

# **RENTEASY**

## **A PROJECT REPORT**

*Submitted by*

**KESHVI PATEL (20BECE30161)**

**TEJAS PATEL (20BECE30188)**

**TIRTH PATEL (20BECE30190)**

**VISHAKHA RAJPUT (20BECE30219)**

*In fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**Computer Engineering**



**LDRP Institute of Technology and Research, Gandhinagar**

**Kadi Sarva Vishwavidyalaya**

**2023-2024**

***LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH***  
***GANDHINAGAR***

**CE-IT Department**



**CERTIFICATE**

This is to certify that the Project Work entitled “**RentEasy**” has been carried out by **Keshvi Patel** (**20BECE30161**) under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-7 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel

**Internal Guide**

**LDRP ITR**

Dr. Sandip Modha

**Head of the Department**

**LDRP ITR**

***LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH***  
***GANDHINAGAR***

**CE-IT Department**



**CERTIFICATE**

This is to certify that the Project Work entitled **“RentEasy”** has been carried out by **Tejas Patel** (**20BECE30188**) under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-7 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel

**Internal Guide**

**LDRP ITR**

Dr. Sandip Modha

**Head of the Department**

**LDRP ITR**

***LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH***  
***GANDHINAGAR***

**CE-IT Department**



**CERTIFICATE**

This is to certify that the Project Work entitled **“RentEasy”** has been carried out by **Tirth Patel** (**20BECE30190**) under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-7 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel

**Internal Guide**

**LDRP ITR**

Dr. Sandip Modha

**Head of the Department**

**LDRP ITR**

***LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH***  
***GANDHINAGAR***

CE-IT Department



**CERTIFICATE**

This is to certify that the Project Work entitled **“RentEasy”** has been carried out by **Vishakha Rajput (20BECE30219)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-7 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel

**Internal Guide**

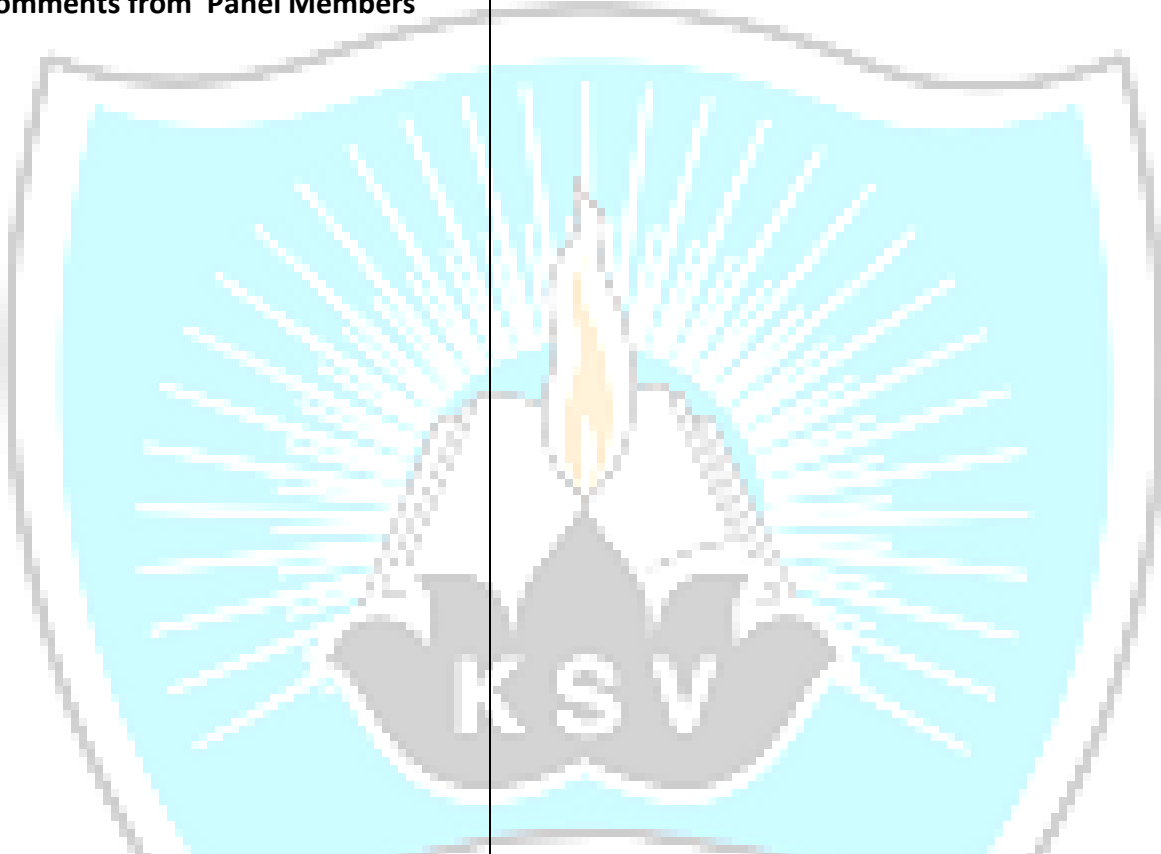
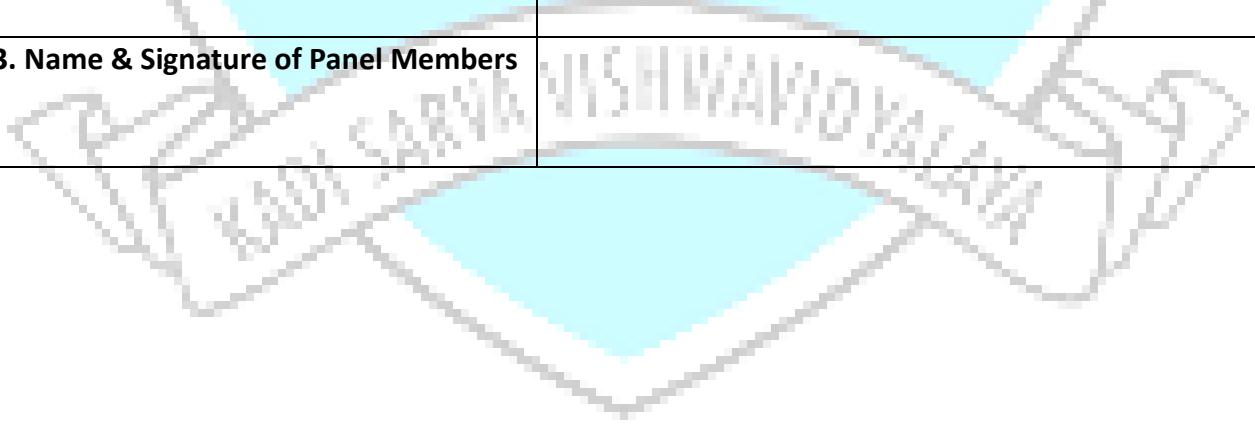
**LDRP ITR**

Dr. Sandip Modha

**Head of the Department**

**LDRP ITR**

## Presentation-I for Project-II

1. Name & Signature of Internal Guide	
2. Comments from Panel Members	
3. Name & Signature of Panel Members	

# ACKNOWLEDGEMENT

With immense pleasure we would like to present this report on our topic **RentEasy**. I am thankful to all that have helped us a lot for successful completion of my project and providing us courage for completing the work.

I thankful to my Head of the Department **Dr. Sandip Modha**, my internal faculty guide **Prof. Hitesh Patel**, for providing guidance throughout my work giving us their valuable time.

Finally, I would like to thank my parents and friends. who have directly or indirectly helped me in making the project work successfully

With Regards,

Keshvi Patel (20BECE30161)

Tejas Patel (20BECE30188)

Tirth Patel (20BECE30190)

Vishakha Rajput (20BECE30219)

# ABSTRACT

The **RentEasy** is a comprehensive property booking application designed to simplify the process of creating, managing, and booking properties. This user-friendly program enables users to generate renting properties details effortlessly and store them securely in a centralized database. The application offers seamless functionality for booking properties from various places, enabling easy booking and canceling.

RentEasy provides a streamlined interface for available properties and booking their properties. Users can conveniently book properties as availability of it. The program ensures data integrity and security, ensuring that all information of user is safely stored and easily accessible when needed.

Key features of RentEasy include a user-friendly property adding tool, a centralized property storage system, booking status checking, and flexible property searching options. This application is an ideal solution of renting properties for individuals seeking an efficient and intuitive platform to book their property needs.

With RentEasy, users can save valuable time and effort by property creation process, maintaining a digital record of properties, and easily search available property. Whether it's adding property, organizing them, or monitoring their availability status, RentEasy offers a comprehensive and user-friendly solution to simplify the renting process and enhance productivity.



# **TABLE OF CONTENT**

<b>Acknowledgement</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Table of content</b>	<b>iii</b>
<b>Table of figures</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction	2
1.2 Scope	2
1.3 Project Summary	2
1.4 Overview	3
1.5 Problem	3
<b>2 Technology</b>	<b>4</b>
2.1 Tools & Technology	5
2.2 Brief of work done	5
<b>3 System Requirements</b>	<b>6</b>
3.1 User characteristics	7
3.2 Hardware & Software Requirements	7
3.3 Constraints	8
3.4 Assumption and dependencies	15
<b>4 System Design</b>	<b>16</b>
4.1 Study of current system	17
4.2 Problems and weakness	17
4.3 Requirements of new system	18
4.4 Feasibility study	19
4.5 Requirements validation	20

4.6	Activity in new system	21
4.7	Features of new system	22
4.8	Class diagram	23
4.9	System activity	24
4.10	Object interaction	25
4.11	Sequence and collaboration diagram	26
4.12	ER diagram	28
4.13	Activity diagram	29
<b>5</b>	<b>System Design</b>	<b>31</b>
5.1	System application design	32
5.2	Database design	34
5.3	Input/output & Interface design	35
<b>6</b>	<b>Test Cases</b>	<b>41</b>
<b>7</b>	<b>Conclusion</b>	<b>47</b>
<b>8</b>	<b>Bibliography</b>	<b>49</b>

## **TABLE OF FIGURES**

<b>Figure no.</b>	<b>Figure Name</b>	<b>Page no.</b>
4.8	Class diagram	23
4.9	System activity diagram	24
4.11.1	DFD level-0 diagram	26
4.11.2	DFD level-0 diagram	27
4.12	ER Diagram	28
4.13	Activity diagram	29

## **Chapter1-Introduction**

### **1.1. Introduction**

### **1.2. Scope**

### **1.3. Project Summary and Purpose**

### **1.4. Overview of the Project**

### **1.5. Problem Definition**

## **1.1. Introduction**

RentEasy is a modern web application designed to revolutionize the property rental and booking process. Catering to small businesses and individuals seeking accommodations, RentEasy offers a user-friendly platform with seamless functionalities.

By providing easy property listing, efficient booking, and personalized customization options, RentEasy simplifies the rental management experience, making it a cost-effective and hassle-free solution for all. Say goodbye to the burdensome process of property management and welcome a streamlined approach with RentEasy.

