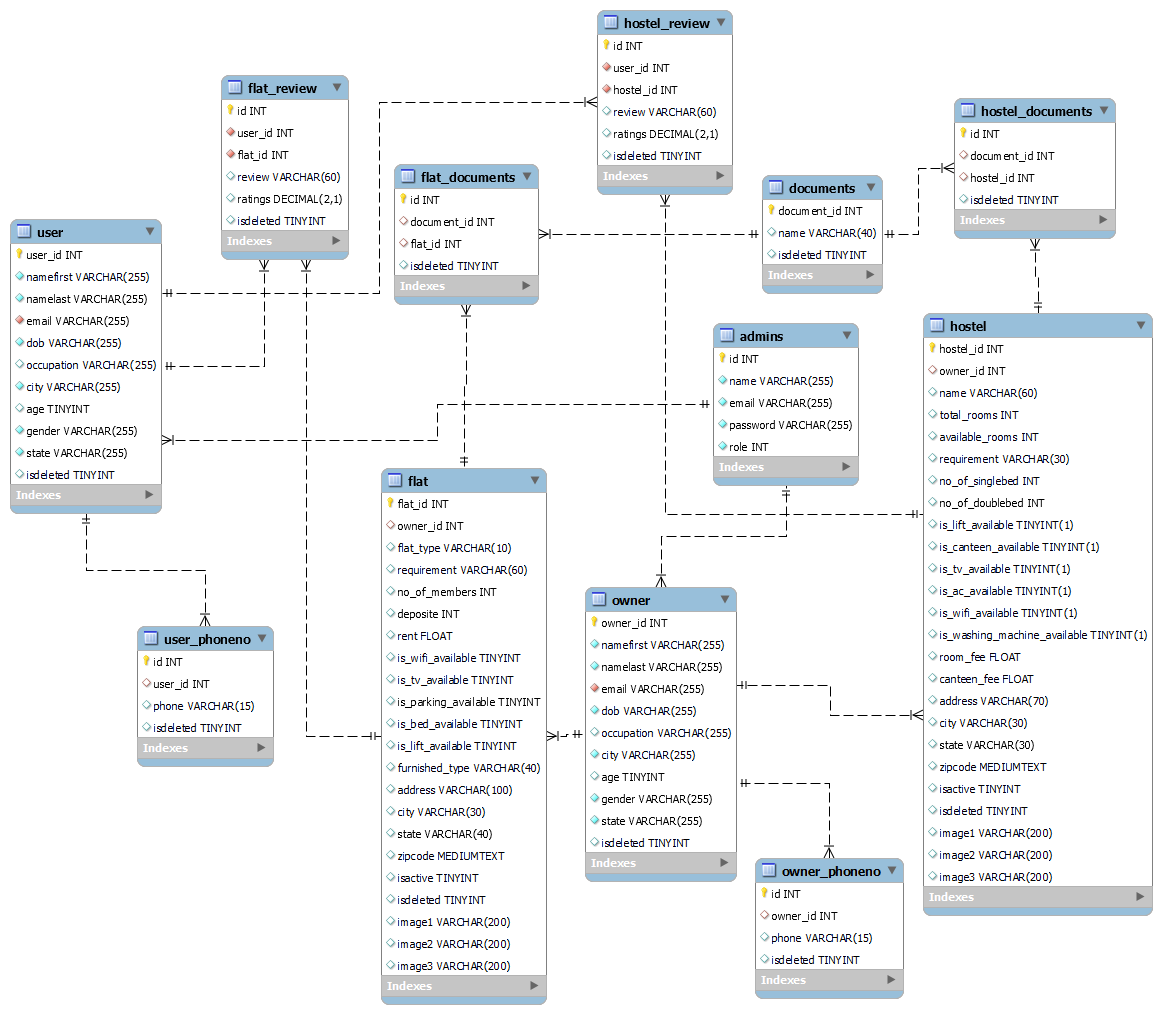
****

Figure 1: ER Diagram

1. **Table Structures:**
2. **Table name: Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| name | varchar(255) | NO |  | NULL |  |
| email | varchar(255) | NO | UNI | NULL |  |
| password | varchar(255) | NO |  | NULL |  |
| role | int | NO |  | 4 |  |

1. **User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| user\_id | int | NO | PRI | NULL |  |
| namefirst | varchar(255) | NO |  | NULL |  |
| namelast | varchar(255) | NO |  | NULL |  |
| email | varchar(255) | NO | UNI | NULL |  |
| dob | varchar(255) | NO |  | NULL |  |
| occupation | varchar(255) | YES |  | NULL |  |
| city | varchar(255) | NO |  | NULL |  |
| age | tinyint | YES |  | NULL |  |
| gender | varchar(255) | NO |  | NULL |  |
| state | varchar(255) | NO |  | NULL |  |
| isdeleted | tinyint | YES |  | 0 |  |

