Right Stay

(MAKE YOUR HOME, ANYWHERE)

Project Report

Submitted in Partial Fulfilment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

ANURAG SHARMA (2100950100012)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MGM's College of Engineering &Technology, Noida

May, 2025

TABLE OF CONTENTS

Page No.

CERTIFICATE

To certify that the Project Report titled Right Stay (MAKE YOUR HOME, ANYWHERE) is submitted by Anurag Sharma (21009501000012), in partial fulfilment of the requirement for the award of degree B.Tech. in the Department of Computer Science and Engineering of MGM's College of Engineering and Technology associated with AKTU Lucknow, is documentation of the writing performed by the candidate under my guidance. The content contained in this report is the work of the author and has not previously been submitted for the award of any other degree.

Date:

Supervisor Signature

Name of Supervisor: Dr. Karamjeet Kaur

Designation: HOD CS

DECLARATION

I hereby declare that this submission is our work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: Anurag Sharma

Roll No: 2100950100012

ACKNOWLEDGEMENT

I pleased to publish the report of B. Tech. Project done during B. Tech. Final Year. I would like to express our deep and sincere gratitude to **Dr. Karamjeet Kaur, Department of Computer Science & Engineering, MGM's College of Engineering and Technology, Noida** who supported us continuously throughout our work. His /Her dedication, attention to detail, and hard work have inspired me. Our efforts have come to fruition only because of the conscious efforts of him/her.

I take this opportunity to express our gratitude to Dr. Karamjeet Kaur, Head, Department of Computer Science & Engineering, MGM's College of Engineering and Technology, Noida for providing full support and help to produce the project.

I would also like to express our gratitude to all the faculty members of the department for their kind assistance and cooperation in developing my project work. Finally, thanks to my friends for their help in the completion of the project.

Signature:

Name: Anurag Sharma

Roll No: 2100950100012

Date:

ABSTRACT

The RightStay platform is a global, technology-driven marketplace that reimagines short-term accommodations by connecting travellers with property owners offering unique lodging options. As an alternative to traditional hotels, RightStay leverages a peer-to-peer model to provide flexible, cost-effective, and personalized travel experiences, enhancing local engagement and cultural immersion. Inspired by the guest-cantered hospitality of traditional Japanese ryokan, RightStay focuses on creating a seamless user experience and high-quality service standards that cater to modern travel preferences while preserving local culture. This influence from Ryokan'sdesign and service models makesRightStay both more user-friendly and business-oriented, integrating personalized guest services, thoughtful interface design, and elevated hospitality practices.

Through an array of innovative modules—including user management, property listing, booking, immersive experiences, and environmental impact—the platform addresses the evolving demands of modern travellers. RightStay's comprehensive system simplifies property management, integrates secure transactions, and offers data-driven personalization to improve user satisfaction. Additionally, by promoting sustainable and community-focused tourism, RightStay aims to mitigate the environmental and social impacts associated with the travel industry.

Through advanced technology and user-centered design, RightStay not only empowers property owners economically but also fosters responsible tourism that aligns economic growth with social and environmental responsibility. This project demonstrates how digital transformation in the vacation rental industry can create a scalable, inclusive, and resilient lodging model that benefits society and supports a sustainable future.

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 LITERATURE REVIEW

Right Stay is a global online marketplace that connects travellers with hosts offering short-term accommodations. As an alternative to traditional hotels, Right Stay allows property owners to list their spaces, enabling guests to find unique lodging options worldwide. The platform's peer-to-peer model promotes flexibility, cost-effectiveness, and local engagement. Beyond accommodations, Right Stay also offers immersive experiences hosted by locals, enhancing the overall travel experience. Despite challenges like regulatory concerns and competition, Right Stay continues to innovate, solidifying its role as a leader in the travel industry. **[1]**

One study has shown that reviews 132 peer-reviewed articles on Airbnb, highlighting that most research is recent and largely published in hospitality and tourism journals, with a focus on studies from the USA, Canada, and Europe. Content analysis categorized the research into six areas: Airbnb guests, hosts, supply and impact on destinations, regulation, tourism sector impacts, and the company itself. Consistent findings have emerged around guest motivations and listing distribution, though significant research gaps remain. This comprehensive overview provides valuable insights for industry practitioners adapting to Airbnb's impact on tourism. (DanielGuttentag,Collegeof Charleston, June 2019). **[2]**

Airbnb has also developed new software to streamline property managers' process of listing rentals on its platform automating tasks like pricing that are currently done manually. This shift aligns Airbnb more closely with the vacation rental industry and expands its presence in the B2B and SaaS sectors. Initially, the software will be available to select partners and is designed to be compatible with property management tools like Kigo. Airbnb spokesperson Christopher Nulty noted that the tool aims to broaden the company's global reach, especially in vacation markets. Airbnb has previously ventured into business travel, partnering with Concur for corporate bookings. **[3]**

One research paper has also examined whether hotel occupancy rates in Japan are affected by the number of Airbnb listings. Using seven years of data across 47 prefectures and five accommodation types, the study finds no significant impact on overall hotel occupancy. However, Ryokan and Common Lodging Houses show positive effects from Airbnb listings. The study suggests that Airbnb does not disrupt Japan's hotel industry recommends relaxing restrictions on accommodation sharing. **[4]**

According to a study found at Nagaoka University of Technology Airbnb negatively impacts hotel profitability in Japan, particularly in tourist areas, while private lodging boosts hotel profits by attracting more tourists. It recommends that local governments support private lodging and encourage collaboration between hotels and private lodging to drive regional tourism growth**. [5]**

According to an article on the medium platform. Hosts must display a notification number or business license on their listing page. Listings without this info have been deactivated to prevent new reservations.Hostswithunregisteredlistingsreceivedanemailnotification.To reactivate a listing, hosts need to update their details, including a notification number, through the host dashboard. If they have unregistered inactive listings that won't be reactivated, they can ignore the email. **[6]**

A research paper examines Indian tourists' views of Airbnb and the factors influencing their purchase intentions. A model based on planned behaviour and social exchange theory identified price, trust, and authenticity as key factors. Airbnb was found to be most comparable to homestays and mid-range hotels rather than high-end hotels. Trust and authenticity positively impact purchase intentions, while price sensitivity slightly deters it. This study fills a research gap on Airbnb's role in India's hospitality market**. [7]**

Theoretical contribution: A study contributes by addressing the gap in sharing economy technology adoption research in a developing country context, particularly within the tourism and hospitality field. To quantify these forces, data from 301 Indian consumers were collected and analysed using an extended meta-UTAUT model, integrating moderating actors of hedonic motivation (HM), trust (TR), and self-efficacy (SE). Results revealed that attitude played a mediational role in the relationship between effort expectancy, social influence, and facilitating conditions on Airbnb usage intention. The model explained 65% of the variance in consumer intention to use Airbnb in India, with performance expectancy, attitude, trust, and self-efficacy found as the significant direct determinants. **[8]**

RightStay is a global online marketplace connecting travellers with hosts for short-term accommodations, similar to Airbnb, offering unique lodging options and local experiences. Digital platforms have transformed the vacation rental industry, enabling peer-to-peer models that provide flexible, cost-effective alternatives to traditional hotels. Research highlights Airbnb's influence, especially in North America and Europe, examining aspects like guest motivations, listing distribution, and regulatory challenges. Airbnb has introduced software to help property managers automate pricing and streamline listings, expanding its reach into B2B and SaaS. Studies on Airbnb's impact on hotel occupancy show mixed results: Japan's Ryokan and smaller lodging types benefit, while hotel profitability declines in tourist areas, prompting recommendations for private lodging support. Regulatory compliance is essential, with platforms requiring hosts to provide business licenses to maintain active listings. **[9]**

1.2 PROBLEM DEFINITION

The vacation rental management/booking system is hard. It’s like operating 50 different hotel rooms spread across an entire region. Each property requires its processes, rules, and team members. Manage 5-10 different third-party apps to manage everything effectively.

A fragmented tech stack creates multiple failure points for something to get missed, which can negatively impact the guest experience (i.e., the door code didn’t get updated before their check-in).

The traditional lodging industry, encompassing both rental places and hotels, faces significant challenges in meeting the evolving demands of modern travellers. Issues such as high costs, lack of flexibility, limited accommodation options, and an impersonal guest experience often leave travellers dissatisfied. Additionally, the industry's rigid structures make it difficult for property owners, especially those with underutilized spaces, to participate in the market, limiting economic opportunities for individuals and communities.

Moreover, the environmental and social impact of large hotel chains has raised concerns about sustainability and community disruption. The over-commercialization of tourism often leads to the displacement of residents and the degradation of cultural heritage sites.

To address these issues, there is a need for a scalable business model that provides diverse, affordable, and flexible accommodation options while empowering property owners and fostering community engagement. Such a model can leverage technology to create a more personalized and sustainable travel experience, reducing the industry's environmental footprint and promoting responsible tourism. By aligning economic growth with social and environmental responsibility, this approach has the potential to transform the lodging industry, making it more inclusive, resilient, and beneficial to society as a whole

1.3 Brief Introduction of the Project

RightStay is a transformative digital platform designed to connect travellers with property owners worldwide, offering short-term accommodations that provide an alternative to traditional hotel stays. By enabling hosts to list their underutilized spaces, RightStay opens up unique, cost-effective lodging options for guests seeking flexibility, local engagement, and personalized experiences. Expanding beyond accommodations, RightStay also includes immersive local experiences, enriching travellers' journeys. Built on a peer-to-peer model, the platform fosters economic growth for property owners while promoting responsible tourism and cultural preservation. RightStay addresses the limitations of

the traditional lodging industry—such as high costs, inflexibility, and environmental impact—by utilizing technology to create a sustainable, community-cantered model that aligns with modern travel trends.

1.4 Proposed Modules:

1. User Management Module

- Handles account creation, login, and profile management for both hosts and travellers.
- Incorporates secure identity verification for added safety.
- Provides features for users to update their preferences and personalize their experience on RightStay.