## **1.2. Scope**

RentEasy aims to provide a comprehensive solution for property rental and booking management. The application's scope includes efficient property listing and management features, allowing property owners to showcase their properties with detailed information and images. Users can seamlessly book rental accommodations based on location, guest count, and room count, while benefiting from personalized customization options.

RentEasy offers secure user authentication, image and media management, favorites and bookings tracking, location-based services, and a responsive design for a smooth user experience. Prisma ORM and MongoDB handle reliable database management, ensuring data integrity and efficiency throughout the platform.

## **1.3. Project Summary and Purpose**

RentEasy is a cutting-edge web application designed to simplify the property rental and booking process. It provides a user-friendly platform for property owners to list and manage their properties effectively, while offering individuals seeking accommodations a seamless booking experience with personalized options.

RentEasy aims to revolutionize the way users find, list, and manage rental properties. The purpose of RentEasy is to streamline the rental management experience, making it convenient, cost-effective, and hassle-free for both property owners and seekers.

## **1.4. Overview of the Project**

RentEasy is a modern web application that simplifies the property rental and booking process for property owners and individuals seeking accommodations. With a user-friendly interface and seamless functionalities, RentEasy caters to small businesses and home industries with limited resources. It offers effortless property listing and management, enabling property owners to showcase their properties effectively.

Users can easily browse and book rental accommodations, benefiting from personalized customization options. RentEasy ensures secure user authentication and provides location-based services for property visualization. The application aims to streamline the rental management experience, offering a cost-effective and hassle-free solution for property owners and seekers.

## **1.5. Problem Definition**

The property rental and booking process poses challenges for property owners and seekers, with existing solutions often being costly and lacking customization options. Small businesses and home industries with limited resources find it difficult to effectively manage their properties, while individuals struggle to find suitable rental accommodations.

The absence of a user-friendly platform for seamless property listing, booking, and secure authentication further compounds these issues. RentEasy aims to address these problems by offering a comprehensive and affordable solution, simplifying the rental management experience for property owners and providing a hassle-free way for seekers to find their ideal rentals.

## **Chapter2-Technology and Literature Review**

### **2.1. About Tools and Technology**

### **2.2. Brief History of Work Done**

## **2.1. About Tools and Technology:**

- ▶ **Operating System:** Any version of Windows family(4.0&above)
- ▶ **Frontend:** NextJs, React, tailwind, Typescript, NextAuth
- ▶ **Orm:** Prisma
- ▶ **Backend:** MongoDB, Cloudinary (for image storage)

## **2.2. Brief History of Work Done**

- ▶ **Planning of the application:** This includes determining the features that the application will need, as well as the user interface and data model. It will be answering the questions like what type of features will the application need. What can be the user interface look like? What data will the application need to store? How will the application work?
- ▶ **Designing a user interface:** Once planning of the application is done, we need to design the user interface. This involves creating the NextJs, Tailwind CSS and Typescript that will be used to display the application's interface. The user interface should be easy to use and navigate. It should also be visually appealing.
- ▶ **Developing the backend logic:** The backend logic is the code that will be used to handle the application's business logic. This includes code for generating invoices, sending invoices to backend, and storing the details of invoices. Along with it, storing the user's profile information and providing authentication support to the application.
- ▶ **Testing the application:** Once the application is developed, we need to test it to ensure that it works correctly. This involves testing the application with different scenarios. We need to also test the application for security vulnerabilities.
- ▶ **Deploying the application:** Once the application is tested, we need to deploy it so that users can access it. It involves making the application available on a web server. We have to make sure that the application is secure and that it can handle a high volume of traffic.



## **Chapter 3 – System Requirements Study**

### **3.1. User Characteristics**

### **3.2. Hardware and Software Requirements**

### **3.3. Constraints**

#### **3.3.1. Regulatory Policies**

#### **3.3.2. Hardware Limitations**

#### **3.3.3. Interfaces to Other Applications**

#### **3.3.4. Parallel Operations**

#### **3.3.5. Higher Order Language Requirements**

#### **3.3.6. Reliability Requirements**

#### **3.3.7. Criticality of the Application**

#### **3.3.8. Safety and Security Consideration**

### **3.4. Assumptions and Dependencies**

### **3.1. User Characteristics**

Analyzing user characteristics is an important aspect of any project. It allows us to clearly define and focus on who the end users are for the project. Also, it allows checking the progress of the project to ensure that we are still developing the system for the end users. The user must have following characteristics:

- User must have basic knowledge of computers.
- User should understand the use of all modules.
- User can easily interact with the proposed system.
- User must know the method to upload an image in the app.
- User should be also being aware about the running process of the system.

### **3.2. Hardware and Software Requirements**

Software and hardware requirements are used to describe the minimum hardware and software requirements to run the software. These requirements are described below:



#### **Hardware Requirements**

- **Client:**
  - 256MB of RAM
  - 1.6 GHz CPU
  - Internet Connection
  - Monitor
  - Keyboard/Mouse
  - Printer
  - GPS
- **Server:**
  - 1.6 GHz CPU
  - 1GB of RAM
  - Internet Connection
  - Monitor
  - Keyboard/Mouse
  - Printer

### ► **Software Requirements**

- Client:
  - OperatingSystem:WindowsorLinux,AndroidoriOS
  - Web Browser: any HTMLcompliantbrowser,GoogleChrome (HighlyPreferred)
  - Node.js version 18.3.1
  - Languages: Multiple Languages
  - License: Open Source
  - Packages:React,Next,Node,TailwindCSS
- Server:
  - OperatingSystem:Windows orLinux,Androidor iOS
  - Database: MongoDB
  - Technologies:HTML,TailwindCSS,Typescript

## **3.3. Constraints**

**Performance Constraints:** The app should be able to handle high user loads and provide a responsive user experience, even during peak times. Performance optimization, such as caching, server scaling and efficient database queries should be implemented to ensure smooth and fast operation.

### **3.3.1. Regulatory Policies**

- **Data Security and Privacy:** RentEasy must prioritize data security and protect the privacy of all user and property data. The application must implement strong encryption mechanisms to safeguard customer information and ensure that user data is securely stored and transmitted. RentEasy should have a robust security policy in place to prevent unauthorized access and data breaches. Additionally, the application must comply with relevant data privacy laws, such as the General Data Protection Regulation (GDPR) or any other applicable data protection regulations.
- **Tax Compliance:** RentEasy must adhere to all relevant tax laws and regulations concerning property rentals and bookings. The application should accurately calculate taxes, fees, and other rates applicable to property transactions. It is crucial to remit collected taxes to the appropriate authorities promptly and maintain accurate records of all transactions to ensure proper tax accounting.

- ▶ **Accessibility:** RentEasy must ensure that the web application is accessible to all users, including individuals with disabilities. Implementing accessibility features such as screen reader support, keyboard navigation, and other assistive technologies will ensure that all users can access and use the application effectively. Accessibility measures should be designed to maintain security and protect user data.
- ▶ **Compliance with Laws and Regulations:** RentEasy must adhere to all other applicable laws and regulations, including those related to fraud prevention, financial reporting, and consumer protection. The application should implement measures to prevent fraudulent activities and maintain transparency in financial reporting.
- ▶ **Property Owner Compliance:** RentEasy must ensure that property owners listing their properties on the platform comply with all relevant laws and regulations related to property rentals, including zoning laws, building codes, safety regulations, and any other legal requirements.

### **3.3.2. Hardware limitations**

- ▶ **Processing Power:** RentEasy requires sufficient processing power to handle property listing, booking requests, and data processing. The application's performance depends on the number of property listings, bookings, and user interactions. Adequate processing power is essential to ensure a smooth user experience, especially during peak usage periods.
- ▶ **Storage:** RentEasy needs ample storage to store property information, user data, images, and other media associated with listings and bookings. The amount of storage required will increase with the growth of the platform and the number of properties listed and booked by users.
- ▶ **Memory:** Sufficient memory is crucial for RentEasy to efficiently store and manage property data, user sessions, and application code. With multiple users simultaneously accessing the platform, enough memory is needed to maintain responsive and seamless operations.
- ▶ **Bandwidth:** RentEasy heavily relies on bandwidth for data communication between users and the application server. Efficient bandwidth management is necessary to ensure quick property image and media loading, smooth booking processes, and seamless user interactions.
- ▶ **Scalability:** As RentEasy's user base and property listings increase over time, the application must be designed to scale effectively. Ensuring horizontal scalability will allow the platform to handle growing traffic without compromising performance.

- ▶ **Security:** Given the sensitivity of user data, RentEasy must have robust security measures to protect personal information, payment details, and sensitive property data. This includes using encrypted connections, secure authentication mechanisms, and data encryption for storage.
- ▶ **Mobile Optimization:** RentEasy should be optimized for mobile devices to cater to users accessing the platform through smartphones and tablets. Ensuring a responsive design and efficient mobile data usage will enhance the user experience on smaller screens.

### **3.3.3. Interfaces to Other Applications**

RentEasy, a property rental and booking web application, can have interfaces with other applications to enhance its functionality and provide a more seamless user experience. Some of the interfaces that RentEasy can have with other applications include:

- ▶ **Customer Relationship Management (CRM) Applications:** RentEasy can integrate with CRM applications to store and manage customer data. This data may include user contact information, rental preferences, booking history, and payment details. Integrating with CRM applications can help RentEasy provide personalized rental suggestions, track customer interactions, and improve customer relationship management.
- ▶ **Payment Processing Applications:** RentEasy can establish interfaces with payment processing applications to facilitate secure and efficient online payments for property bookings. Users can make payments using various methods, such as credit/debit cards, online wallets, or other payment gateways. Integrating with payment processing applications ensures a smooth and secure payment experience for users.
- ▶ **Accounting Applications:** RentEasy can connect with accounting applications to track financial data related to property rentals and bookings. This may include income from bookings, expenses related to property management, taxes, and financial reporting. By integrating with accounting applications, RentEasy can generate financial reports, track profitability, and maintain accurate financial records.
- ▶ **ERP Applications:** RentEasy can integrate with Enterprise Resource Planning (ERP) applications to streamline its operations and improve overall efficiency. ERP integration can help centralize and manage data related to property listings, bookings, user information, payments, and financial transactions. This can result in a more organized and cohesive management of all aspects of RentEasy's operations.

By establishing these interfaces with other applications, RentEasy can optimize its

performance, enhance user experience, and provide a comprehensive solution for property rental and booking management. The seamless integration of various applications will contribute to RentEasy's effectiveness in meeting the needs of property owners and seekers, making it a robust and efficient platform in the rental market.

### **3.3.4. Parallel Operation for RentEasy**

RentEasy, a property rental and booking web application, involves several parallel operations that enhance its functionality and provide users with a seamless experience. Some of the parallel operations performed by RentEasy include:

- ▶ **Edit Property Listings:** This feature allows property owners to make changes to their existing property listings, such as updating property details, rental rates, availability dates, or property descriptions. This enables property owners to keep their listings up-to-date and correct any errors or changes in property information.
- ▶ **View Property Details:** This feature enables users to view detailed information about listed properties, including property specifications, amenities, images, and location. Users can access and compare property details simultaneously to make informed decisions while searching for suitable rental accommodations.
- ▶ **Book Properties:** This feature allows users to book rental properties in parallel. Multiple users can initiate bookings for different properties simultaneously, ensuring efficient property allocation and avoiding any potential booking conflicts.
- ▶ **Save Favorites:** RentEasy can enable users to save their favorite properties to their account, allowing them to access and compare their preferred listings from multiple devices.
- ▶ **Mark Bookings:** This feature allows property owners and users to mark bookings as confirmed or reserved, indicating the status of a particular booking. This helps in tracking bookings, managing property availability, and avoiding double bookings.