**3. Owner**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| owner\_id | int | NO | PRI | NULL |  |
| namefirst | varchar(255) | NO |  | NULL |  |
| namelast | varchar(255) | NO |  | NULL |  |
| email | varchar(255) | NO | UNI | NULL |  |
| dob | varchar(255) | NO |  | NULL |  |
| occupation | varchar(255) | YES |  | NULL |  |
| city | varchar(255) | NO |  | NULL |  |
| age | tinyint | YES |  | NULL |  |
| gender | varchar(255) | NO |  | NULL |  |
| state | varchar(255) | NO |  | NULL |  |
| isdeleted | tinyint | YES |  | 0 |  |

**4.User\_Phone**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| user\_id | int | YES | MUL | NULL |  |
| phone | varchar(15) | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

**5. Owner\_Phone**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| owner\_id | int | YES | MUL | NULL |  |
| phone | varchar(15) | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

1. **Hostel**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| hostel\_id | int | NO | PRI | NULL | auto\_increment |
| owner\_id | int | YES | MUL | NULL |  |
| name | varchar(60) | YES |  | NULL |  |
| total\_rooms | int | YES |  | NULL |  |
| available\_rooms | int | YES |  | NULL |  |
| requirement | varchar(30) | YES |  | NULL |  |
| no\_of\_singlebed | int | YES |  | NULL |  |
| no\_of\_doublebed | int | YES |  | NULL |  |
| is\_lift\_available | tinyint(1) | YES |  | NULL |  |
| is\_canteen\_available | tinyint(1) | YES |  | NULL |  |
| is\_tv\_available | tinyint(1) | YES |  | NULL |  |
| is\_ac\_available | tinyint(1) | YES |  | NULL |  |
| is\_wifi\_available | tinyint(1) | YES |  | NULL |  |
| is\_washing\_machine\_available | tinyint(1) | YES |  | NULL |  |
| room\_fee | float | YES |  | NULL |  |
| canteen\_fee | float | YES |  | NULL |  |
| address | varchar(70) | YES |  | NULL |  |
| city | varchar(30) | YES |  | NULL |  |
| state | varchar(30) | YES |  | NULL |  |
| zipcode | mediumtext | YES |  | NULL |  |
| isactive | tinyint | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |
| image1 | varchar(200) | YES |  | NULL |  |
| image2 | varchar(200) | YES |  | NULL |  |
| image3 | varchar(200) | YES |  | NULL |  |

1. **Hostel Documents**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| document\_id | int | YES | MUL | NULL |  |
| hostel\_id | int | YES | MUL | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

1. **Hostel Review**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| user\_id | int | NO | MUL | NULL |  |
| hostel\_id | int | NO | MUL | NULL |  |
| review | varchar(60) | YES |  | NULL |  |
| ratings | decimal(2,1) | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

1. **Flat**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| flat\_id | int | NO | PRI | NULL | auto\_increment |
| owner\_id | int | YES | MUL | NULL |  |
| flat\_type | varchar(10) | YES |  | NULL |  |
| requirement | varchar(60) | YES |  | NULL |  |
| no\_of\_members | int | YES |  | NULL |  |
| deposite | int | YES |  | NULL |  |
| rent | float | YES |  | NULL |  |
| is\_wifi\_available | tinyint | YES |  | NULL |  |
| is\_tv\_available | tinyint | YES |  | NULL |  |
| is\_parking\_available | tinyint | YES |  | NULL |  |
| is\_bed\_available | tinyint | YES |  | NULL |  |
| is\_lift\_available | tinyint | YES |  | NULL |  |
| furnished\_type | varchar(40) | YES |  | NULL |  |
| address | varchar(100) | YES |  | NULL |  |
| city | varchar(30) | YES |  | NULL |  |
| state | varchar(40) | YES |  | NULL |  |
| zipcode | mediumtext | YES |  | NULL |  |
| isactive | tinyint | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |
| image1 | varchar(200) | YES |  | NULL |  |
| image2 | varchar(200) | YES |  | NULL |  |
| image3 | varchar(200) | YES |  | NULL |  |

1. **Flat Documents**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| document\_id | int | YES | MUL | NULL |  |
| flat\_id | int | YES | MUL | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

1. **Flat Review**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| id | int | NO | PRI | NULL | auto\_increment |
| user\_id | int | NO | MUL | NULL |  |
| flat\_id | int | NO | MUL | NULL |  |
| review | varchar(60) | YES |  | NULL |  |
| ratings | decimal(2,1) | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

1. **Document**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| document\_id | int | NO | PRI | NULL | auto\_increment |
| name | varchar(40) | YES |  | NULL |  |
| isdeleted | tinyint | YES |  | NULL |  |