2. Property Listing & Management Module

- Allows hosts to list properties with detailed descriptions, photos, and amenities.
- Offers dynamic pricing options, availability calendar, and booking settings.
- Includes a dashboard for hosts to manage multiple listings and monitor booking requests.

3. Search & Discovery Module

- Enables travellers to search for accommodations based on location, price, property type, amenities, and availability.
- Integrates filters for travellers to discover unique local experiences hosted by residents.
- Provides personalized recommendations based on past bookings and user preferences.

4. Booking & Reservation Module

- Manages real-time bookings, including instant and request-based bookings for greater flexibility.
- Ensures secure and user-friendly payment processing with options for cancellations and refunds.
- Provides booking status updates, reservation summaries, and automated confirmation messages.

5. Immersive Experiences Module

- Allows local hosts to offer unique experiences (tours, workshops, etc.) that travellers can book separately or along with accommodations.
- Includes customizable itineraries, ratings, and reviews for each experience.
- Enables hosts to set availability and manage bookings for experiences.

6. Reviews & Feedback Module

- Collects feedback from both guests and hosts to ensure transparency and improve service quality.
- Displays ratings, reviews, and testimonials for accommodations and experiences.
- Includes moderation tools to ensure the quality and authenticity of reviews.

7. Support & Communication Module

- Provides an in-app messaging system for guests and hosts to communicate securely.
- Includes a customer support interface with FAQs, chat, and contact options for dispute resolution.
- Supports multilingual options for better accessibility and global reach.

8. Admin & Analytics Module

- Provides an admin dashboard for monitoring user activity, revenue, and platform health.
- Generates data-driven insights and analytics on user behavior, booking trends, and regional performance.
- Includes tools for content moderation, compliance monitoring, and regulatory reporting.

9. Environmental & Social Impact Module

- Encourages sustainable practices by highlighting eco-friendly accommodations and experiences.
- Provides data on the platform's environmental impact and promotes responsible tourism initiatives.
- Features community engagement tools that allow hosts to support local businesses and charities.

Together these modules aim to create a comprehensive and user-friendly platform, making RightStay an inclusive and sustainable alternative in the vacation rental industry.

1.4.1 Technological Innovation:

- **User Experience:** Studies highlight that user interface (UI) and user experience (UX) design are critical in attracting and retaining users. A well-designed platform enhances user engagement and satisfaction (Smith et al., 2021). For instance, Airbnb's intuitive interface and responsive design have been pivotal in its success (Johnson & Lee, 2019).
- **Data Analytics:** The integration of data analytics allows platforms to offer personalized recommendations and optimize pricing strategies (Kumar et al., 2020). Machine learning algorithms, for instance, can predict user preferences and adjust search results accordingly.

1.4.2 Market Dynamics:

- **Consumer Behaviour**: Research indicates a growing preference for personalized and unique travel experiences, which platforms like Airbnb effectively cater to by offering diverse accommodation options (Chen & Huang, 2018). The shift towards experiential travel is driving demand for innovative features.
- **Competitive Advantage:** Effective use of technology can provide a significant competitive edge. Features such as real-time booking, secure payment systems, and interactive maps contribute to a platform's success and user loyalty (Davis et al., 2022).

1.5 Hardware & Software Requirements

Hardware Requirements: -

1. Development Machine:

- **CPU:** Multi-core processor (Intel i5/Ryzen 5 or higher recommended).
- **RAM:** Minimum 8 GB (16 GB or more preferred).
- **Storage:** SSD for faster read/write speeds (at least 256 GB).
- **Network:** Stable internet connection for development and testing.

2. Server:

- **Cloud Hosting:** AWS, Google Cloud, or Digital Ocean (choose based on scale).
- **Configuration:** At least 2 vCPUs, 4 GB RAM, and 50 GB SSD for initial deployment (scale as needed).

3. Testing Devices:

Mobile devices (iOS and Android) are used to test the application across different platforms.

Software Requirements: -

1. Frontend:

- **Languages:** HTML, CSS, JavaScript.
- **Frameworks/Libraries:** React, Angular, or Vue.js.
- **Build Tools:** Webpack, Babel, or similar.

2. Backend:

- **Languages:** Node.js, Python (Django/Flask), Ruby on Rails, or PHP (Laravel).
- **Frameworks:** Express.js (for Node.js), Django/Flask (for Python), or Rails (for Ruby).

3. Database:

- **SQL Databases:** PostgreSQL, MySQL.
- **NoSQL Databases:** MongoDB (if needed).

4. APIs:

- RESTful API or GraphQL for communication between frontend and backend.
- Payment Gateway APIs (Stripe, PayPal, etc.).
- Third-party APIs for maps (Google Maps API), messaging, etc.

5. DevOps Tools:

- **Version Control:** Git and platforms like GitHub or GitLab.
- **Containerization:** Docker (optional but recommended for deployment).
- **CI/CD:** Jenkins, GitHub Actions, or GitLab CI for automated deployments.

6. Project Management:

- Tools like Jira, Trello, or Asana for task management.
- Design tools like Figma or Adobe XD for UI/UX design.

Other Tools: -

- **Authentication:** OAuth, and JWT for user authentication and authorization.
- **Hosting:** Nginx or Apache for serving web applications.
- **Monitoring:** Tools like New Relic, Sentry, or LogRocket for performance monitoring and error tracking.

Additional Considerations: -

- **Legal Compliance:** Ensure your project complies with local regulations regarding rentals.
- **Security:** Implement HTTPS, data encryption, and secure coding practices.
- **Testing:** Unit testing and end-to-end testing frameworks (Jest, Cypress, etc.)

CHAPTER-2
SYSTEMS ANALYSIS AND SPECIFICATION

2.1 Functional Model: -

Under our proposed methodology, this application has two larger consumers. Another half is the user from the hotel side and another one is the consumer who wants to book from outside.

The more relevant terms would be Onboarding, Update/check-in, and Bookings as viewed from a hotel-side user. Similarly on the other side (External User (consumer)) perspective, the common would be as part of Search, Bookings, Analytics, and non-functional requirements such as: -

- Low Latency
- High availability
- High consistency
- Scale

Now, let's consider a more high-level view of the system design and the function of the model within the system.

LB (Load Balancer): — A Load balancer sits between the client and the server, receiving the client's request and then forwarding it to the server which can serve the client's request.

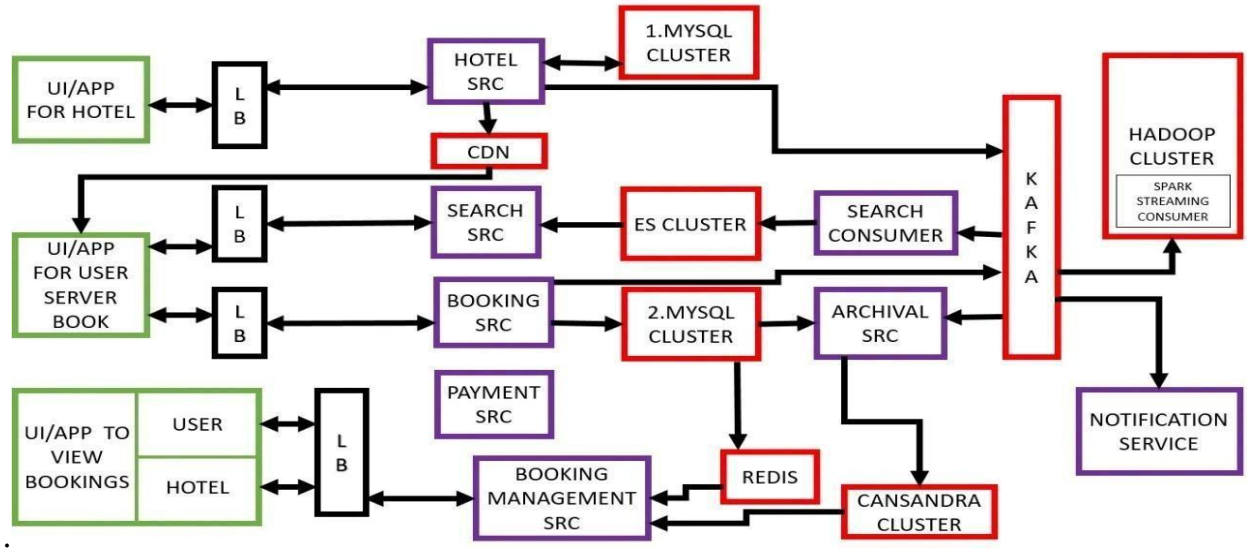


Figure 2.1 **Functional Model**

Understanding how the booking service works?

1. Check-in available rooms

2. Insert in booking and then reduce the available rooms.

3. Put in REDIS

4. Put in KAFKA

Redirect to payment.