By performing these parallel operations, RentEasy ensures a smooth and efficient property rental and booking process, meeting the needs of both property owners and seekers while optimizing overall application performance.

### **3.3.5. Higher Order Language Requirements**

Higher-order languages (HLLs) can be leveraged in RentEasy, the property rental and booking web application, to enhance performance, flexibility, and overall functionality. Here are some ways HLLs can be utilized in RentEasy:

► **Create the User Interface:** HLLs can be used to create a dynamic and interactive user interface for RentEasy. With HLLs, developers can design reusable components, implement event handling, and enhance user interactions. This allows users to easily browse properties, book rentals, manage listings, and access their dashboard seamlessly.

► **Generate Property Listings:** HLLs can be employed to generate property listings dynamically. The application can use functions that fetch property information from the database and create property cards with relevant details, such as images, descriptions, and availability dates.

► **Store and Manage Property Data:** HLLs can facilitate efficient data management in RentEasy. Functions can be used to handle database operations, such as creating, reading, updating, and deleting property data. This ensures smooth property management for property owners and accurate information for property seekers.

► **Calculate Booking Costs:** HLLs can be utilized to calculate booking costs based on various factors, such as rental rates, duration of stay, additional services, and taxes. These functions can dynamically compute the total amount due for a booking, making the booking process more flexible and transparent.

► **Generate Reports:** HLLs can play a vital role in generating reports from property and booking data. Functions can be developed to query the database and generate reports in different formats, such as PDF, CSV, or Excel. This enables property owners to track rental revenues, occupancy rates, and other relevant metrics for financial analysis.

By employing HLLs in RentEasy, developers can build a powerful and scalable application with modular and reusable code components. The flexibility offered by HLLs allows RentEasy to adapt to changing requirements and ensure an efficient property rental and booking experience for both property owners and users.

### **3.3.6. Reliability Requirements**

Reliability is equally crucial for RentEasy, the property rental and booking web application, to ensure a seamless and trustworthy user experience. Here are the reliability requirements for RentEasy:

► **Availability:** RentEasy must maintain high availability to accommodate users at any time. The application's infrastructure and servers should be robust enough to handle concurrent user requests and traffic spikes. Ensuring a reliable hosting environment and implementing disaster recovery measures will help minimize downtime and ensure continuous availability.

► **Accuracy:** RentEasy must generate accurate and consistent property listings, booking details, and financial transactions. This means that property information, rental rates, availability dates, and booking statuses must be correctly reflected in the application. Accurate calculations for rental costs, taxes, and fees are essential for a reliable booking experience.

► **Security:** RentEasy must prioritize data security and protect sensitive information. Robust security measures, such as encryption of customer data, secure authentication, and secure payment processing, should be implemented to safeguard user information and financial transactions. Compliance with data privacy laws and regular security audits are essential to ensure the application's security.

► **Scalability:** RentEasy should be designed with scalability in mind to accommodate a growing number of users and property listings. The application's architecture should be scalable to handle increasing data volumes, concurrent bookings, and user interactions without compromising performance.

► **Performance:** RentEasy must perform efficiently to generate property listings and process bookings quickly. Fast data retrieval, efficient database queries, and optimized code are essential to deliver a responsive user experience. Regular performance testing and optimization are necessary to maintain the application's performance standards.

### **3.3.7. Criticality of Application**

The criticality of RentEasy, the property rental and booking web application, is also significant due to its core functions that impact both property owners and users:

► **Revenue Tracking:** RentEasy plays a crucial role in tracking revenue for property owners. By generating invoices for property bookings, the application provides property owners with valuable information about the income earned from their rental properties. This financial data is essential for property owners to assess their property's performance, make informed business decisions, and optimize their pricing strategies.

► **Payment Tracking:** RentEasy helps in tracking payments received from customers for property bookings. For property owners, this information is vital for monitoring cash flow, ensuring timely collection of payments, and managing financial stability. For users, RentEasy provides a transparent payment process, allowing them to track their payment



status and ensure their bookings are confirmed.

► **Compliance with Regulations:** RentEasy can aid property owners in complying with regulations related to tax reporting and financial auditing. By providing accurate and detailed invoices, the application supports property owners in maintaining records that may be required for tax purposes and financial compliance.

► **User Experience:** The criticality of RentEasy is also reflected in the user experience it provides to both property owners and users. The application's reliability, security, and performance directly impact user satisfaction and their willingness to continue using the platform for property rentals and bookings.

### **3.3.8. Safety and Security Consideration**

Safety and security are paramount considerations for RentEasy, the property rental and booking web application, to ensure user confidence and protect sensitive information:

► **Data Security:** RentEasy implements robust data security measures to store and handle user and property data securely. This includes using encryption protocols to safeguard sensitive information, such as user contact details, payment information, and property listings. Access to user data is restricted to authorized personnel only, and data is stored on secure servers with strict access controls.

► **Privacy:** RentEasy prioritizes user privacy and ensures that user data is not collected or shared without explicit user consent. The application provides clear and transparent privacy policies, giving users control over their data and allowing them to manage their privacy settings. Users can choose what information they share with the application and who can access it.

► **Malware Protection:** RentEasy is designed with robust security measures to protect against malware threats, such as viruses and Trojans. The application's servers and codebase undergo regular security audits and scanning to identify and address potential vulnerabilities. RentEasy's web application and mobile apps are made available only through trusted and official sources, reducing the risk of downloading from unverified sources.

► **Secure Authentication:** RentEasy employs secure authentication mechanisms to protect user accounts from unauthorized access. This includes implementing strong password policies, enabling multi-factor authentication, and preventing brute-force login attempts.

### **3.4. Assumptions and Dependencies**

#### **3.4.1 Assumptions**

We will provide a user-friendly interface for RentEasy, ensuring that users can easily navigate and interact with the application. However, users should possess the necessary knowledge to provide specific property details and booking requirements for accurate and relevant property listings and bookings.

- ▶ The data storage server used by RentEasy is assumed to be highly secure and protected. All user and property data will be stored with robust encryption and access controls to ensure data security.
- ▶ RentEasy is designed for users who have basic technical knowledge of technology. While the interface will be user-friendly, users are expected to have a basic understanding of using web applications and interacting with online platforms.
- ▶ The requirements for RentEasy can vary significantly based on the size and scope of the company or user base. The system is designed to handle large-scale operations and a growing number of users and property listings.

#### **3.4.2. Dependencies**

- ▶ RentEasy is dependent on the availability of the internet for all users to access the application successfully. Users must have a stable and reliable internet connection to interact with the platform, browse property listings, and make bookings.
- ▶ Users must enter valid login credentials (username and password) to access RentEasy and perform further actions. If users provide incorrect or invalid login information, the system will not grant access.
- ▶ The efficiency of RentEasy's performance and data handling is crucial for its successful operation. The application's dependency on efficient data retrieval, database management, and code optimization ensures a smooth and responsive user experience.

## **Chapter 4 – System Analysis**

### **4.1 Study of Current System**

### **4.2 Problem and Weaknesses of Current System**

### **4.3 Requirements of New System**

#### **4.3.1 User Requirements**

#### **4.3.2 System Requirements**

### **4.4 Feasibility Study**

#### **4.4.1 Does the System Contribute to the Overall Objectives of the Organization?**

#### **4.4.2 Can the System be Implemented using the current technology and within the given cost and schedule constraints?**

#### **4.4.3 Can the System be integrated With Other System Which are Already in Place?**

### **4.5 Requirements Validation (is concerned with showing that the requirements actually define the system which the customer wants)**

### **4.6 Activity/Process in New System (Use event table)**

### **4.7 Features of New System**

### **4.8 Class Diagram**

### **4.9 System Activity (Use case and/or scenario diagram)**

### **4.10 Object Interaction**

### **4.11 Sequence and Collaboration Diagram**

### **4.12 ER diagram**

### **4.13 Activity diagram**

#### **4.1. Study of Current System**

The current system for property rental and booking management is often manual and paper-based, leading to various drawbacks such as inefficiency, increased costs, and environmental concerns. To overcome these challenges, RentEasy, as a modern web application, offers a more efficient and eco-friendly solution:

- ▶ **Online Property Rental and Booking:** RentEasy replaces the traditional paper-based approach with an online platform, allowing property owners to list their properties digitally and users to browse, book, and manage their bookings online. This eliminates the need for physical paperwork, streamlining the rental process and reducing environmental impact.
- ▶ **Cloud-based Property Management:** RentEasy utilizes a cloud-based system for property management, enabling property owners to store, track, and manage their listings and bookings online. Cloud storage ensures easy access to property information from anywhere, and it promotes accuracy and efficiency in property management.
- ▶ **Mobile Booking and Management:** RentEasy provides a mobile-friendly interface, allowing users and property owners to access the platform from their mobile devices. Mobile booking and management apps enable users to create bookings, view property details, and manage their rentals on the go, enhancing convenience and turnaround times.

#### **4.2. Problem and Weakness of current System**

The current system of manual property rental and booking management faces several problems and weaknesses, which RentEasy addresses and overcomes:

- ▶ **Inefficiency:** The manual handling of property rental and booking processes can be inefficient, leading to delays in property listing updates, booking confirmations, and responses to user inquiries. RentEasy's digital platform streamlines these processes, enabling real-time updates and faster communication between property owners and users, improving overall efficiency.
- ▶ **Cost:** Traditional property rental processes involve significant costs related to physical paperwork, printing, postage, and storage of paper-based records. RentEasy's paperless approach reduces these expenses, making property management more cost-effective for property owners and environmentally sustainable.
- ▶ **Environmental Impact:** The paper-based approach in the current system contributes to environmental pollution through paper production and waste. RentEasy's eco-friendly digital platform significantly reduces paper usage, promoting a greener and more environmentally responsible approach to property rental and booking management.
- ▶ **Security:** Physical paperwork in the current system can be vulnerable to loss, theft, or

damage, potentially compromising sensitive property and user information. RentEasy implements robust data security measures, ensuring the safe storage and handling of user and property data, protecting privacy and financial security.

► **Compliance:** The current system may require businesses to comply with various regulations related to property management and financial transactions. RentEasy's automated processes and data management facilitate compliance with relevant regulations, making it easier for property owners to meet their legal obligations.

### **4.3. Requirements of New System**

#### **4.3.1. User Requirements**

- The RentEasy app should have a user-friendly interface, ensuring ease of use for property owners and users. The navigation and property search should be intuitive and straightforward.
- RentEasy should provide property owners with the flexibility to customize their property listings, including the option to add property images, descriptions, amenities, and rental rates.
- Users should be able to track the status of their bookings, such as whether a booking is confirmed or pending, and access booking details in their dashboard for better financial management.
- RentEasy should offer users the ability to export booking and property data in various formats, such as PDF, CSV, or Excel, for their record-keeping and analysis needs.