1. **UML Diagrams:**

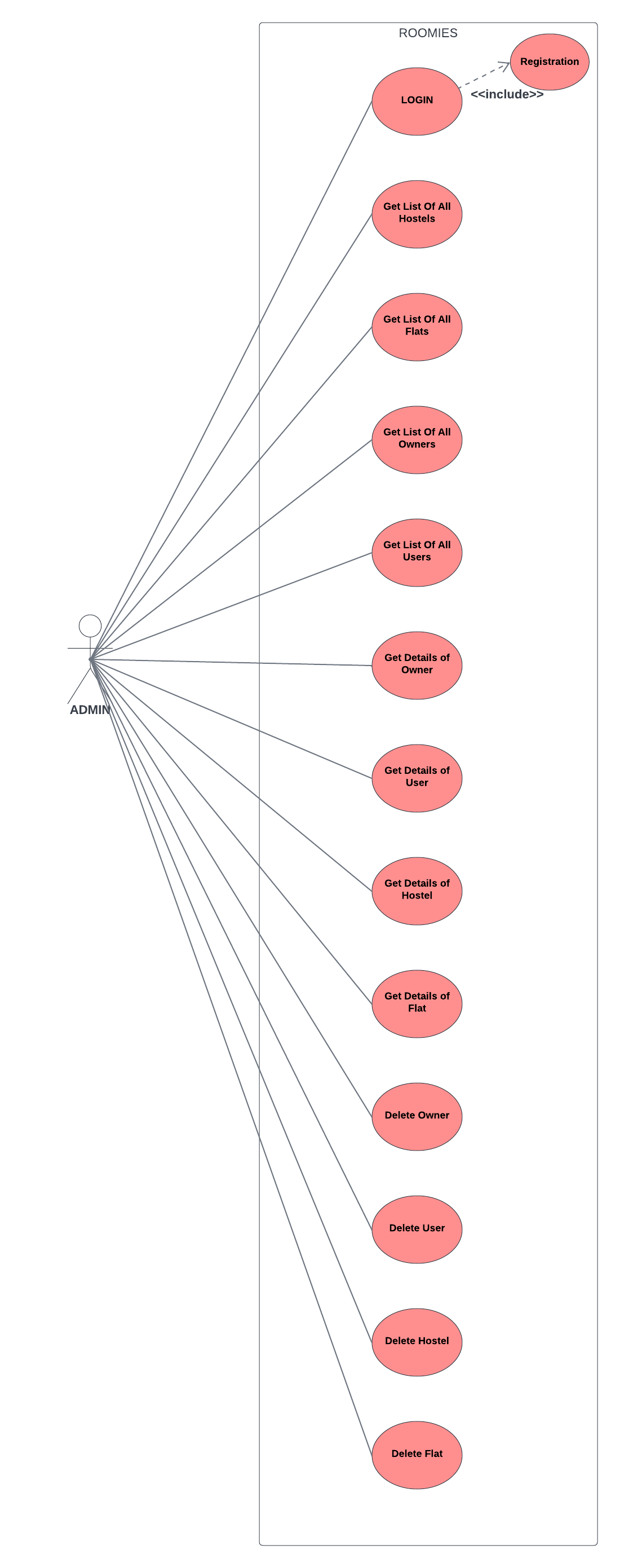


Figure 2: Admin Use Case

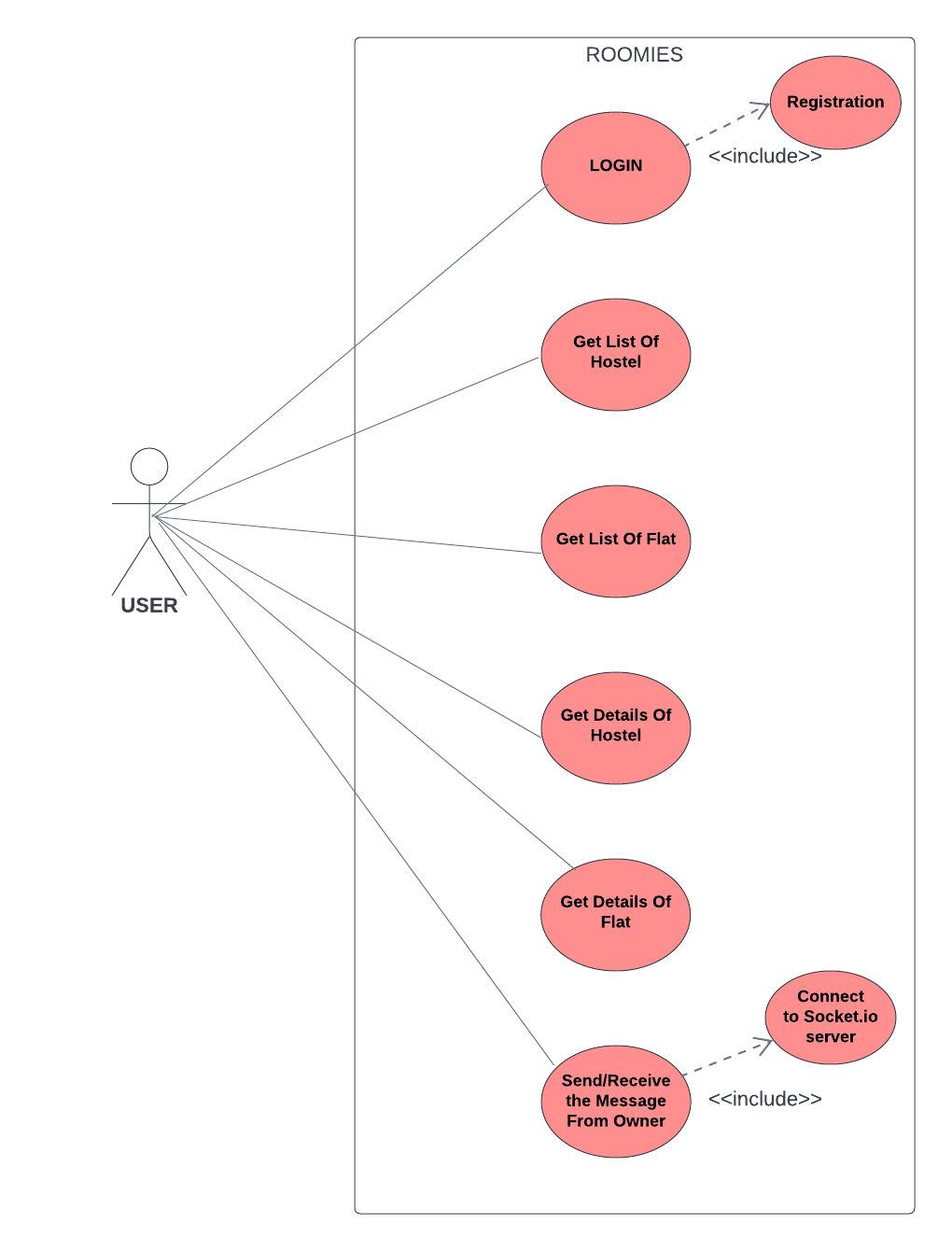


Figure 3: User Use Case

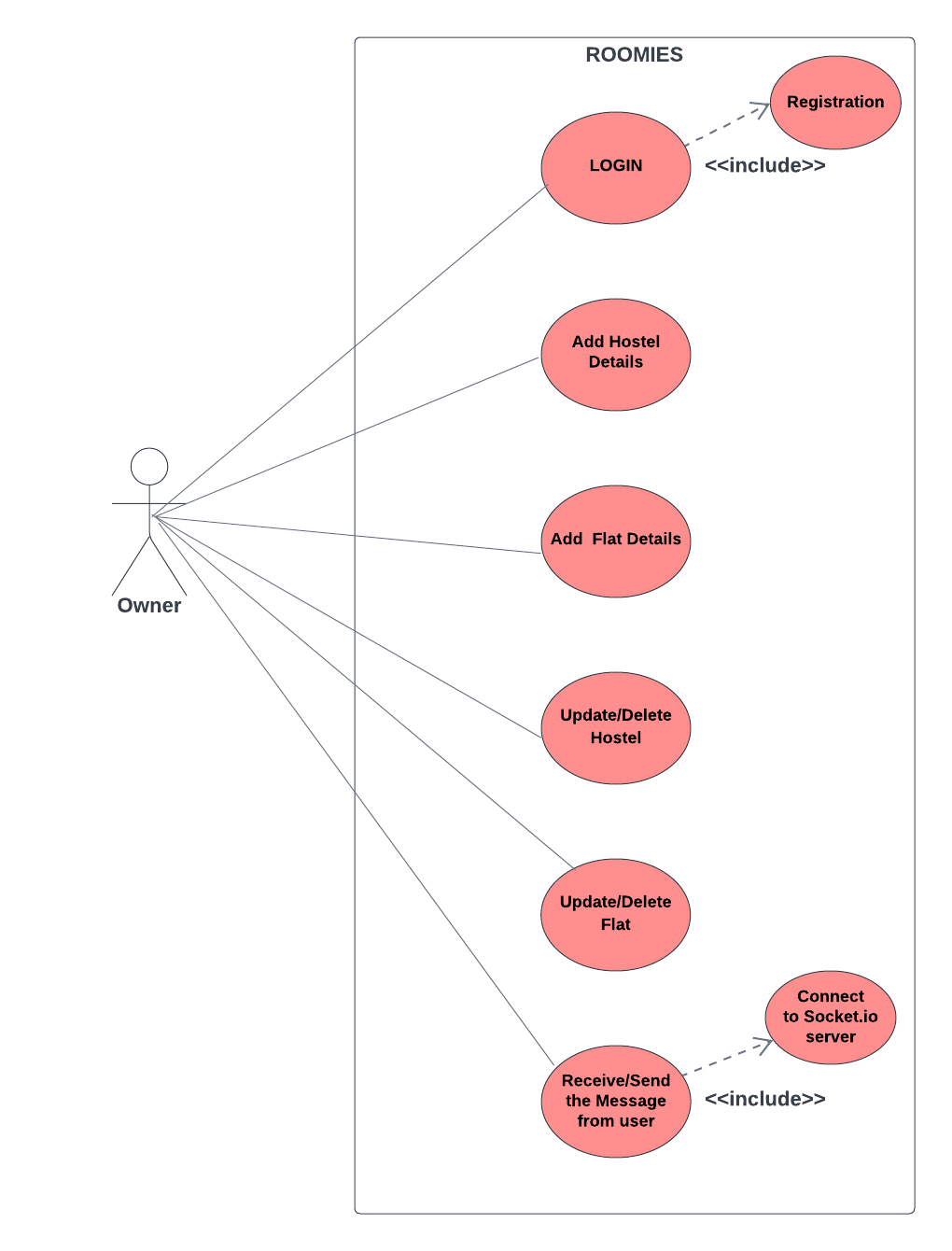


Figure 4: Owner Use Case

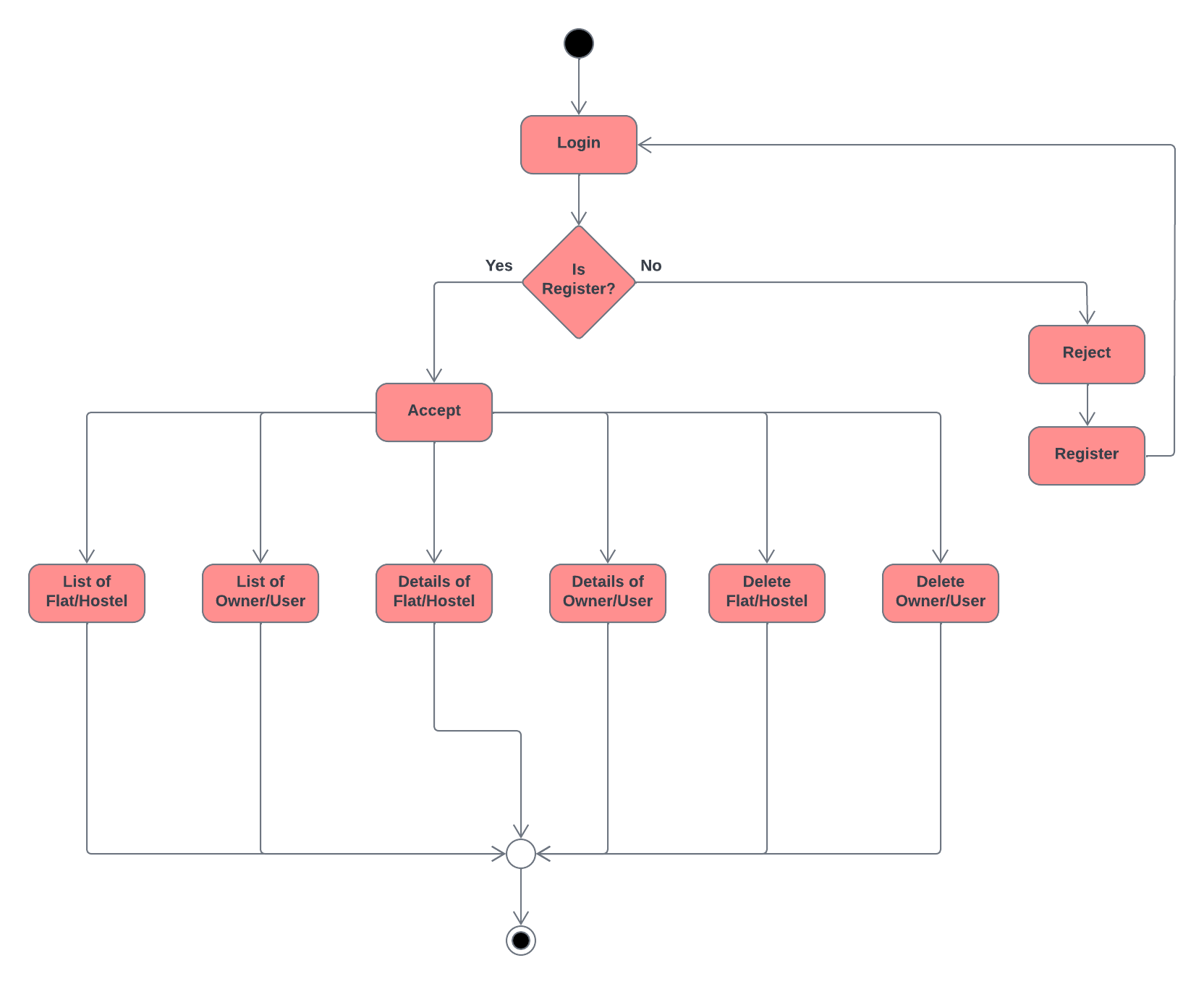


Figure 5: Admin Activity Diagram

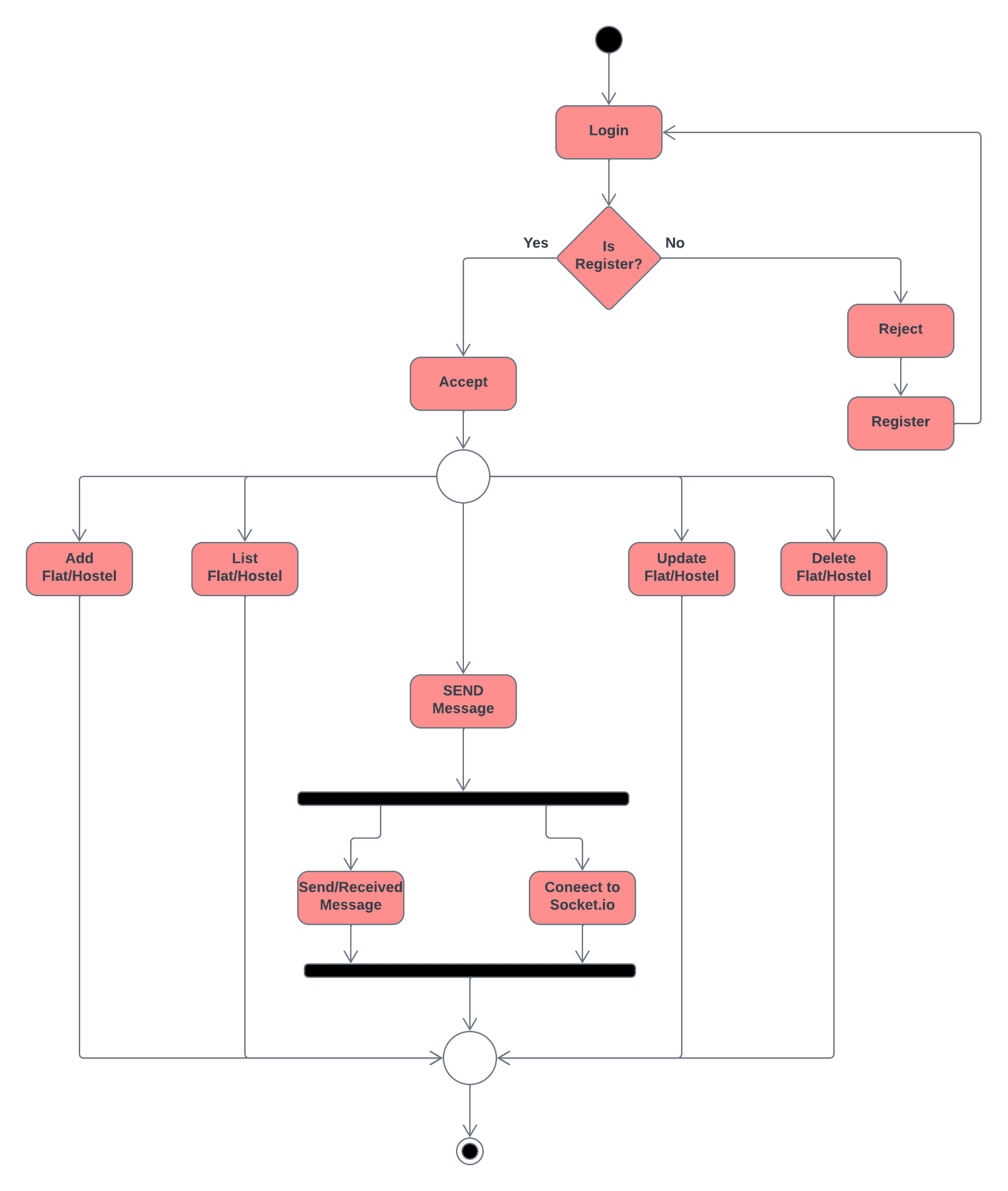