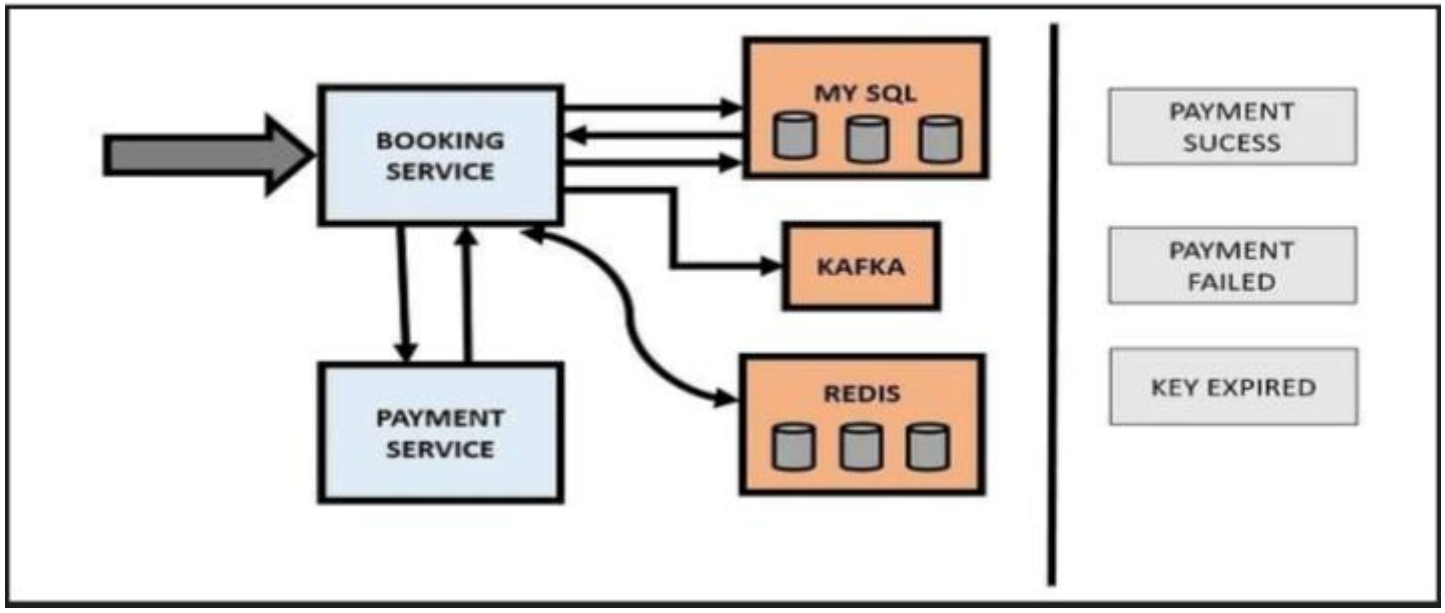


Figure 2.2 **Booking Service Working system**

2 .2 Data Model

2.2.1 DFD (Data Flow Diagram)

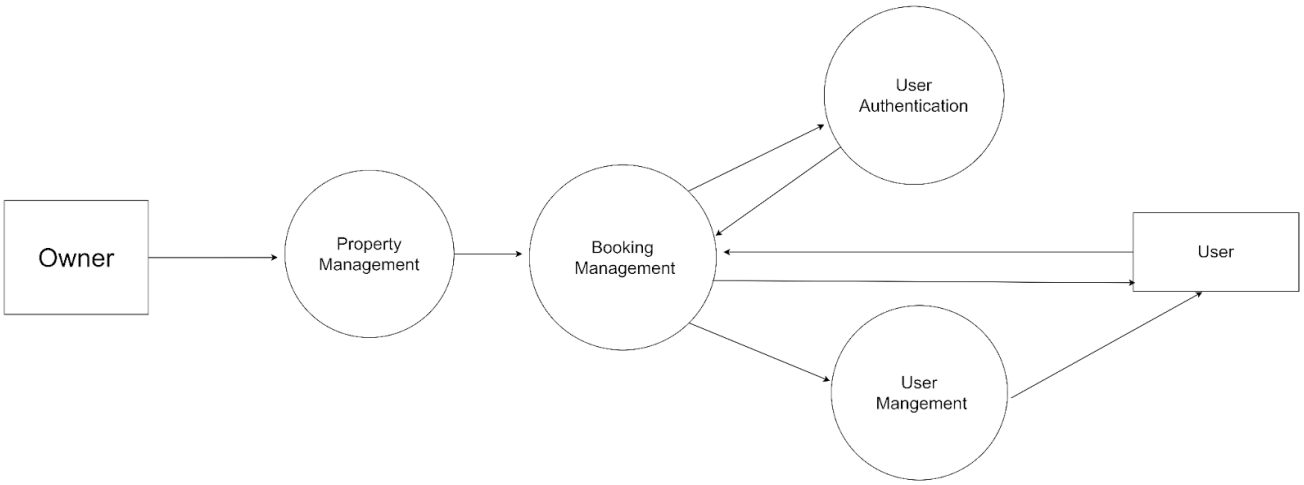
Figure 2.3 **DFD level-0**

The diagram below is the highest level that you can go to see how external entities (people interacting with the system) interact with the system:

Property Owner: Can list properties on the RightStay system.

User: Can search for and book properties through the system.

RightStay System: Acts as the central platform, allowing property owners to list properties and users to book them. When a user books a property, they receive a booking confirmation from the system.



. Figure 2.4 **DFD level-1**

At this level, we break down the RightStay system into smaller components or subsystems. This is where we see the core parts of the system and how the external entities interact with them:

Property Owner interacts with the Property Management component to add or update properties.

A user interacts with the Booking Management component to search for properties and make bookings.

The User Management component is responsible for authenticating (verifying) users and managing their accounts.

Booking Management interacts with User Management to ensure only logged-in or authenticated users can book properties.

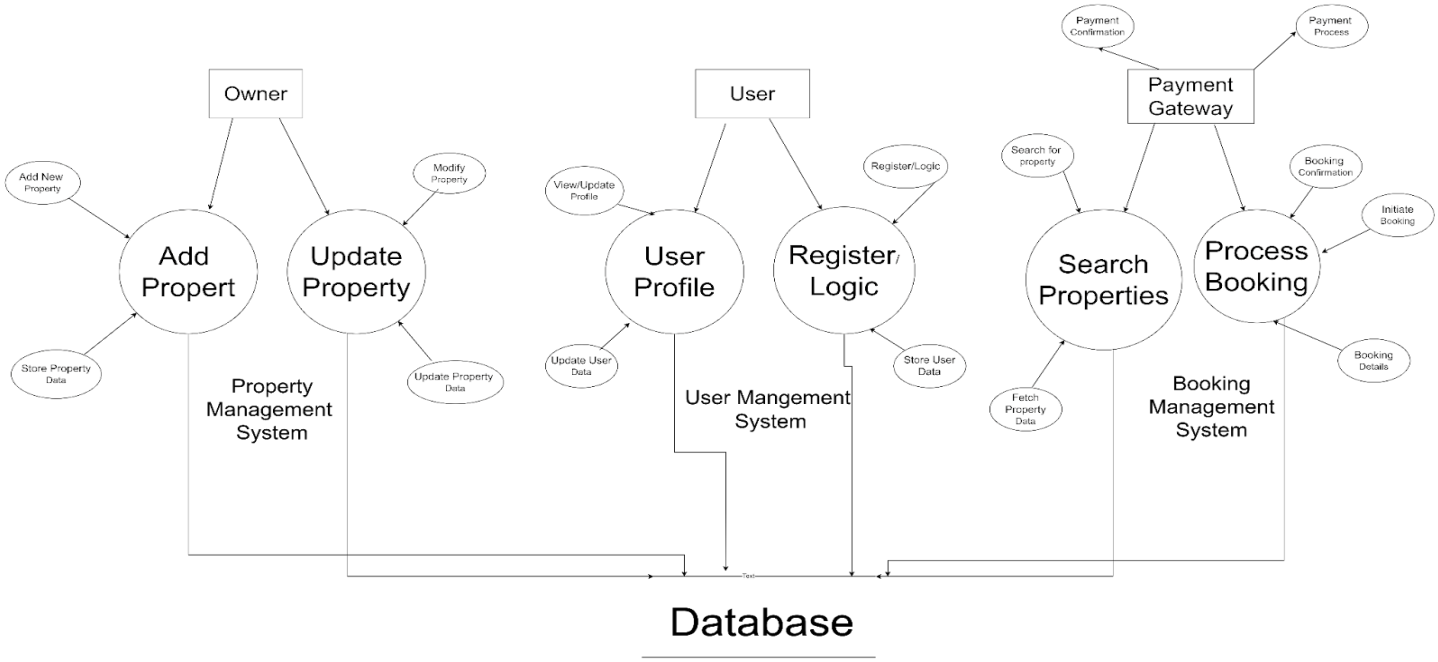


Figure 2.5 **DFD level-2**

1. Property Management Module

- **Add Property:** The property owner can submit details to list a new property. The system saves this data into a database.

- **Update Property:** The owner can update details (like price, availability, etc.) of their listed properties, which also get updated in the database.

2. Booking Management Module

- **Search Properties:** A user can search for properties based on filters (e.g., location, dates). The system retrieves the relevant data from the database to show the available properties.
- **Process Booking:** When a user finds a property and wants to book it, they initiate the booking process. The system checks availability, processes the booking details, and asks for payment through an external Payment Gateway.

3. User Management Module

- **Register/Login:** Users can create accounts or login to existing accounts. This part of the system stores their login credentials and profile information in a database.

User Profile: Users can view or update their profile details, such as changing their name or contact information. Any updates are stored in the database.