#### **4.3.2. System Requirements**

- RentEasy should be compatible with popular operating systems, such as Windows, macOS, and Linux, to ensure broad accessibility for property owners and users.
- The web application should be compatible with major web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge, to provide a seamless experience for users on different platforms.
- RentEasy should be designed to run smoothly on a range of hardware configurations, ensuring optimal performance for users with varying devices and internet connections.
- The application should have stable internet connectivity to enable users to access the platform, sign up/login, browse properties, make bookings, and update

their listings and bookings seamlessly.

#### **4.4. Feasibility Study**

##### **4.4.1. Does the System Contribute to the Overall Objectives of the Organization?**

RentEasy can contribute to the overall objectives of the organization by streamlining property rental and booking management. The application automates the booking process, reducing manual efforts and freeing up employees to focus on other tasks. By eliminating the need for paper-based invoices and optimizing invoice processing, RentEasy can help reduce costs. The app's accuracy in handling property listings and bookings can enhance customer satisfaction, providing a more efficient and convenient experience for users.

##### **4.4.2. Can the System be Implemented using the current technology and within the given cost and schedule constraints?**

The feasibility of implementing RentEasy depends on various factors. The application should be compatible with the current technology, including operating systems, web browsers, and hardware used by the organization. RentEasy also requires internet connectivity to function effectively. In terms of cost constraints, the implementation cost should align with the organization's budget, which may vary based on the app's features and the organization's size. The app's implementation timeframe will depend on its complexity and the available resources.

##### **4.4.3. Can the System be Integrated With Other System Which are Already in Place?**

RentEasy should be designed to integrate seamlessly with other existing systems, such as accounting software. This integration will help streamline property management workflows and save time for property owners. The level of integration will be determined by the app's features and the compatibility with the systems already in place.

## **4.5. Requirements Validation**

Requirement's validation involves ensuring that the identified requirements accurately define the system the customer wants.

- **Use Case Validation:** Validate the requirements by creating the use cases or user stories that describe specific interactions or scenarios within the app. Review these use cases with stakeholders to ensure they cover all necessary functionality and accurately represent the desired user experience.
- **Peer Review:** Engage subject matters experts or experienced professionals to conduct a peer review of the requirements. They can provide valuable insights, identify any gaps or inconsistencies, and validate the feasibility and relevance of the requirements.
- **Non-functional Requirements:** These are the requirements that define the non-functional aspects of the app, such as its performance, security and usability. For example, we need to validate that the app is fast enough to generate invoices, that it is secure from unauthorized access, and that it is easy to use for users of all skill levels.
- **Functional Requirements:** These are the requirements that define the core functionality of the app. For example, we need to validate that the app can generate invoices, that the invoices are accurate, and that the app can be used by users of all skill levels.

#### **4.6. Activity/Process in New System**

The activity/process in the new system can be represented using an event table, which outlines the actions or events performed by the user and the corresponding system responses. It provides a high-level overview of the system's functionalities and interactions.

<b><u>EVENT</u></b>	<b><u>DESCRIPTION</u></b>
User signs up	Users can create a new account on RentEasy by providing essential details such as name, email, contact information, and property details for property owners.
User logs in	Registered users can log in to RentEasy using their email and password for secure access to their accounts and property listings.
Browse properties	Users can access the platform and explore various listed properties available for rent based on their preferences and location.
Book property	Users can book a property by selecting desired dates, specifying the number of guests, and completing the booking process with payment information.
List a property	Property owners can list their properties by providing property details, images, availability dates, and rental rates for potential users to view and book.
Edit property listing	Property owners can edit their property listings, updating property details, rental rates, and availability as needed.
Delete property listing	Property owners have the option to delete their property listing from RentEasy if it is no longer available for rent.

**Table 1 - Event table for an invoice management app**



#### **4.7. Features of New System**

- ▶ **Property Listing Management:** RentEasy allows property owners to list their properties with ease, providing a user-friendly interface to enter essential property details, amenities, images, and rental rates.
- ▶ **Customizable Property Listings:** Property owners can customize their property listings by adding logos and selecting from various templates to showcase their properties uniquely.
- ▶ **Booking Tracking:** RentEasy enables users to track the status of their bookings, indicating whether a booking is confirmed or pending, providing better financial management for property owners.
- ▶ **Booking Confirmation and Notification:** Users receive booking confirmation and notifications through the platform, ensuring a smooth and timely booking process.
- ▶ **Secure Data Handling:** RentEasy prioritizes data security to protect sensitive customer and payment information, ensuring user privacy and financial security.
- ▶ **Scalability:** The RentEasy application is designed to be scalable, accommodating the growth of the platform and its user base as the business expands.
- ▶ **Dashboard Access:** Users and property owners have access to personalized dashboards, providing a centralized view of their bookings, property listings, and financial details.
- ▶ **Mobile Compatibility:** RentEasy offers a mobile-friendly interface, allowing users and property owners to access and manage their accounts and bookings on the go.

## 4.8. Class Diagram

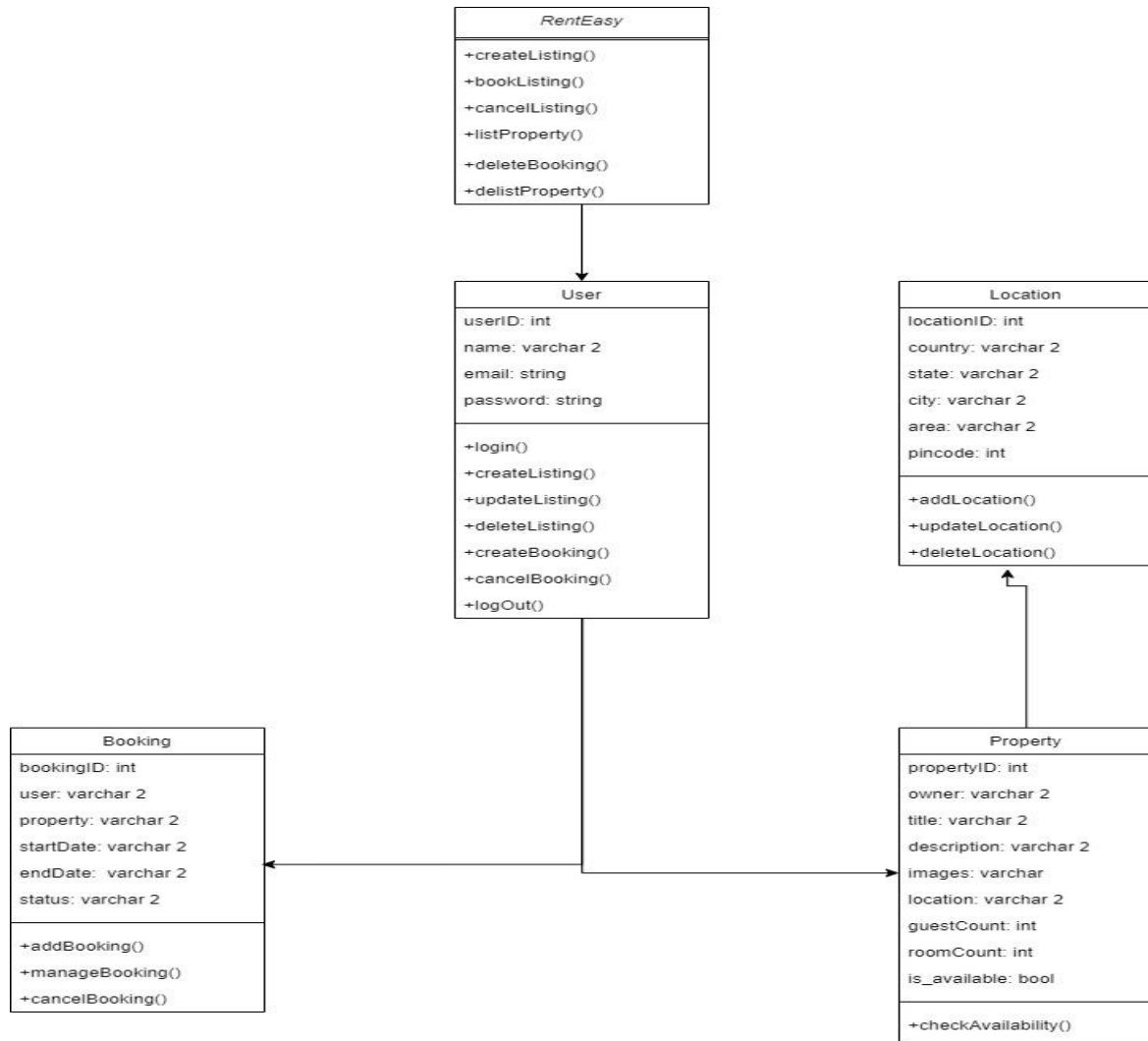
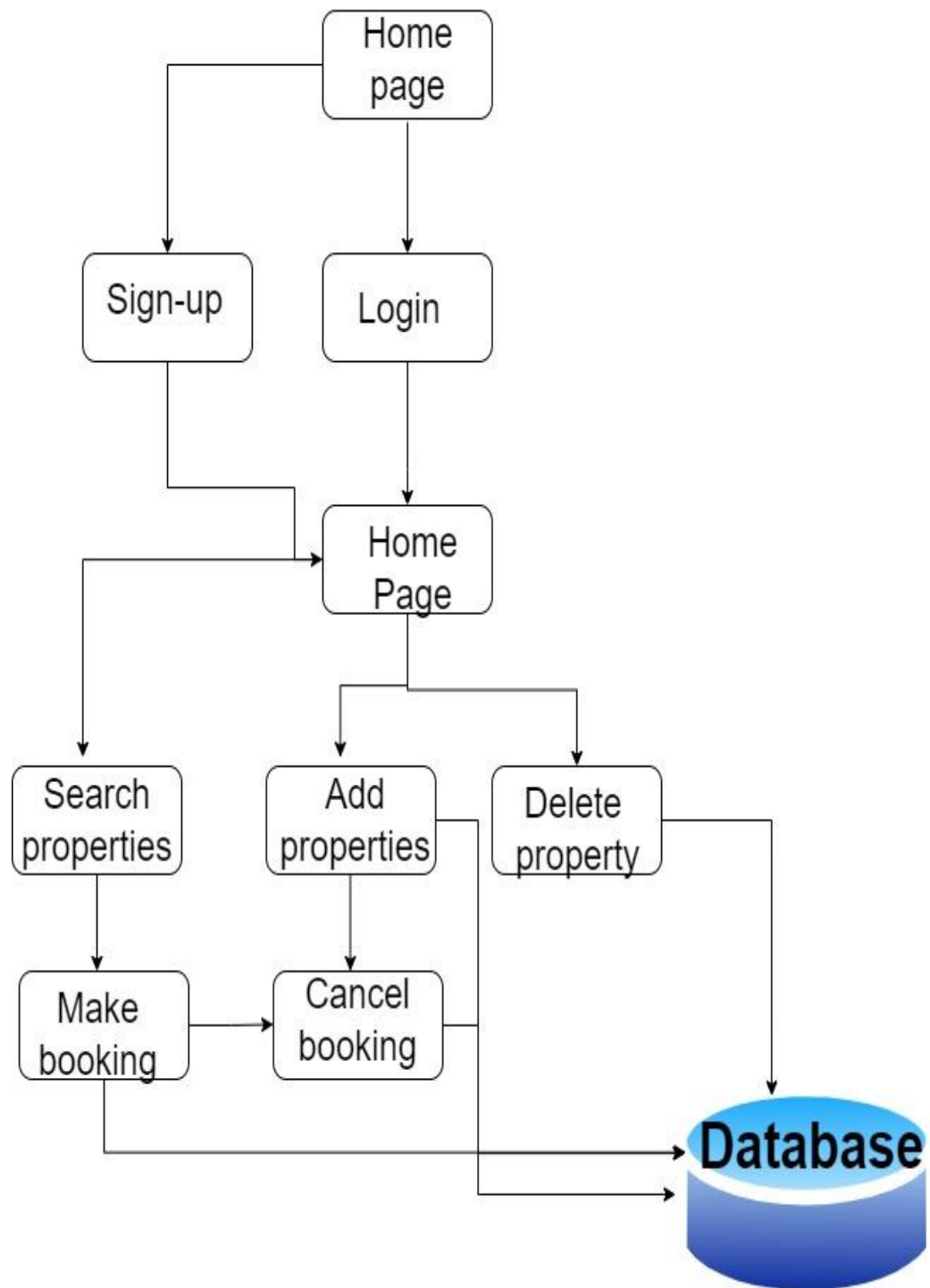