Figure 6: Owner Activity Diagram

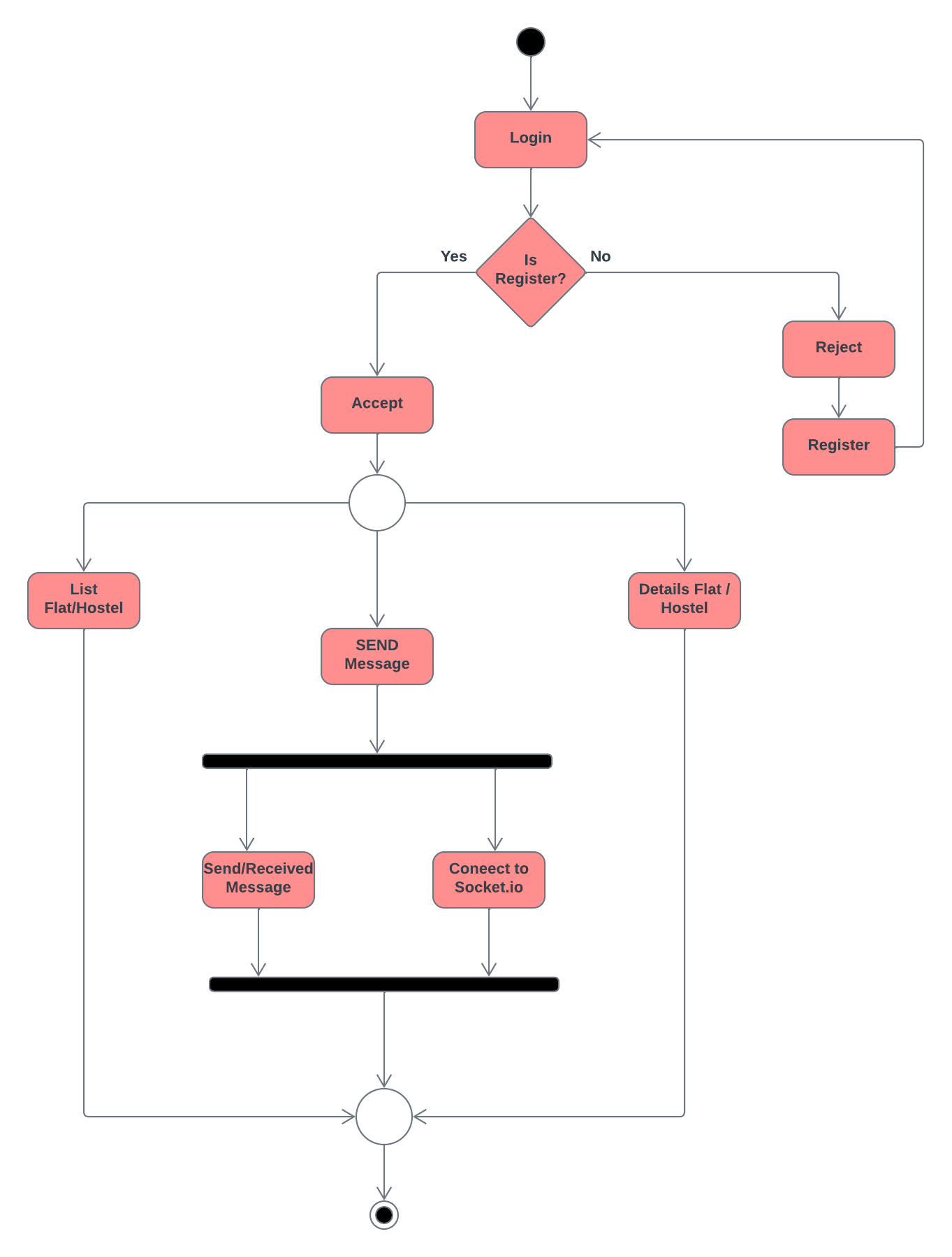


Figure 7: User Activity Diagram

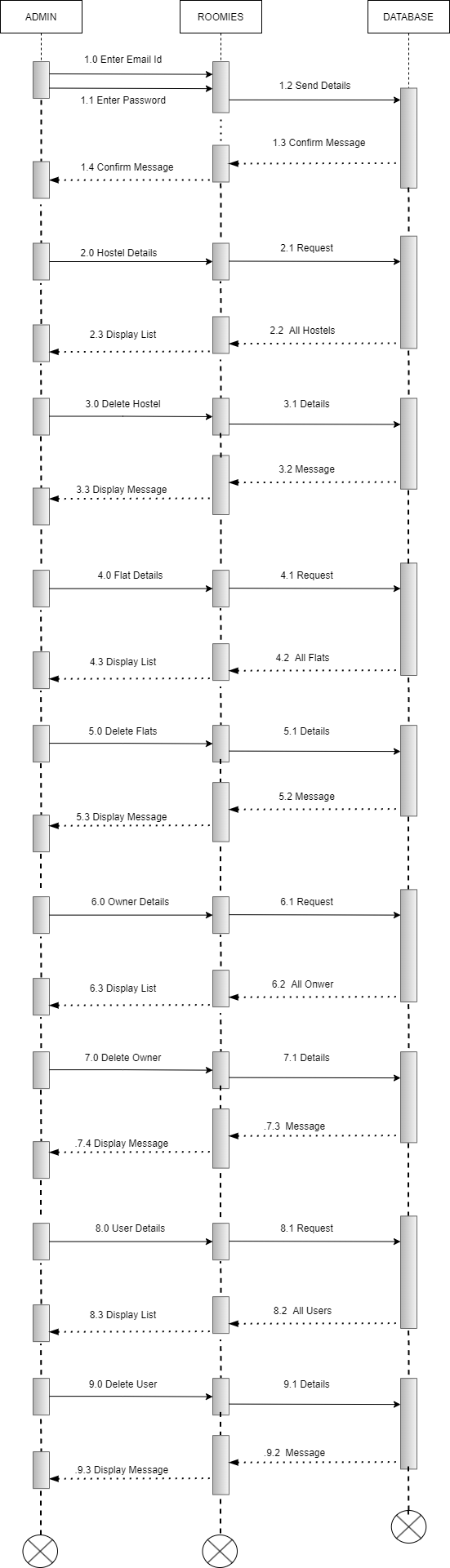


Figure 8: Admin Sequence Diagram

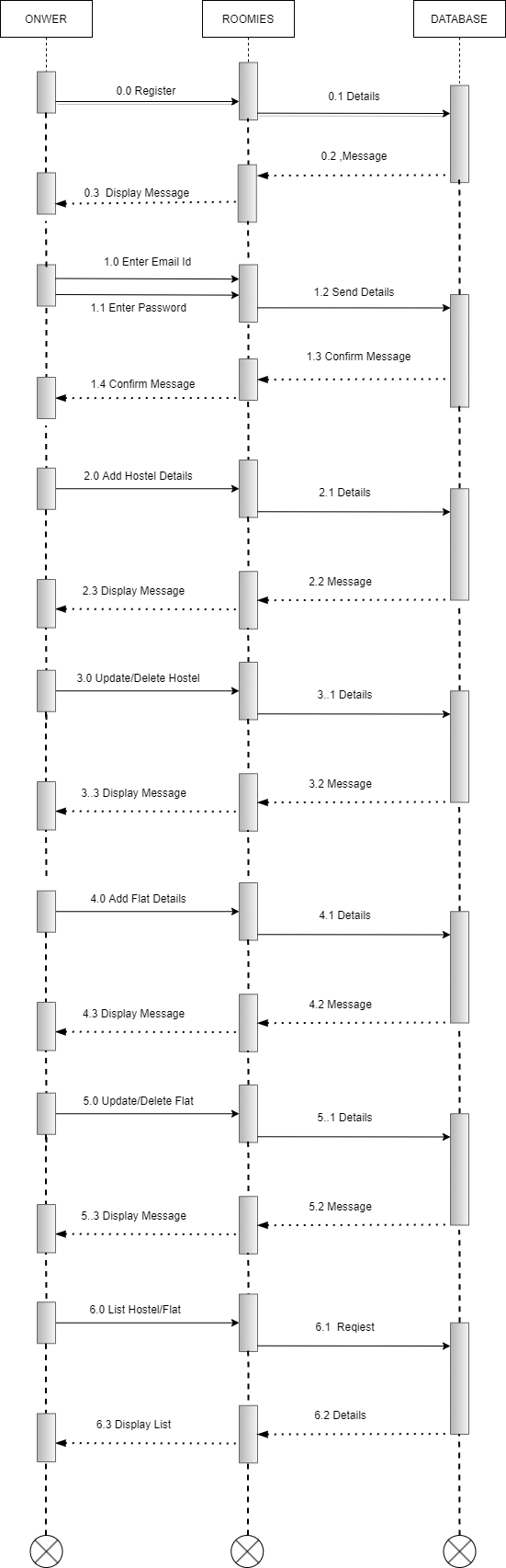


Figure 9 : Owner Sequence Diagram

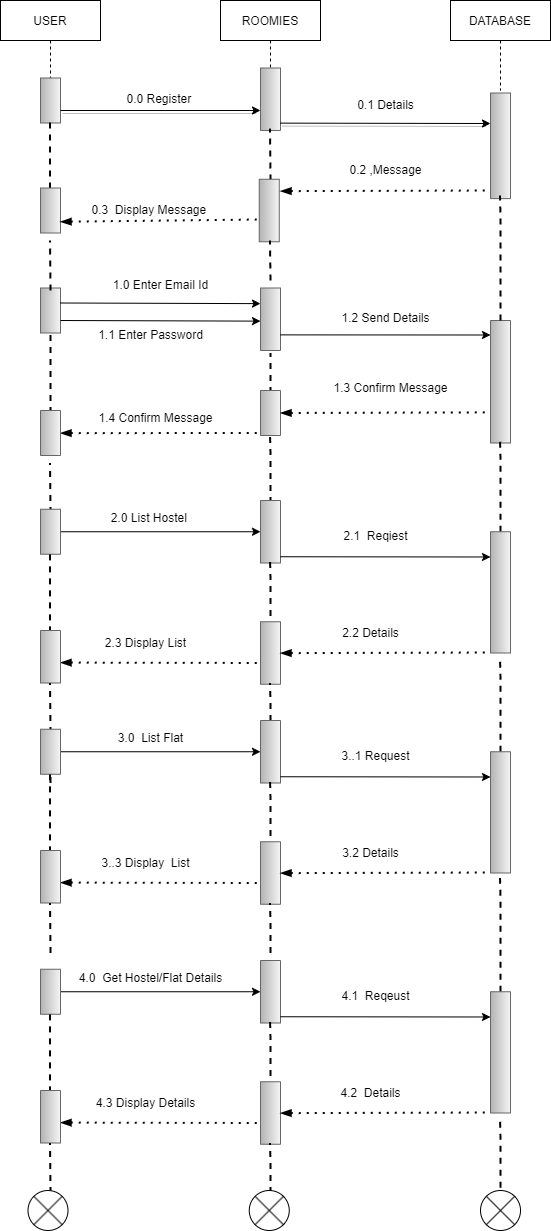


Figure 10: User Sequence Diagram

1. **End-to-End Flow of Application:**
   * User will log in to the portal or will have to register if he is not a registered user.
   * After registration User will log in and a profile page will be displayed to him/her which will display the details.

**Customer :**

* 1. Customer on their homepage can see the city name with image where they can click the city of their preference for finding hostels/flats.
  2. From the navigation bar, the User can click on the ‘hostel/flat tab**’** button and reach the hostel/flat list.
  3. Further after the user clicks on the ‘details button’ complete details of the hostel/flat in elegant UI .
  4. Also by clicking the **chat** option customer can have one-to-one interaction with the owner.

**Owner:**

1. Owners can list required information of hostels/flats as per requirement.
2. Owners can also update details of hostels/flats uploaded by themselves whenever they want.
3. They can read messages sent by customers and can reply to them accordingly.

**Admin :**

1. Admin can see and update details of both owners as well as the customer.
2. Admin can delete flat and the hostel also can see details of them.
3. **Future Scope of Project**

* Strict Verification of Users to avoid Fraud.
* Improvement in design.
* Mobile Application.
* Including Real Estate Module.
* Tie-up with Packers & Movers.

**Thank You!**