2.2.2 Entity-Relationship Model:-

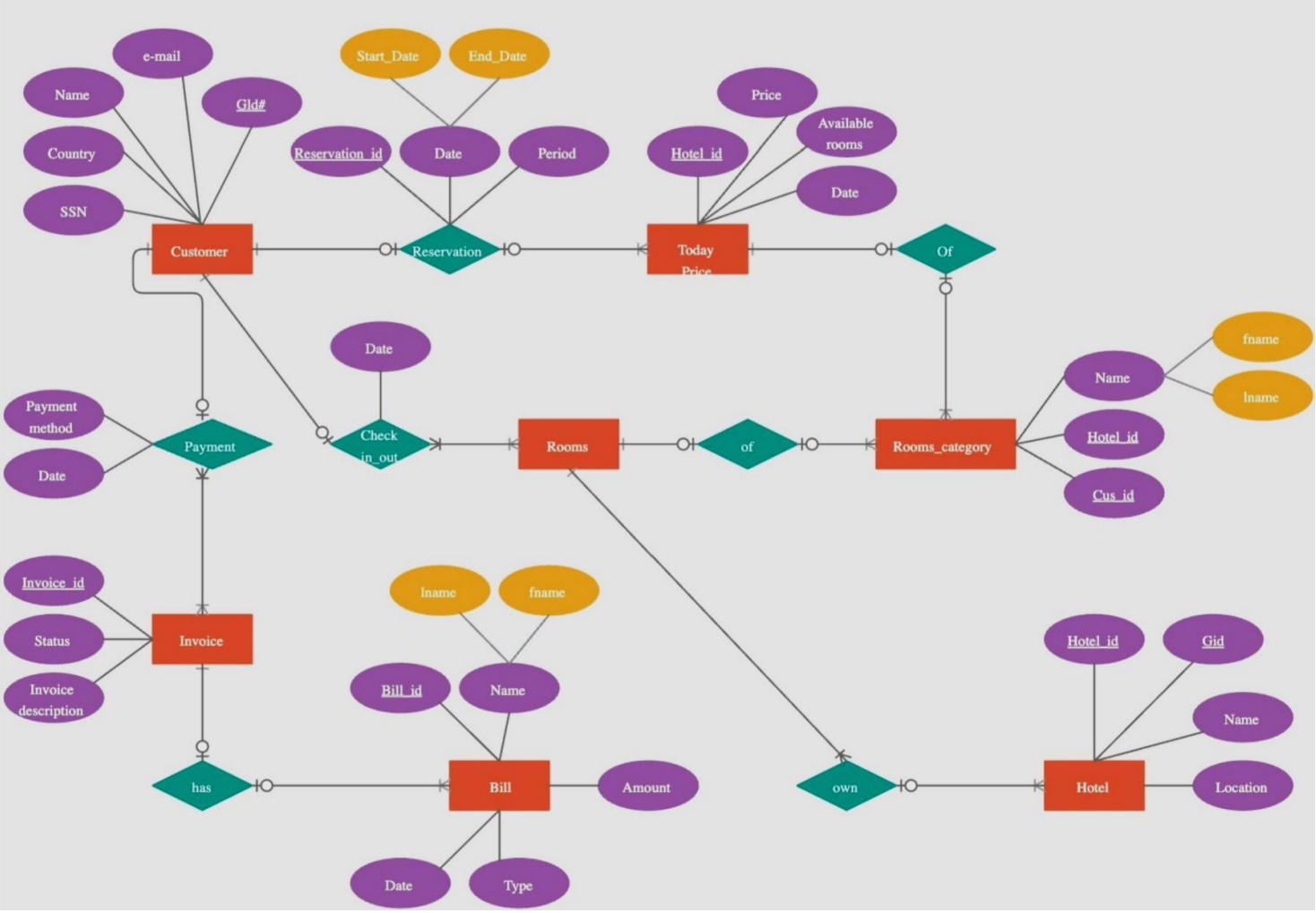


Figure 2.6 **ER-Diagram**

1. Entities and Their Attributes:

In the above diagram, Entities are the core objects represented as rectangles in the diagram. Each entity has specific attributes (shown in ovals) that provide detailed information about it.

Customer:

- Attributes: Name, Email, Country, SSN (Social Security Number), GId# (likely a unique guest identifier).
- Represents the guest staying or booking the hotel.

Reservation:

- Attributes: Start Date, End Date, Reservation.
- Captures the reservation details, linking the customer with the hotel stay period.

Today's Price:

- Attributes: Price, Available rooms, Date, Hotel_id.
- Tracks room availability and pricing based on the day for specific hotels.

Rooms Category:

- Attributes: Name, Hotel_id, Cus_id.
- Represents categories or types of rooms, such as Deluxe, Standard, or Suites.

Hotel:

- Attributes: Hotel_id, Gid, Name, Location.
- Store details about the hotels in the system, including their location.

Invoice:

- Attributes: Invoice_id, Status, Invoice description.
- Represents the billing document for the customer's stay.

Bill:

- Attributes: Bill_id, Date, Type, Amount, Name.
- Tracks the billing details generated for each reservation, including the total amount and date.

2. Relationships (Diamond shapes):

These Relationships describe the connection between the entities.

Reservation (Customer Reservation):

- Shows that a customer makes one or more reservations.

Check-in / out (room reservation):

- Links rooms to reservations, specifying when a room is booked or vacated.

Payment (Customer Invoice):

- Describes the payment relationship, where a customer makes payments linked to invoices.

Off (Today_Price – Rooms_Category):

- Connects daily room prices with room categories.

Own (Hotel – Rooms_Category):

- Indicates that hotels own or manage multiple room categories.

Has (Invoice – Bill):

- An invoice contains detailed billing information linked to the customer's stay.

3. Cardinality and Connectivity: The diagram uses symbols like lines with small circles or perpendicular bars to define the nature of connections between entities.

- A circle (O) indicates that participation in the relationship is optional.
- A line with a bar (|) means mandatory participation.

For example:

A. A customer can make zero or many reservations, as indicated by the circle and bar on either side of the Reservation relationship.

B. A room category must belong to a hotel, as shown by the mandatory (bar) connection with the Hotel.

4. How the System Works:

1. Customers make reservations with a hotel, specifying the start and end dates.
2. The rooms are categorized, and the system records their price for each day.
3. When customers check in or out, the system updates room availability accordingly.
4. After the stay, an invoice is generated, and customers pay the amount as specified.

2.2.3 Class diagram

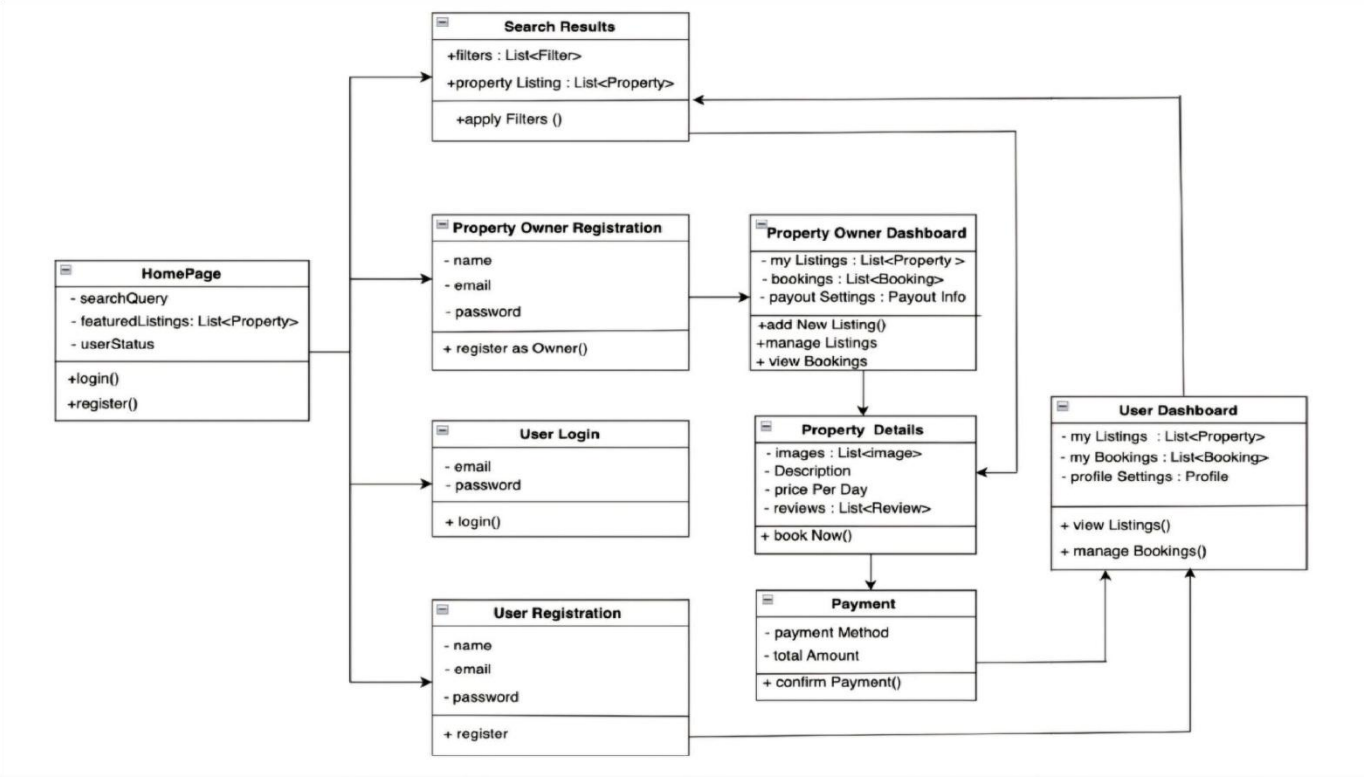


Figure 2.7 **Class Diagram**

This class diagram for "RightStay" shows different components and how they interact within the system. Here's a simple explanation of each class:

1. Homepage

a. Attributes:

- **search Query:** The user can type in a search query (e.g., location, property type).
- **featured listings:** A list of featured properties displayed on the homepage.
- **user Status:** Indicates if a user is logged in or not.

b. Methods:

- **login ():** Allows users to log into the platform.
- **Register ():** Allows new users or property owners to register.

2. Property Owner Registration

a. Attributes:

- **name:** The owner's name.
- **email:** The owner's email.
- **password:** The owner's password for account access.

b. Methods:

- **registerAsOwner ():** Allows property owners to register and list properties.

3. User Registration

a. Attributes:

- **name:** The user's name.
- **email:** The user's email.
- **password:** The user's password for account access.

b. Methods:

- **register ():** Allows regular users to create an account to book properties.

4. User Login

a. Attributes:

- **email:** The user's email used for login.
- **password:** The user's password.

b. Methods:

- **login ():** Allows existing users to log into the platform.

5. Search Results

a. Attributes:

- **filters:** A list of filters (e.g., price, location, etc.) used for searching.
- **property listings:** A list of properties matching the search criteria.

b. Methods:

- **applyFilters (): A**pplies the selected filters to refine the search.

6. Property Owner Dashboard

a. Attributes:

- **myListings:** A list of properties owned by the owner.
- **bookings:** A list of bookings for the owner's properties.
- **payoutSettings:** Settings related to how the owner gets paid.

b. Methods:

- **addNewListing ():** Allows the owner to add a new property listing.
- **manageListings ():** Manage the properties (edit, update, or delete listings).
- **viewBookings ():** View the bookings for the owner's properties.