Fig – 4.8 : class diagram

#### 4.9. System Activity



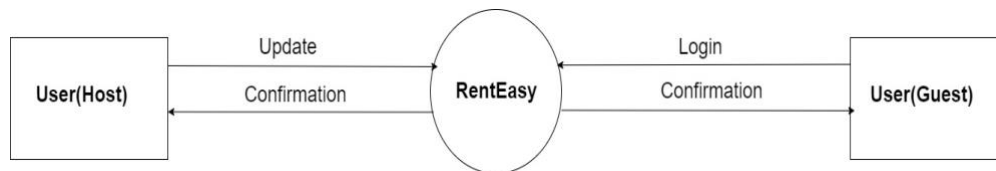
**Fig – 4.9 : System Activity Diagram**

#### **4.10. Object Interaction**

- ▶ **User interacts with the app to create an account**  
The user provides essential information such as name, email address, contact, property details (for property owners), etc., to create a RentEasy account.
- ▶ **User interacts with the app to browse and search for properties**  
Users can explore listed properties, filter and search based on location, rental rates, amenities, and other preferences.
- ▶ **User interacts with the app to book a property**  
Users can book a property by selecting desired dates, specifying the number of guests, and completing the booking process with payment information.
- ▶ **The app interacts with the database to store booking data**  
The booking details, including property information and user details, are stored in the database for future reference.
- ▶ **User interacts with the app to view booking details**  
Users can view their booking details, including booking status, property details, and payment status, in their dashboard.
- ▶ **User interacts with the app to manage bookings**  
Users can modify booking details, such as updating dates, adding guests, or canceling bookings if necessary.
- ▶ **User interacts with the app to list a property**  
Property owners can list their properties by providing property details, images, availability dates, and rental rates.
- ▶ **The app interacts with the database to store property listings**  
The property listing details are stored in the database for users to view and book.
- ▶ **User interacts with the app to edit property listings**  
Property owners can edit their property listings, updating property details, rental rates, and availability as needed.
- ▶ **User interacts with the app to delete property listings**  
Property owners have the option to delete their property listings from RentEasy if they are no longer available for rent.