7. Property details

a. Attributes:

- **images:** A list of images showing the property.
- **description:** The description of the property.
- **pricePerNight:** The nightly price of the property.

8. Methods:

- **bookNow ():** Allows the user to book the property.

a. Attributes:

- **paymentMethod:** The method the user selects to pay (credit card, PayPal, etc.).
- **totalAmount:** The total amount to be paid for the booking.

b. Methods:

- **confirmPayment ():** Confirms and processes the payment.

9. User Dashboard

a. Attributes:

- **myListings:** A list of properties listed by the user (if they are a property owner).
- **myBookings:** A list of properties the user has booked.
- **profileSettings:** Settings related to the user's profile (e.g. Use Case Diagram)

2.3. A process-flow Model

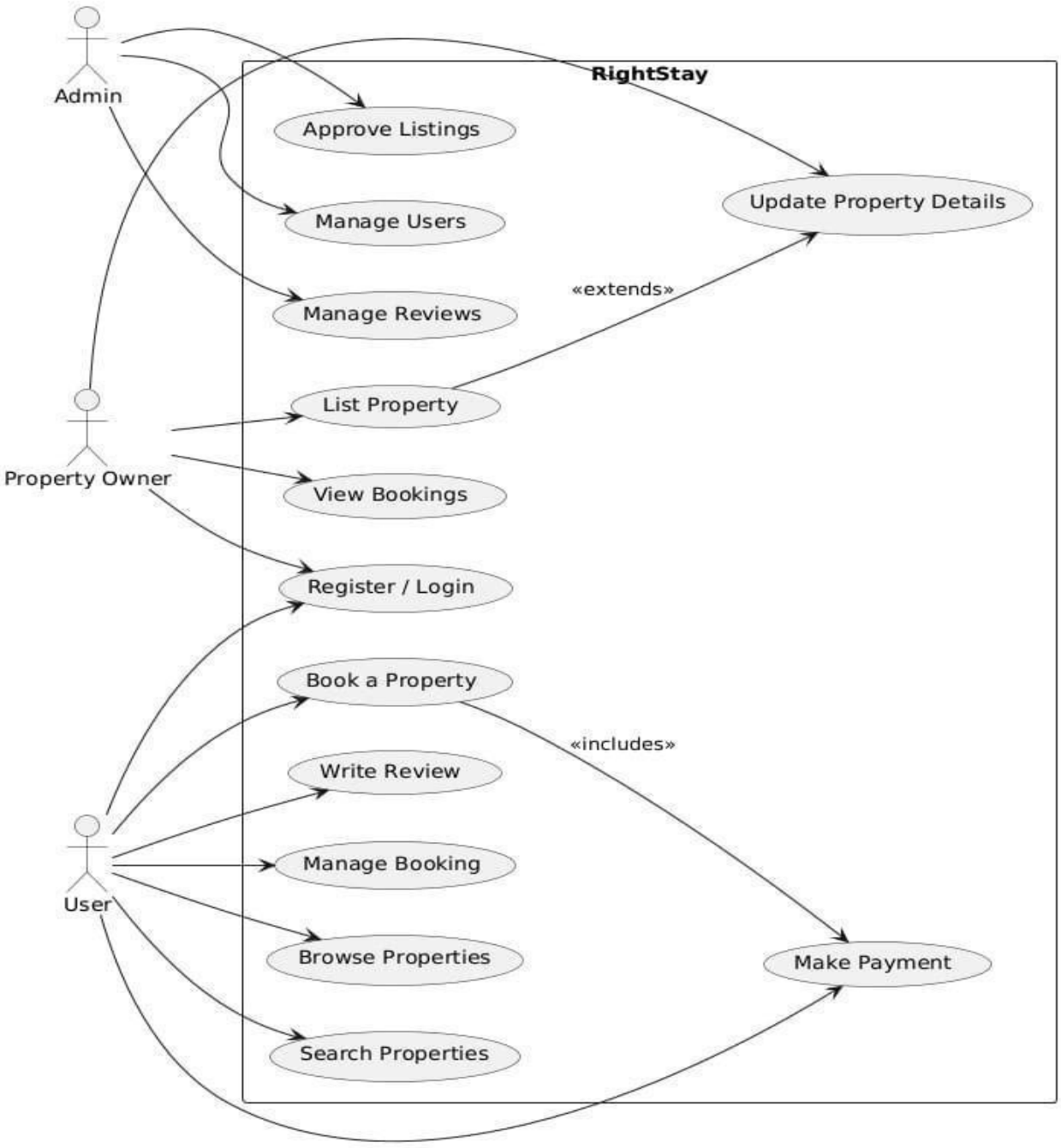


Figure 2.8 **Use Case Diagram**

Actors:

- **User:** Can register/login, browse properties, search, book, make payments, write reviews, and manage their bookings.
- **Property Owner:** Can register/log in, list properties, update property details, and view bookings.
- **Admin:** Manages listings, users, and reviews.

Use cases:

- User actions such as browsing, booking, and payment.
- Property owner actions such as listing and managing properties.
- Admin actions for overseeing the platform.

2.4 Activity diagram

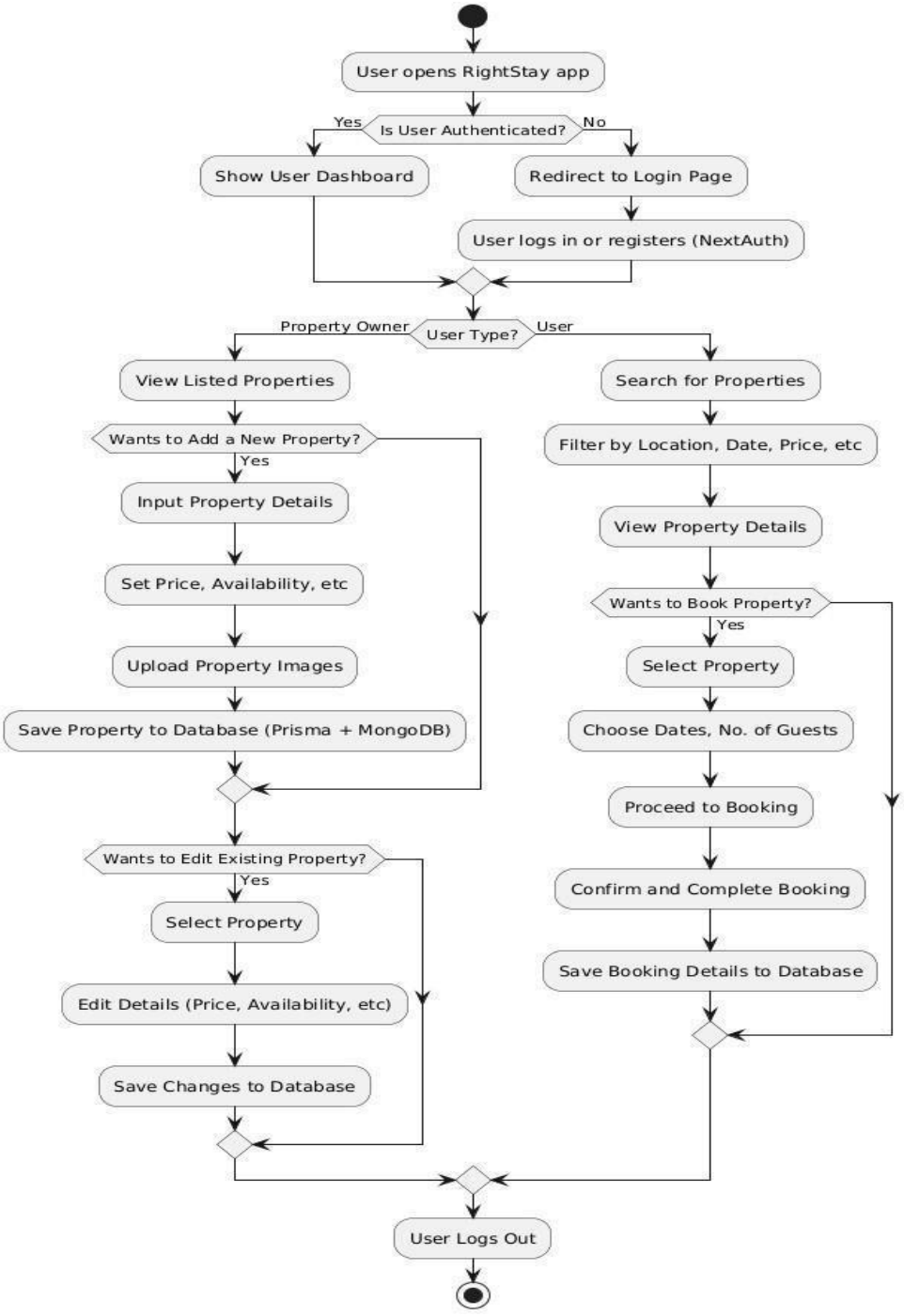


Figure 2.9 **Activity diagram**

The activity diagram you provided illustrates the various workflows within the RightStay platform, involving different actors such as the property owner, admin, user, and payment gateway.

Property Owner Activities:

Login to RightStay:

- The property owner logs into the RightStay platform using their credentials.

List New Property:

- The owner starts the process of listing a new property
- providing property details such as location, size, and amenities.

Upload Photos and Set Pricing:

- The owner uploads relevant images of the property and specifies the rental price.

Property Validation:

- Before submitting the listing, the property is either validated automatically or a validation request is sent to the admin.
- If the validation fails (e.g., missing information), the owner receives an error or update request and needs to correct and resubmit the details.

Submit Property for Approval:

- Once validated, the property is sent for admin approval, and the owner waits for confirmation.

Property Listed Successfully:

- After approval, the property is listed successfully on the platform.

Admin Activities:

Login to Admin Dashboard:

The admin logs into the dashboard to manage property submissions.