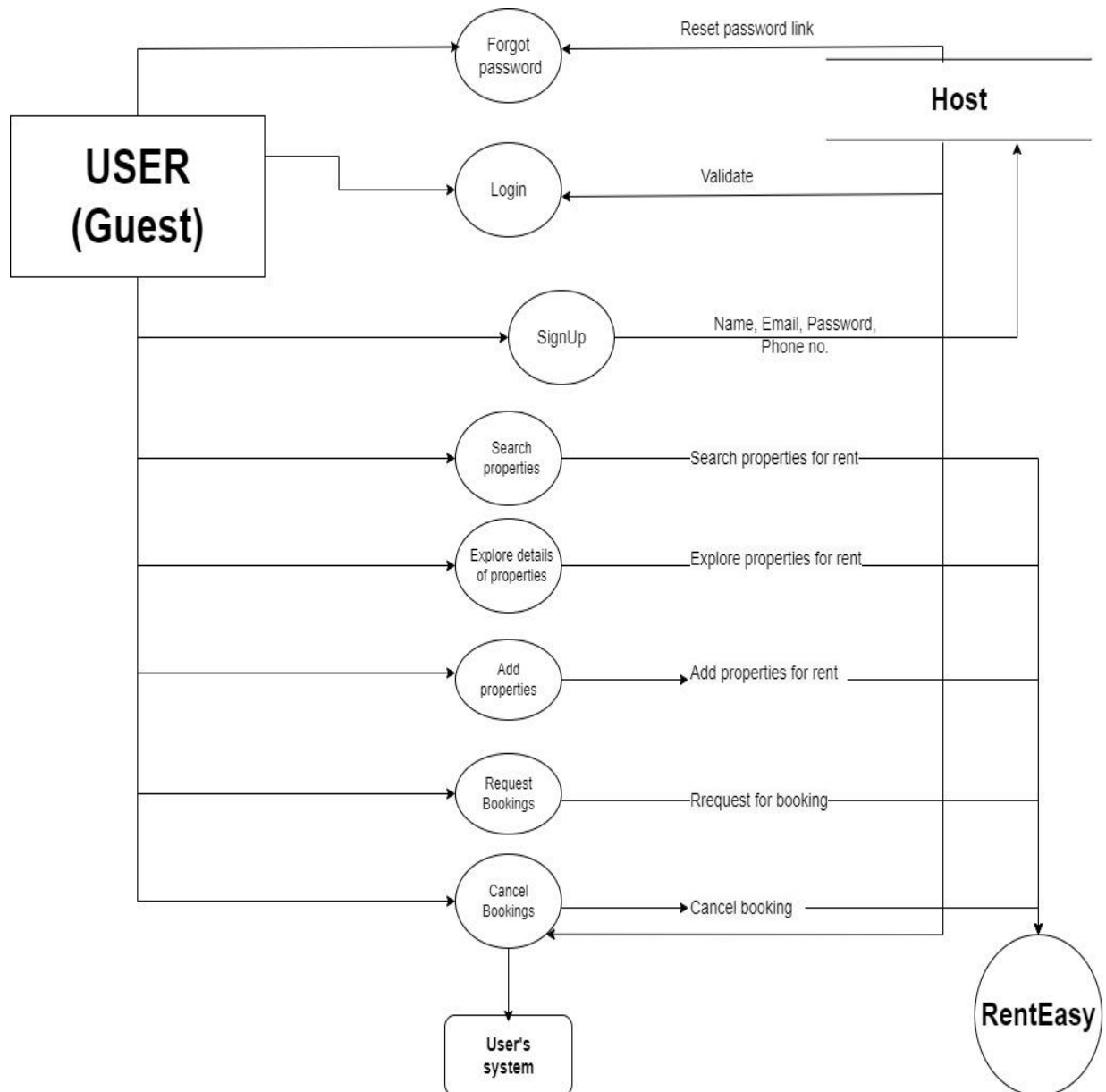
## **4.11. Data Flow Diagram (DFD)**

### **Context Level Data Flow Diagram**



**Fig – 4.11.1 : Context Level Data Flow Diagram**

### First Level Data Flow Diagram



**Fig – 4.11.2 : First Level Data Flow Diagram**

## 4.12. ER Diagram

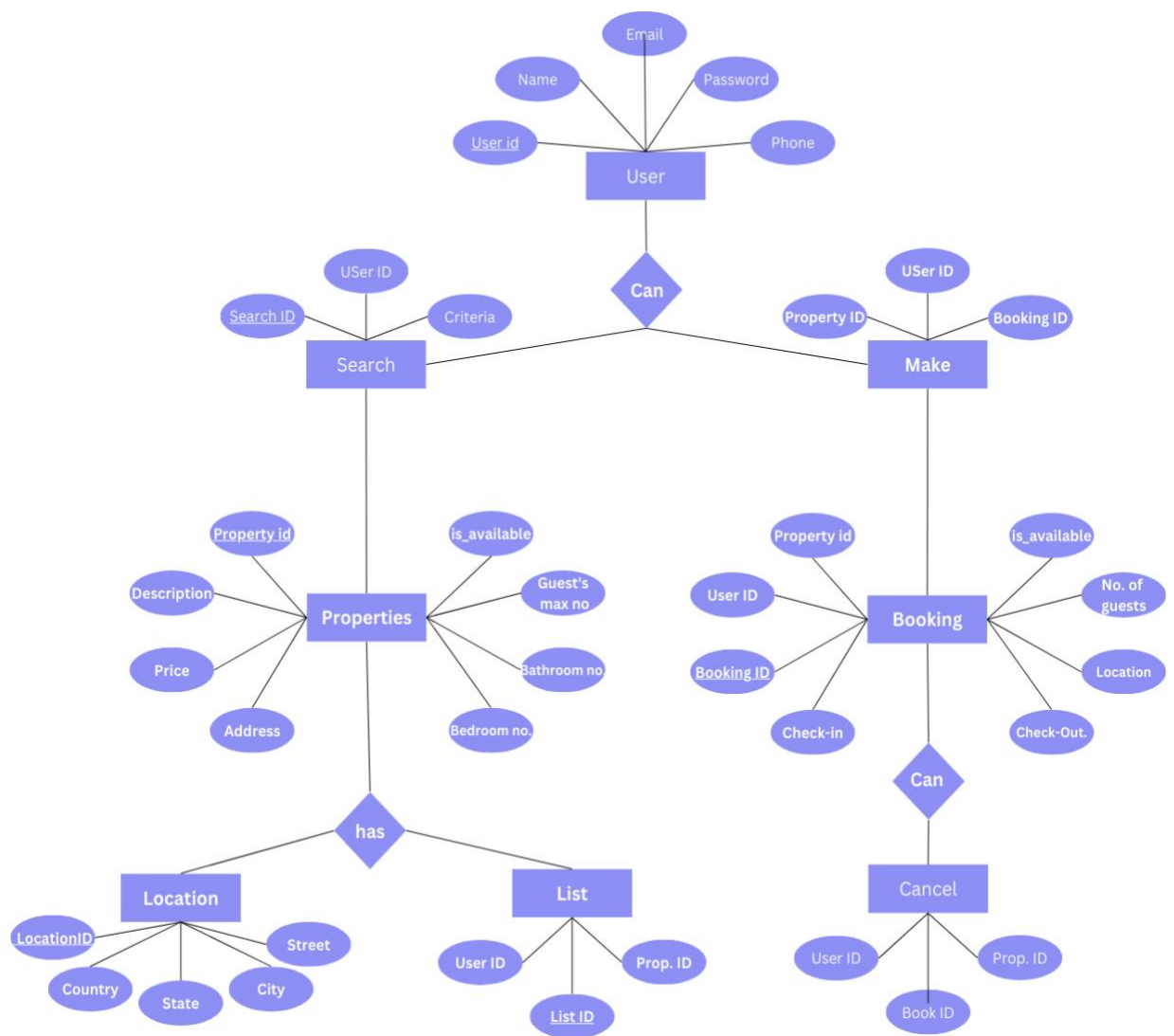
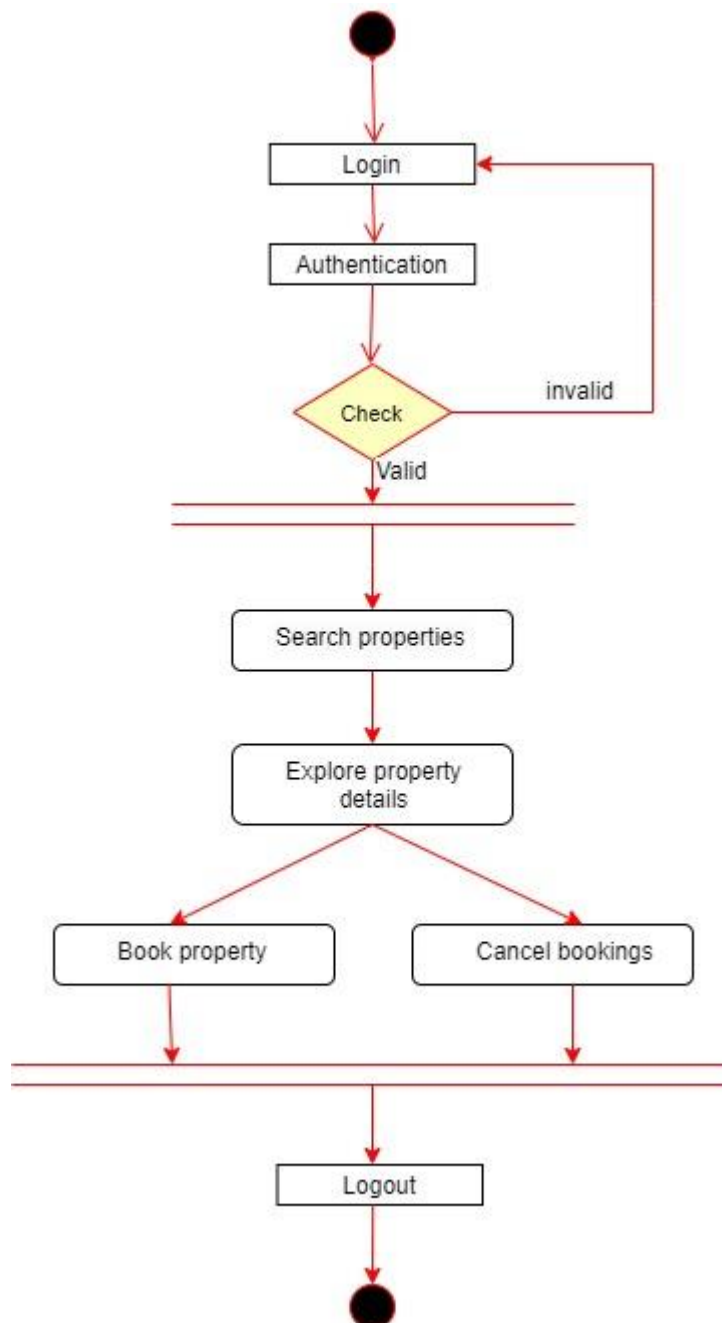


Fig – 4.12 : Entity-Relationship diagram

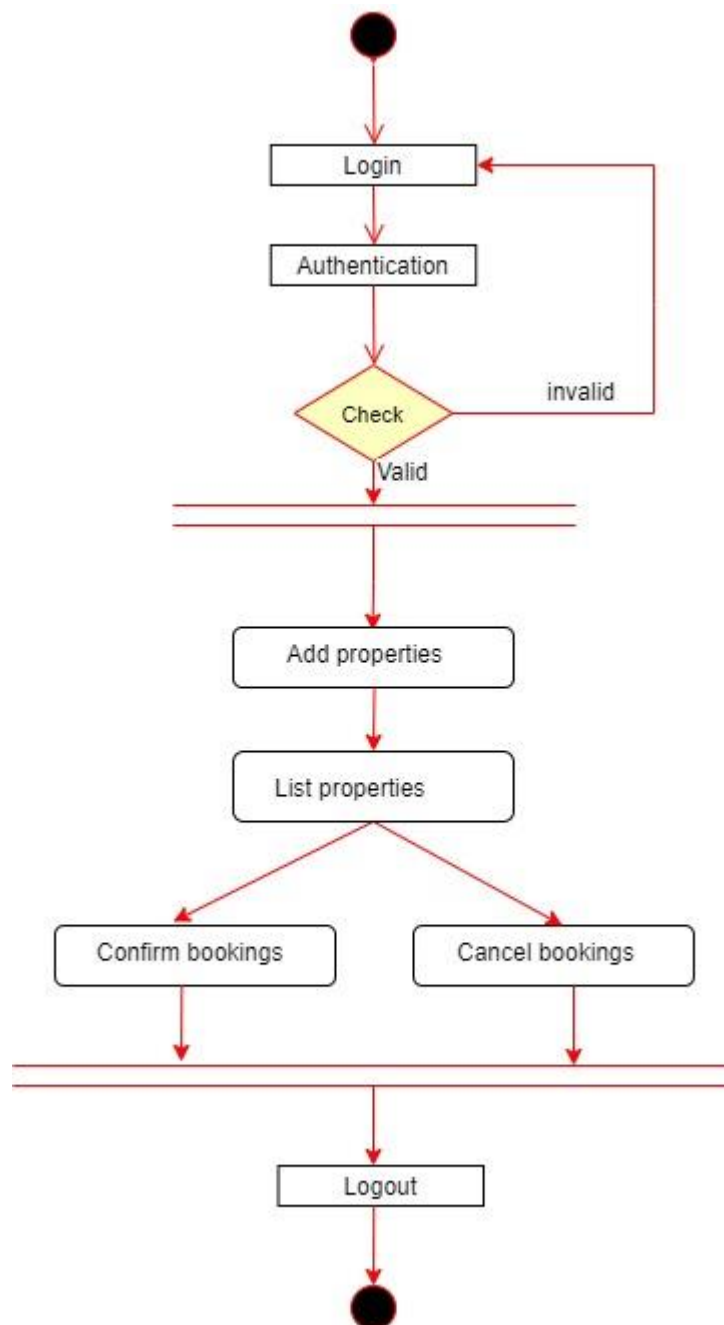
### **4.13. Activity Diagram**

**Activity diagram for User(Guest)**



**Fig – 4.13.1 : Activity diagram for guest**



**Activity diagram for User(Host)****Fig – 4.13.2 : Activity diagram for host**

## **Chapter 5 – System Design**

### **5.1. System Application Design**

#### **5.1.1. Method Pseudo Code**

### **5.2. Database Design/Data Structure Design**

#### **5.2.1. Table and Relationship**

#### **5.2.2. Logical Description of Data**

### **5.3. Input/output and Interface Design**

#### **5.3.1. State Transition / UML Design**

#### **5.3.2. Samples of Forms, Reports and Interface**

## **5.1. System Application Design**

Here is a system application design for an renteasy webapp:

SystemComponents

### **Frontend**

- Userinterface(UI) for users to interact with the app.
- Backend model for the frontend to communicate with the backend.

### **Backend**

- Data Models of Properties and bookings.
- Data Models for UserAuthentication.
- Other services like managing all the listings and bookings.

### **SystemArchitecture**

- The frontend and backend are decoupled,so that they can be developedand scaled independently.
- The frontend is a desktop app,which can be deployed to any laptops.

### **SystemFeatures**

- Users can create listings
- Users can book properties
- Users can delete listings
- Users can cancel bookings
- User can search for properties

### **Scalability**

- The app is able to handle a large number of users and requests.

### **Performance**

- The app is able to respond to user requests quickly.
- The app does not slow down or crash when underheavy load.

### **Usability**

- The app is easy to use and navigate.
- The app is visually appealing and easy to read.

### 5.1.1. Method Pseudo Code

```
import getListings, { IListingsParams } from
"@/app/actions/getListings";
import { getCurrentUser } from "@/actions/getCurrentUser";
import ClientOnly from "@/components/ClientOnly";
import EmptyState from "@/components/EmptyState";
import Container from "@/components/Container";
import ListingCard from "@/components/listings/ListingCard";

interface HomeProps {
  searchParams: IListingsParams;
}

const Home = async ({ searchParams }: HomeProps) => {
  const listings = await getListings(searchParams);
  const currentUser = await getCurrentUser();

  if (listings.length === 0) {
    return (
      <ClientOnly>
        <EmptyState showReset />
      </ClientOnly>
    );
  }
  return (
    <ClientOnly>
      <Container>
        <div
          className=" pt-24 grid grid-cols-1 sm:grid-cols-
2 md:grid-cols-3 lg:grid-cols-4 xl:grid-cols-5 2xl:grid-cols-6 gap-
8">
          {listings.map((listing: any) => (
            <ListingCard
              currentUser={currentUser}
              key={listing.id}
              data={listing}/>
            ))}
        </div>
      </Container>
    </ClientOnly>
  );
};

export default Home;
```

Fig 5(a): Pseudo Code

## **5.2. Database Design/ Data structure Design**

### **5.2.1. Table and Relationship**

<b><u>TABLE</u></b>	<b><u>DESCRIPTION</u></b>
Users	Stores information about each user's details, such as it's email address, contact, company name, address, etc
Properties	Stores information about Properties like Property id, user id, location, name, description, price, availability, etc
Bookings	Stores information like booking id, user id, property id, user name, property name, total amount, start date, end date, availability, address, description, etc

**Table2:TableandRelationship**

### **5.2.2. Logical Description of Data**

#### **Users:**

ID: A unique identifier for each user.

Name: User Name

Email: The email ID of the user, used for authentication.

Password: Encrypted password for user login.

Contact: Phone number of the user.

#### **Property:**

ID: A unique identifier for each property.

User ID: A foreign key referencing the user who owns the property.

Title: Title of the property listing.

Description: Description of the property.

Address: The address of the property.

Rental Rate: The rental rate or cost for renting the property.

Amenities: A list of amenities or features provided by the property.

Availability: A Boolean value indicating whether the property is available for

booking (true) or not (false).

**Bookings:**

ID: A unique identifier for each booking.

User ID: A foreign key referencing the user who made the booking.

Property ID: A foreign key referencing the property that is booked.

Check-in Date: The date on which the booking starts (check-in date).

Check-out Date: The date on which the booking ends (check-out date).

Guest Count: The number of guests staying in the property for the booking.

Status: The status of the booking, such as confirmed, pending, or canceled.