Review Property Submission:

The admin reviews the property submitted by the owner, ensuring that all details are complete and valid.

Approve or Request Update:

If the property submission is complete, the admin approves it; if not, the admin sends a validation request back to the owner to correct the property details.

Approved Property Published:

Once approved, the property is published on the platform for users to view and book.

User Activities:

Browse Available Properties:

- The user browses the available properties on the RightStay platform.

Filter by Location, Date, and Price:

- Users can filter properties based on their preferences such as location, dates, and budget.

Select Property:

- Once a suitable property is found, the user selects the property for booking.

View Property Details:

- The user can view detailed information about the selected property.

Book Property:

- After reviewing the property details, the user proceeds to book the property.

Confirm Payment:

- The user confirms payment through the platform.

Receive Booking Confirmation:

- Upon successful payment, the user receives a booking confirmation.

Payment Gateway Activities:

Process User Payment:

- The payment gateway processes the payment request from the user.

Payment Success or Failure:

- If the payment is successful, a confirmation is sent to RightStay. If the payment fails, a failure notice is sent to the user.

End of Workflow:

Booking Confirmed:

- After the payment is confirmed, the booking is finalized for the user.

2.5 Sequence Diagram

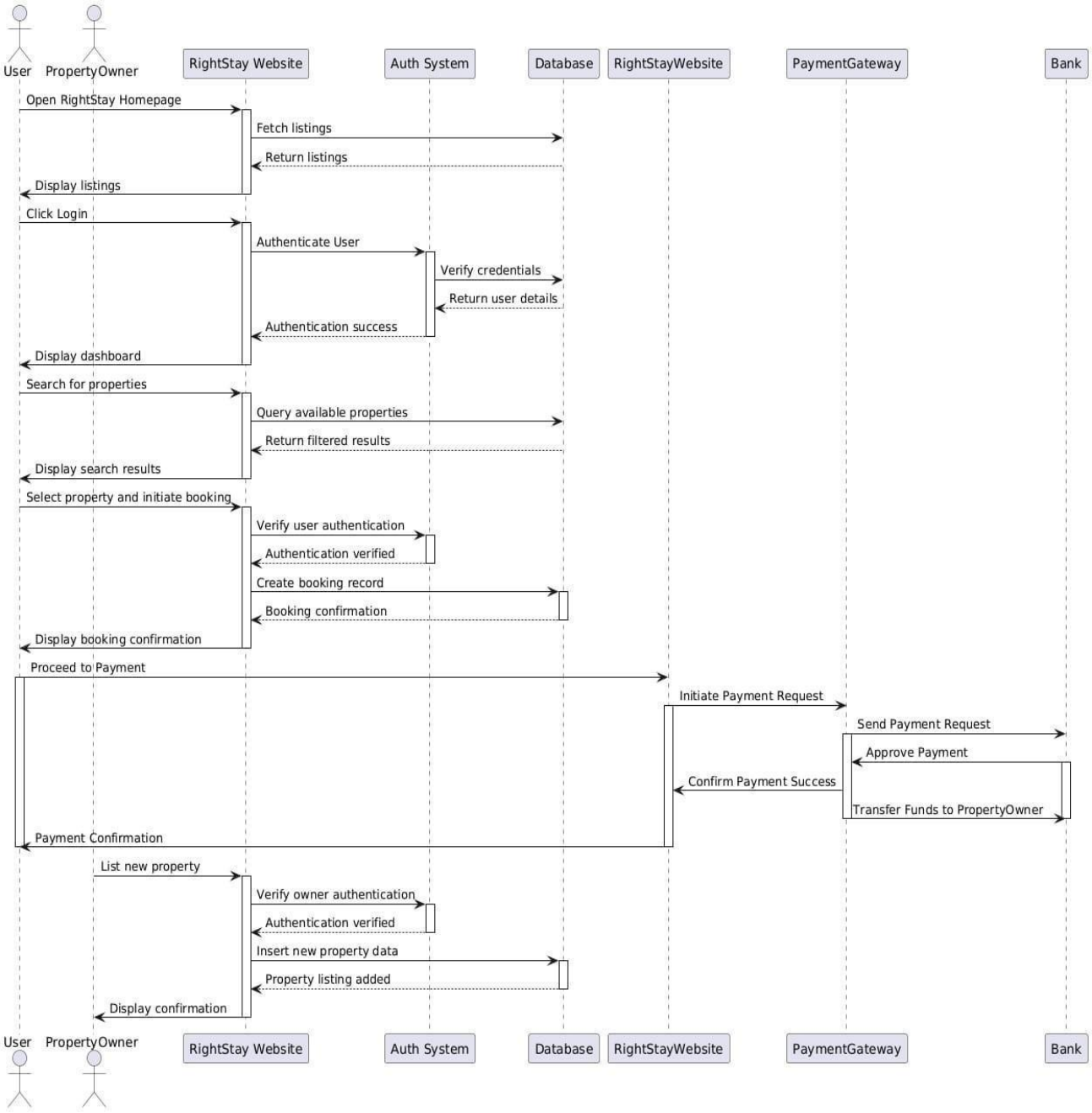


Figure 2.10 **Sequence Diagram**

The provided sequence diagram represents the flow of events for a user and a property owner interacting with the RightStay website for booking and listing properties.

User Journey:

1. **Opening the Website:**

   The user opens the RightStay homepage, which triggers the system to fetch property listings from the database, and then the available listings are displayed to the user.

2. **Login:**

   When the user clicks the "Login" button, the system sends a request to the authentication system to verify the user's credentials. Once the credentials are validated, the authentication system sends the user details back to the RightStay website, and the user is shown their dashboard.

3. **Searching for Properties:**

   The user can search for specific properties. This action triggers the RightStay system to query the database for properties that match the user's search criteria, and the filtered results are displayed.

4. **Booking a Property**:

   Once the user selects a property to book, the system verifies their authentication and then creates a booking record in the database. The user receives confirmation of the successful booking.

5. **Proceeding to Payment:**

   After the booking is confirmed, the user proceeds to the payment stage. The RightStay system sends a payment request to the payment gateway, which forwards the request to the bank. Once the payment is approved, the funds are transferred to the property owner's account. The user is then notified of the successful payment.

Property Owner Journey:

1. Listing a New Property:

A property owner can list a new property on the RightStay website. After logging in, their credentials are verified by the authentication system. Once authenticated, they submit the new property data, which is saved in the database. A confirmation is displayed to the property owner that their listing has been successfully added.

2.6. A behavioral model

1. User Roles and Actions

- **Guest (User)**
    - ○ Applies filters (e.g., location, price, amenities).
    - ○ Views property details.
    - ○ Creates an account or logs in.
    - ○ Book a stay.
    - ○ Make payments.
    - ○ Views booking history.
    - ○ Leaves reviews and ratings for stays.
    - ○ Messages host for inquiries.
- **Host**
    - ○ Registers as a host.
    - ○ Adds or manages property listings.
    - ○ Sets property availability and prices.
    - ○ Views booking requests and confirms/rejects bookings.
    - ○ Manages bookings (e.g., cancellation policies).
    - ○ Communicates with guests.
    - ○ Responds to guest reviews.
- **Admin**
    - ○ Monitors user activity.
    - ○ Manages user accounts and listings.
    - ○ Resolves disputes between hosts and guests.
    - ○ Reviews flagged listings and reviews.
    - ○ Manages platform settings (e.g., fees, content guidelines).
    - ○ Generates reports on platform activity.

2. User Journey and Behaviour

- **Guest Browsing:**

  Guests access the platform and use search filters to find properties. When they apply filters, the app fetches relevant results in real time.

- **Property Viewing** :

  Guests click on properties to view detailed information (images, descriptions, amenities, etc.). The system tracks views and engages the user with similar properties to increase engagement.

- **Booking Process** :

  When guests proceed to book, they are prompted to log in or sign up if they haven't already.

  A booking confirmation page reviews the total cost and prompts payment.

- **Payment** :

  Payments are processed through a secure payment gateway. The system updates the booking status upon successful payment.

- **Reviewing Experience** :

  Guests can rate and review their stay after check-out. Hosts are notified of new reviews and can respond.

3. System Behaviours and Interactions

- **Real-Time Availability Check** :The system checks property availability for requested dates in real time to avoid double booking.
- **Message Notifications** :Hosts and guests receive notifications for new messages, inquiries, or booking updates
- **Recommendation Engine** :Based on user browsing and booking history, the platform recommends similar properties.
- **Booking Confirmation and Cancellation** : Hosts confirm bookings, and the system updates the status accordingly. Cancellation options depend on the host’s policies, with fees or refunds calculated dynamically.
- **Dispute Resolution** : Admins monitor flagged reviews or disputes between guests and hosts.

4. Behavioural Scenarios

- **Scenario 1: Guest Searching and Booking a Stay**
  - *Input:* Guest searches for properties with specific filters.
    - *Behavior*: The system retrieves listings based on filter criteria and shows relevant properties.
    - *Output:* The guest selects a property, views detail, books the stay, and completes the payment.
- **Scenario 2: Host Adding a New Property**
  - *Input:* Host submits property details, images, and availability dates.
    - *Behavior*: The system validates inputs, stores data, and publishes the property for guests to see.
    - *Output:* Property becomes searchable; the host receives booking notifications when guests book.
- **Scenario 3: Admin Reviewing a Dispute**
  - *Input:* A dispute is flagged, prompting admin action.
    - *Behaviour* : Admin reviews messages, bookings, and past interactions, then contacts both parties for clarification.
      - *Output:* Admin resolves the dispute and updates account statuses if needed.

5. User Interface Responses

- **Guest UI:** Responsive property lists, real-time updates on bookings, payment verification, and review prompts post-stay.
- **Host UI:** Dashboard with property and booking management, notification center for guest interactions.
- **Admin UI:** Tools for monitoring user activity, managing listings, handling disputes, and generating reports.

2.7. System Design

2.7.1 Technical Feasibility

"RightStay" is feasible from a technical standpoint due to several factors:

- **Technology stack:** The use of modern technologies, including React for the front and Node.js/Express for the backend, ensures a stable foundation for development.
- **Cloud infrastructure:** Hosting on cloud services such as AWS or Google Cloud allows the system to scale easily with user demand, ensuring the platform remains responsive even during traffic spikes.
- **AI integration:** Features like personalized suggestions and dynamic pricing will be implemented using accessible machine learning tools like TensorFlow and PyTorch.
- **Third-party APIs:** By integrating third-party APIs for payments (Stripe, PayPal), messaging (Twilio), and mapping (Google Maps), the system can focus on core functionalities without needing to build these features from scratch.
- **Scalability:** With a microservices architecture, each service can scale independently, ensuring smooth operations regardless of traffic fluctuations.

2.7.2 Operational Feasibility

From an operational perspective, "RightStay" is designed for smooth functionality:

- **User-friendly interface:** The platform is built for ease of use, ensuring that even non-technical users can navigate the system. Hosts and guests will be able to complete their tasks with minimal effort, from listing properties to booking accommodations.
- **Automation:** Processes such as booking management, payments, and communication between users are automated, reducing manual intervention. AI-driven systems will help match guests with properties, while support features like chatbots and FAQ sections will handle common customer inquiries.
- **Operational efficiency:** The system minimizes operational overhead by using automated solutions for managing listings, booking confirmations, and payments. Moreover, AI-based algorithms will drive the recommendation engine, boosting both host visibility and guest satisfaction.
- **Security and risk managemen**t: Key features like data encryption, user verification, and built-in fraud detection ensure the platform is secure. Additionally, insurance options can be offered to hosts and guests to protect against damages and losses.

Operationally, the system is built to handle a large user base and scale efficiently as demand grows.

2.7.3. Economic Feasibility

The economic feasibility of "RightStay" shows that it is cost-effective to implement and can become profitable:

- **Development Costs:** Initial investment will be needed for infrastructure (cloud services, development teams, software licenses, and security measures). However, using scalable cloud platforms and open-source software reduces significant upfront costs.
- **Revenue Generation:** Several streams of revenue are planned, including:
- **Booking fees:** A small percentage of the booking fee will be charged to both hosts and guests on every successful transaction.
- **Subscription models:** Hosts will have access to premium services like enhanced property visibility, advanced data analytics, and marketing tools through a subscription-based service.
- **Advertisement:** The platform can monetize by offering advertising slots for property-related services (insurance, cleaning services, etc.).
- **Operating Costs:** Ongoing expenses, such as cloud infrastructure, API usage (for payments, notifications, etc.), and customer support, will be carefully managed through automation and cloud optimization strategies.
- **Scalability:** The economic model supports scaling. As the user base grows, operational costs will not increase linearly, enabling the platform to generate higher profits without significantly increased expenses.

Results

1) **Increased Market Share**:

   Growth Metrics:Platforms incorporating advanced technology have seen substantial growth in user base and market share. For instance, Airbnb's use of real-time analytics and personalized recommendations has contributed to its dominant position in the market (Wang & Zhang, 2021).

2) **Enhanced User Satisfaction**:

   Feedback Analysis:User feedback consistently highlights the importance of seamless interactions and reliable customer support. Platforms that address these needs through technological innovations and responsive design achieve higher satisfaction ratings (Martin et al., 2023).

3) **Operational Efficiency:**

   Automation and Optimization:Automation of backend processes and optimization of search algorithms have led to increased operational efficiency. Platforms that implement these strategies effectively manage large volumes of data and transactions (Singh&Patel,2022).

CHAPTER 3
MODULE IMPLEMENTATION & SYSTEM INTEGRATION

The Module Implementation & System Integration phase marks a critical step in the development process, where individual software modules—designed and developed separately—are integrated into a unified system. In this phase, each module is first implemented based on its specifications, tested independently for functionality, and then systematically combined to work together as a cohesive system. This integration ensures that modules interact correctly through defined interfaces and data flows.

During integration, compatibility, data consistency, and communication between components are validated to prevent system failures. The goal is to create a fully functional, efficient, and reliable system from independently developed parts. This phase also includes debugging, resolving inter-module issues, and refining performance to ensure the system meets overall project objectives and user requirements

3.1 Module Implementation

The project was divided into several interdependent modules to ensure modularity, scalability, and ease of maintenance. Each module was developed independently following the design specifications and then tested individually to ensure correct functionality. Each part of the system was broken down into well-defined, manageable modules, each handling a specific function. This modular approach not only simplifies development but also makes the system easier to test, debug, and maintain. The system is composed of several core modules, each handling a specific aspect of the application. These were developed independently and later integrated to form a fully functional system. The design follows a modular architecture, ensuring flexibility, scalability, and maintainability.

A. User Interface module

This module is responsible for user interaction. It includes all the screens, forms, and navigation flows. The frontend was implemented using [mention tools/languages – e.g., HTML, CSS, JavaScript, React, etc.]. It ensures a responsive and user-friendly experience.

- **Role:** Serves as the main interaction layer between the user (student/teacher/admin) and the system.

- **Functions:**

a. Clean and intuitive dashboard for easy navigation.
b. Interfaces for login, attendance tracking, notifications, and reporting.
c. Real-time camera feed for face recognition.

- **Tools & Technologies:**

a. Built using React Native or Flutter for mobile responsiveness.
b. Web version may use HTML/CSS/JavaScript.

- **UX Focus**:

a. Minimal steps to complete attendance.
b. Feedback alerts for attendance success/failure.

2. Authentication Module

- **Role:** Controls user access and ensures only authorized users can perform specific operations.

- **Functions:**

a. User roles: Admin, Teacher, Student.
b. Secure login via encrypted credentials.
c. Face recognition acts as biometric authentication for students.

- **Tech Stack:**

a. JWT (JSON Web Token) for secure session management.
b. Firebase Authentication or OAuth 2.0 for third-party integration.

3. Database Module

- **Role**: Stores and manages all system data reliably.

- **Stored Data Includes:**

a. User profiles and roles.
b. Attendance logs (face match, timestamp, location).
c. Face embeddings extracted via machine learning.
d. Geolocation data per attendance entry.

- **Technology:**

a. MySQL or MongoDB for structured/unstructured data.
b. Uses foreign key constraints for relational integrity.

4. Backend / API Module

- **Role:** Implements business logic and serves data to the front-end.

- **Functions**:

a. API endpoints for registration, login, attendance submission, report generation.
b. Processes face images using ML model.
c. Cross-verifies face recognition and geo-location before confirming attendance.

- **Frameworks:**

a. Python (Flask/Django) or Node.js.
b. Integration with OpenCV, Dlib, or DeepFace libraries for facial recognition.

5. Admin Panel Module

- **Role:** Enables admin-level monitoring and control over the system.

- **Functions:**

a. Add/remove users.

b. View attendance reports across students, departments, or time ranges.

c. Export attendance data in formats like PDF/CSV.

- **Design:**

a. Web dashboard with responsive UI and data visualization (using Chart.js or D3.js).
b. Role-based access for security.

6. Notification Module

- **Role:** Keeps users informed through alerts and status updates.

- **Functions:**

a. Sends reminders to students to mark attendance.
b. Notifies teachers and admins of low attendance or anomalies.

- **Technology:**

a. Firebase Cloud Messaging (FCM) or Twilio for real-time messaging.

b. Event-based triggers linked to backend logic.

3.2 SYSTEM INTEGRATION

After independent development, modules are connected and tested as a single system. Integration ensures smooth communication and proper data flow across all components.

1. Frontend & Backend Integration

- **How It Works:**
    - **a.** The mobile or web frontend sends API requests to the backend.
    - **b.** Backend validates the data, performs necessary processing, and returns responses.
- **Example:**
    - **a.** When a student clicks “Mark Attendance,” the app sends a request with the captured face image and GPS coordinates to the backend API.

2. Face Recognition System Integration

- **Workflow:**
    - **a.** Uses live camera feed or image upload.
    - **b.** Image is preprocessed (resized, normalized).
    - **c.** Facial embeddings are extracted using ML models.
    - **d.** Compared with stored embeddings in the database using cosine similarity or Euclidean distance.
- **Success Threshold:**
    - a. A match score above a set threshold confirms identity.

3. Geo-Fencing System Integration

- **Workflow:**
    - **a.** Student's device captures GPS coordinates.

b. Backend checks if coordinates fall within a predefined classroom radius.

c. Attendance is only confirmed if face recognition and geo-fencing checks pass.

- **Libraries/Tools:**

a. Google Maps API or native location services (Android/iOS).

4. Database Integration

- **Flow:**

a. Backend queries or updates data via ORM or SQL queries.

b. Data consistency is maintained using transaction handling and indexing.

c. Time-stamped entries are stored with both image and location metadata.

- **Security:**

a. Passwords encrypted.

b. Audit logs maintained for admin actions.

5. Admin Dashboard Integration

- **Data Flow:**

a. Frontend dashboard fetches real-time reports from backend APIs.

b. Filters and charts are dynamically populated based on date, student, or department.

- **Features:**

a. Admin can trigger notifications or manually edit records

6. **Notification System Integration**

- **Trigger-Based System:**

    a. Specific events like attendance failure, new announcements, or reminders activate the notification system.

- **Communication Channels:**

    a. SMS, Email, In-App Push notifications.
    b. Integration Tools:
    c. Backend invokes services like FCM or Twilio through secure APIs.