### 5.3. Input/output and Interface Design

**Input:**

- The user is able to create an account with RentEasy.
- The user can list their own property by providing details such as property title, description, address, rental rate, amenities, etc.
- The user can create a booking for a property by selecting the check-in date, check-out date, and specifying the number of guests.
- The user can cancel a booking if needed.
- The user can delist their property from RentEasy if they no longer want to offer it for rent.

**Output:**

- RentEasy stores the user's property listing details in the database.
- RentEasy stores the booking details in the database.
- The app displays the user's listed properties on their dashboard.
- The app displays all the generated bookings on the user's dashboard.
- The user receives confirmation and notifications regarding their bookings and property listings.

**Interface Design:**

- RentEasy has a user-friendly interface with intuitive navigation for easy usage.
- The app uses clear and concise language to guide users through the property listing and booking process.
- RentEasy is responsive, ensuring a seamless user experience on various devices and screen sizes.
- Users can easily create, edit, and manage their property listings and bookings.

- Users have the option to upload images of their property to enhance the listing's appeal.
- RentEasy provides an option for users to manage their bookings and view the status of each booking.

### 5.3.1. State Transition / UML Diagram

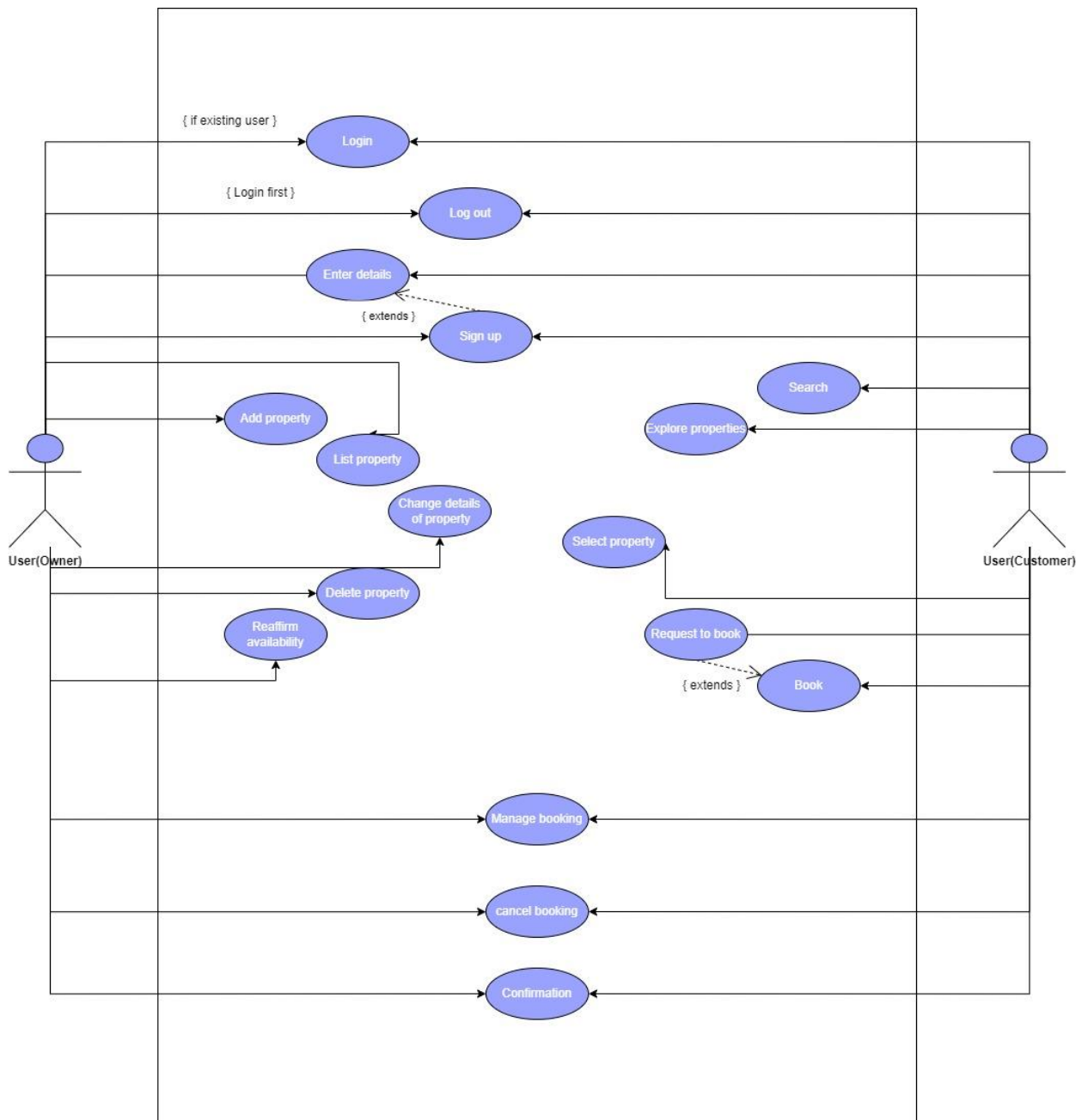


Fig – 5.3.1 : Use Case Diagram

### 5.3.3. Samples of Forms, Report and Interface

#### SignupPage:

The screenshot displays the RentEasy website interface with a 'Register' modal form overlaid. The background shows a grid of property listings with categories like Beach, Windmills, Modern, and Countryside. The modal form is titled 'Register' and includes a close button (X). It contains the following elements:

- Welcome to RentEasy**
- Create an account!**
- Email** input field
- Name** input field
- Password** input field
- Continue** button (red)
- Continue with Google** button (with Google logo)
- Already have an account? Log in** link

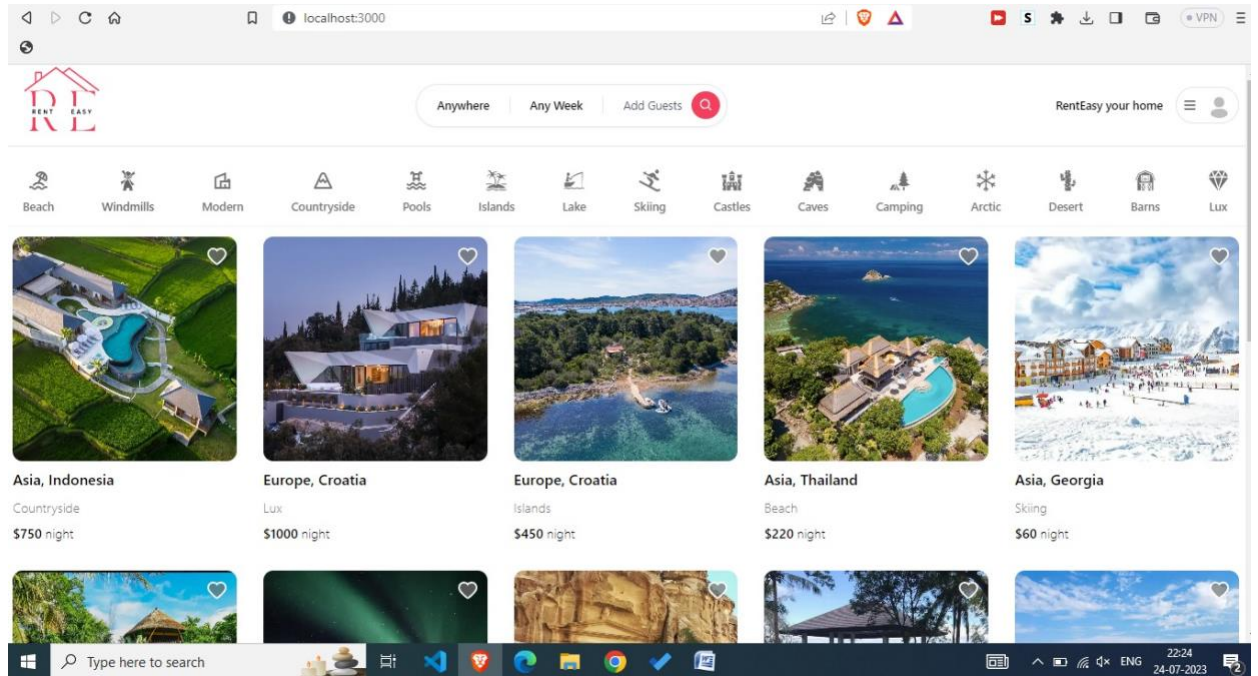
#### LoginPage:

The screenshot displays the RentEasy website interface with a 'Login' modal form overlaid. The background shows the same grid of property listings as the previous image. The modal form is titled 'Login' and includes a close button (X). It contains the following elements:

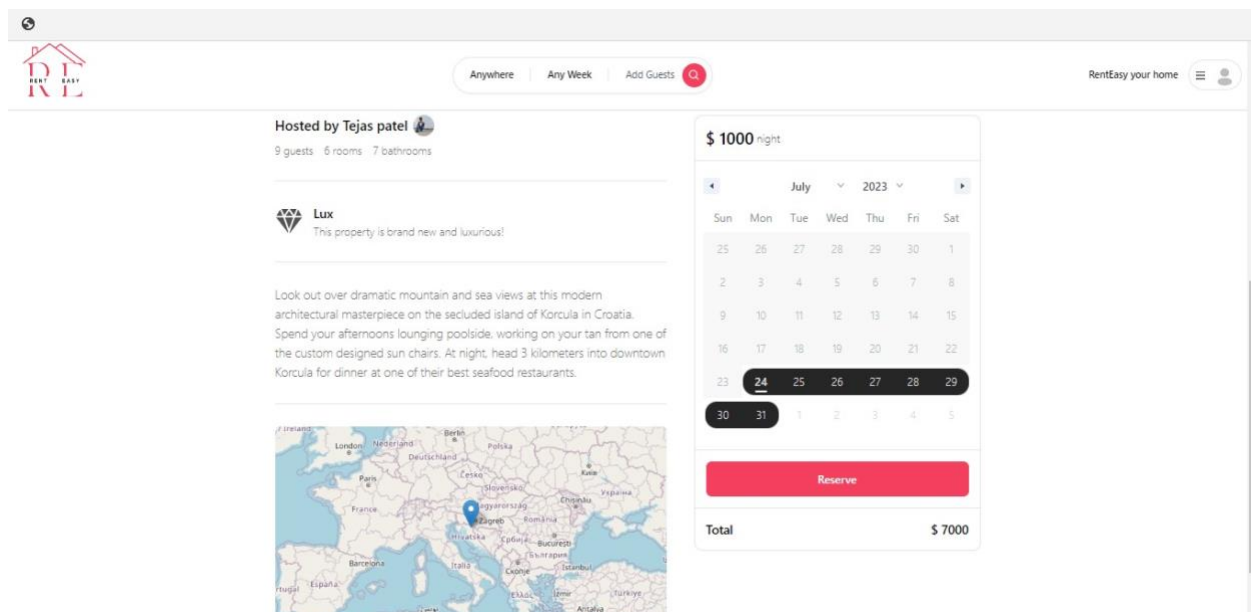
- Welcome back**
- Login to your account!**
- Email** input field
- Password** input field
- Continue** button (red)
- Continue with Google** button (with Google logo)
- First time using RentEasy? Create an account** link

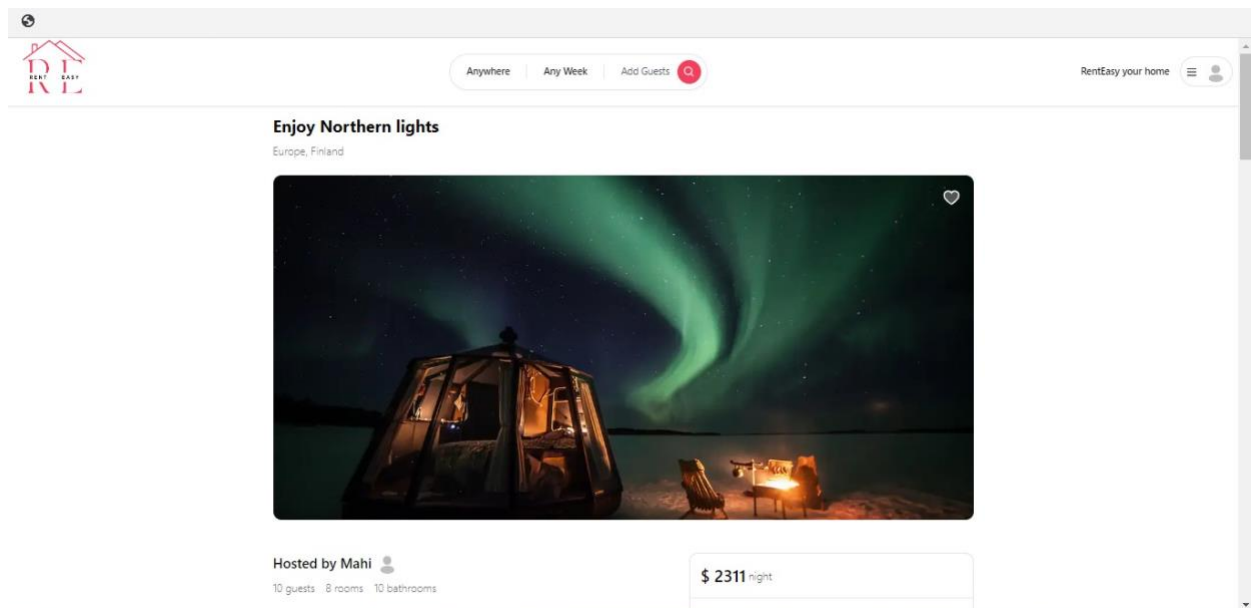


## Home Page:



## Property Description:



**Single Property Page:**

## **Chapter 6 - System Testing**

### **6.1.Test Cases**

## **6.1. Test Cases**

Test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective and verify compliance with a specific requirement.

### **❖ Testing Plan**

#### **Planning Steps**

- 1) Functionality Testing**
- 2) Usability Testing**
- 3) Interface Testing**
- 4) Performance Testing**
- 5) Security Testing**

#### **1) Functionality Testing**

Test for – all the links in web pages, database connection, forms used in the web pages for submitting or getting information from user, Cookie testing.

##### **➤ Check for all the links:**

- Test the outgoing links from all the pages from specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send the email to users from web pages.
- Test to check if there are any orphan pages.
- Lastly in link checking, check for broken links in all above-mentioned links.

##### **➤ Test Forms in all pages:**

Forms are the integral part of any website. Forms are used to get information from users and to keep interaction with them. So what should be checked on these forms?

- First check all the validations on each fields.
- Check for the default values of fields.
- Wrong inputs to the fields in the forms.

##### **➤ Cookies testing:**

Cookies are small files stored on user machine. These are basically used to maintain the session mainly login sessions. Test the application by enabling or disabling the cookies in your browser options. Test if the cookies are encrypted before writing to user machine. If you are testing the session cookies (i.e. cookies expire after the session ends) check for login sessions and user stats after session end. Check effect on application security by deleting the cookies.

➤ **Validate your HTML/CSS:**

If you are optimizing your site for Search engines, then HTML/CSS validation is very important. Mainly validate the site for HTML syntax errors. Check if site is crawl able to different search engines.

➤ **Database Testing:**

Data consistency is very important in web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality. Check if all the database queries are executing correctly, data is retrieved correctly and also updated correctly. More on database testing could be load on DB.

## 2) **Usability Testing:**

➤ **Test for navigation:**

Navigation means how the user surfs the web pages, different controls like buttons, boxes or how user using the links on the pages to surf different pages. Usability testing includes: Web site should be easy to use. Instructions should be provided clearly. Check if the provided instructions are correct means whether they satisfy purpose. Main menu should be provided on each page. It should be consistent.

➤ **Content:**

Content should be logical and easy to understand. Check for spelling errors. Use of dark colours annoys users and should not be used in site theme. Check if the website follows some standards that are used for web page and content building.

Content should be meaningful. All the anchor text links should be working properly. Images should be places properly with proper sizes. These are some basic standards that should be followed in web development.

➤ **Other user information for user help:**

Like search option, sitemap, help files, etc. Sitemap should be present with all the links in web sites with proper tree view of navigation. Check for all links on the sitemap. “Search in the site” option will help users to find content pages they are looking for easily and quickly. These are all optional items and if present should be validated.

### 3) **Interface Testing:**

The main interfaces are:

- Web server and application server interface
- Application server and Database server interface

Check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users. Check what happens if user interrupts any transaction in-between? Check what happens if connection to web server is reset in between?

### 4) **Performance Testing:**

Web application should sustain to heavy load. Web performance testing should include:

- Web Load Testing
- Web Stress Testing

Test application performance on different internet connection speed. In web load testing test if many users are accessing or requesting the same page. Can system sustain in peak load times? Site should handle many simultaneous user requests, large input data from users, Simultaneous connection to DB, heavy load on specific pages etc.

**Stress testing:** Generally, stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checked how system reacts to stress and how system recovers from crashes. Stress is generally given on input fields, login and sign up areas.

In web performance testing, web site functionality on different operating systems, different hardware platforms are checked for software, hardware memory leakage errors.

### 5) **Security Testing:**

Following are some test cases for web security testing:

- Test by pasting internal URL directly into browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing URL options directly, which is not related to logged in user. Access should be denied for this user to view others stats.
- Try some invalid inputs in input fields like login username, password, input text boxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given download option.
- Test if SSL is used for security measures. If used proper message should get displayed when user switch from non-secure http:// pages to secure https:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on web server.

## ❖ Testing Strategies:

### 1) White Box Testing:

White Box Testing (WBT) is also called Structural or Glass Box Testing. White Box Testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all the internal components have been adequately exercised.

#### ➤ Why do we do White Box Testing?

To ensure:

- That all independent paths within a module have been exercised at least once.
- All logical decisions verified on their true and false values.
- All loops executed at their boundaries and within their operational bounds internal data structures validity.

#### ➤ Need of White Box Testing?

To discover the following types of bugs:

- Logical error tends to creep into our work when we design and implement functions, conditions or controls that are out of the program.
- The design errors due to difference between logical flow of the program and the actual implementation.
- Typographical errors and syntax checking.

**Limitation of WBT:**

Not possible for testing each and every path of the loops in program. This means exhaustive testing is impossible for large systems. This does not mean that WBT is not effective. By selecting important logical paths and data structure for testing is practically possible and effective.

**2) Black Box Testing:**

- Black Box Testing treats the system as a “black-box”, so it doesn’t explicitly use Knowledge of the internal structure or code. Or in other words the Test engineer need not know the internal working of the “Black-box” or application.
- Main focus in black box testing is on functionality of the system as a whole. The term ‘behavioural testing’ is also used for black box testing and white box testing is also sometimes called ‘structural testing’. Behavioural test design is slightly different from black-box test design because the use of internal knowledge isn’t strictly forbidden, but it’s still discouraged.
- Each testing method has its own advantages and disadvantages. There are some bugs that cannot be found using only black box or only white box. Majority of the application are tested by black box testing method. We need to cover majority of test cases so that most of the bugs will get discovered by black box testing.
- Black box testing occurs throughout the software development and Testing life cycle i.e. in Unit, Integration, System, Acceptance and regression testing stages.
- **Advantages of Black Box Testing:**
  - Tester can be non-technical
  - Used to verify contradictions in actual system and the specifications.
  - Test cases can be designed as soon as the functional specifications are complete.



- **Disadvantages of Black Box Testing:**

- The test inputs need to be from large sample space.
- It is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult. Chances of having unidentified paths during this testing.

## **6.2. Test Cases**

Test Case ID	Test Case Description	Expected Result
TC-01	Create a property listing with all required information	The property listing should be created successfully
TC-02	Create a property listing with invalid information	The property listing should not be created and an error message should be displayed
TC-03	Edit a property listing	The property listing details should be updated successfully
TC-04	Delete a property listing	The property listing should be deleted successfully
TC-05	View property bookings	The app should display all bookings for the selected property accurately
TC-06	Create a booking for a property	The booking should be created successfully
TC-07	Cancel a booking	The booking should be canceled successfully
TC-08	Delist a property from RentEasy	The property should be delisted successfully
TC-09	Test the security of the app	The app should be secure, and no sensitive data should be compromised
TC-10	Test the scalability of the app	The app should be scalable and able to handle a large number of properties and bookings

## **Chapter 7 - Conclusion**

In conclusion, RentEasy is a sophisticated property rental platform that aims to revolutionize the way users list, book, and manage properties. The application has been thoughtfully designed to address the inefficiencies and limitations of traditional property rental systems. By embracing modern technologies and user-friendly interfaces, RentEasy provides a seamless and intuitive experience for both property owners and potential tenants.

The platform offers a range of robust features, allowing property owners to effortlessly list their properties with all the necessary details. They can easily edit, update, or remove listings based on their preferences and availability. The ability to add images and customize property descriptions enhances the attractiveness of the listings, attracting a broader audience of potential renters.

For tenants, RentEasy offers an efficient and secure booking process. Users can search for properties based on their desired location, rental rates, amenities, and other criteria. Once a suitable property is found, tenants can proceed to create bookings with ease, selecting check-in and check-out dates, and specifying the number of guests. The app also provides detailed tracking of booking status, making it convenient for users to manage their accommodations.

RentEasy's commitment to data security and privacy ensures that sensitive user information, including personal details and payment information, is handled with utmost care and protection. This level of security instills trust among users and contributes to RentEasy's credibility as a reliable and safe platform for property rental needs.

In essence, RentEasy is poised to revolutionize the property rental market, bringing a blend of modern technology, user-centric design, and data security to streamline the process of listing and booking properties. By providing a platform that caters to the needs of both property owners and tenants, RentEasy aims to elevate the rental experience and foster a thriving community of property management and rental services.

## **Chapter 8 – Bibliography**

## **Bibliography**

- <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>
- <https://react-icons.github.io/react-icons/search?q=cross>
- <https://www.mongodb.com/>
- <https://www.google.com/>
- <https://nextjs.org/docs>
- <https://tailwindui.com/components?ref=sidebar>