CHAPTER 4
TESTING AND EVALUATION

4.1 Testing

4.1.1 Testing Samples:-

- **Home page**

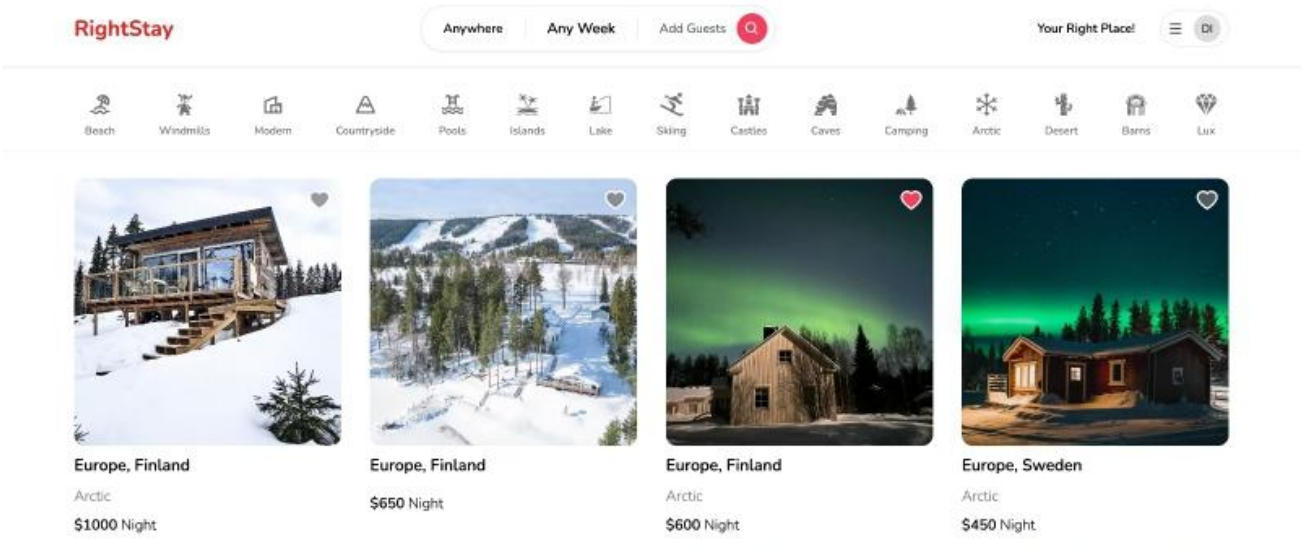


Figure 4.1

1. Header Section:

- **Logo:** "RightStay" is placed on the top left in a clean, red font.
- **Search Filters:**
- **Location Selector:** “Anywhere”
- **Date Picker:** “Any Week”
- **Guest Selector**: “Add Guest

- **Login page :**

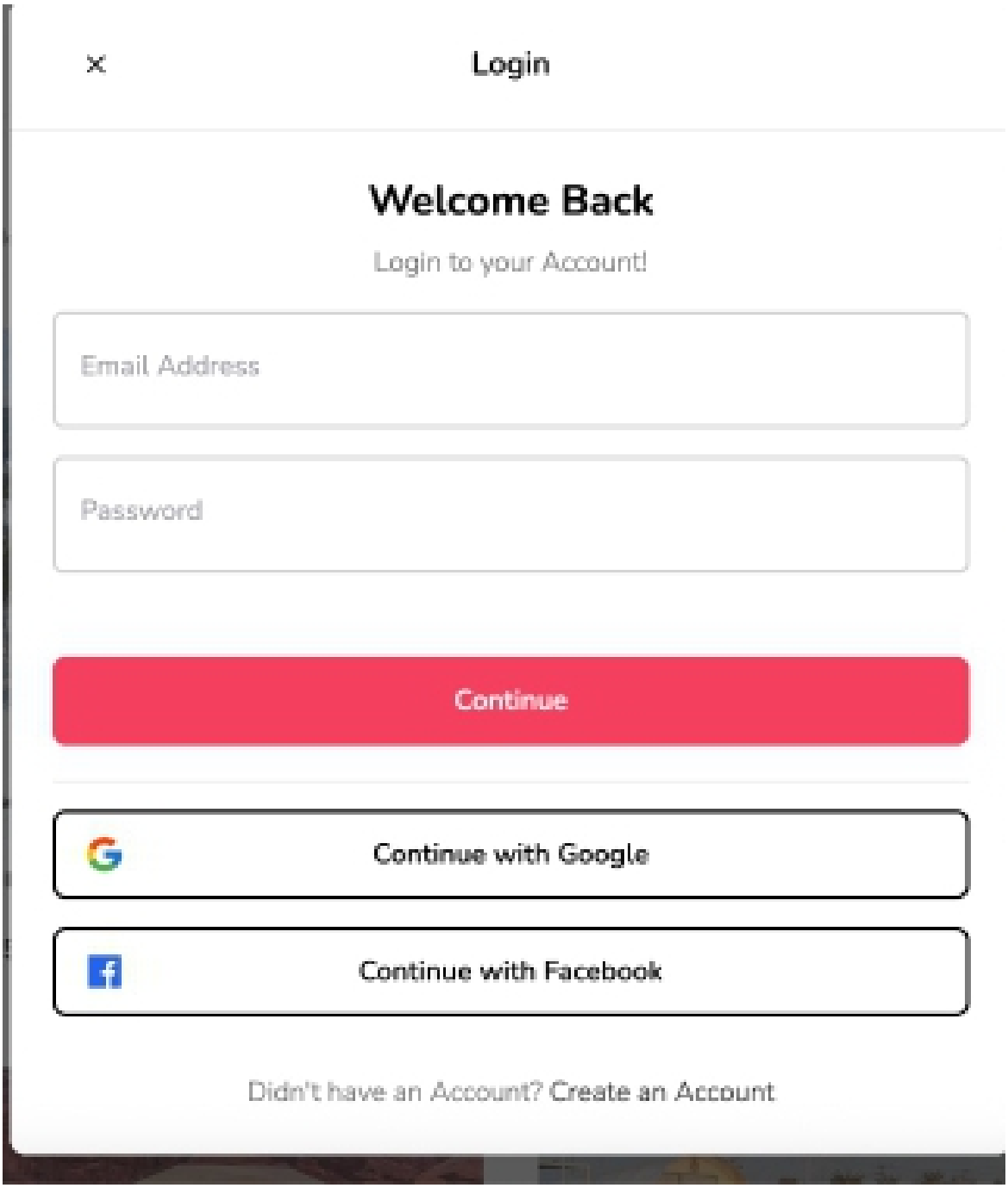


Figure 4.2

- **Register**

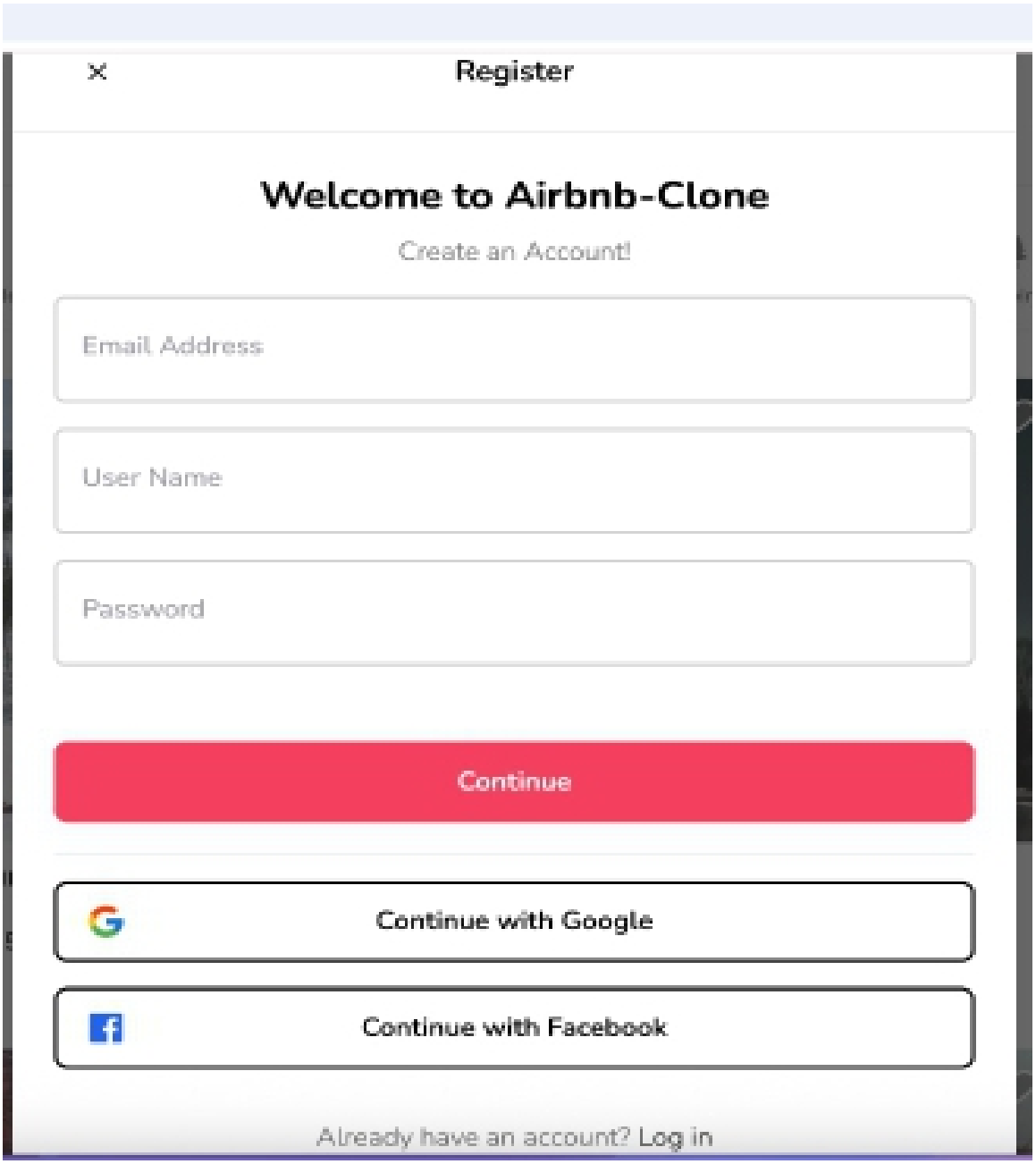


Figure 4.3

- **Filter Option**

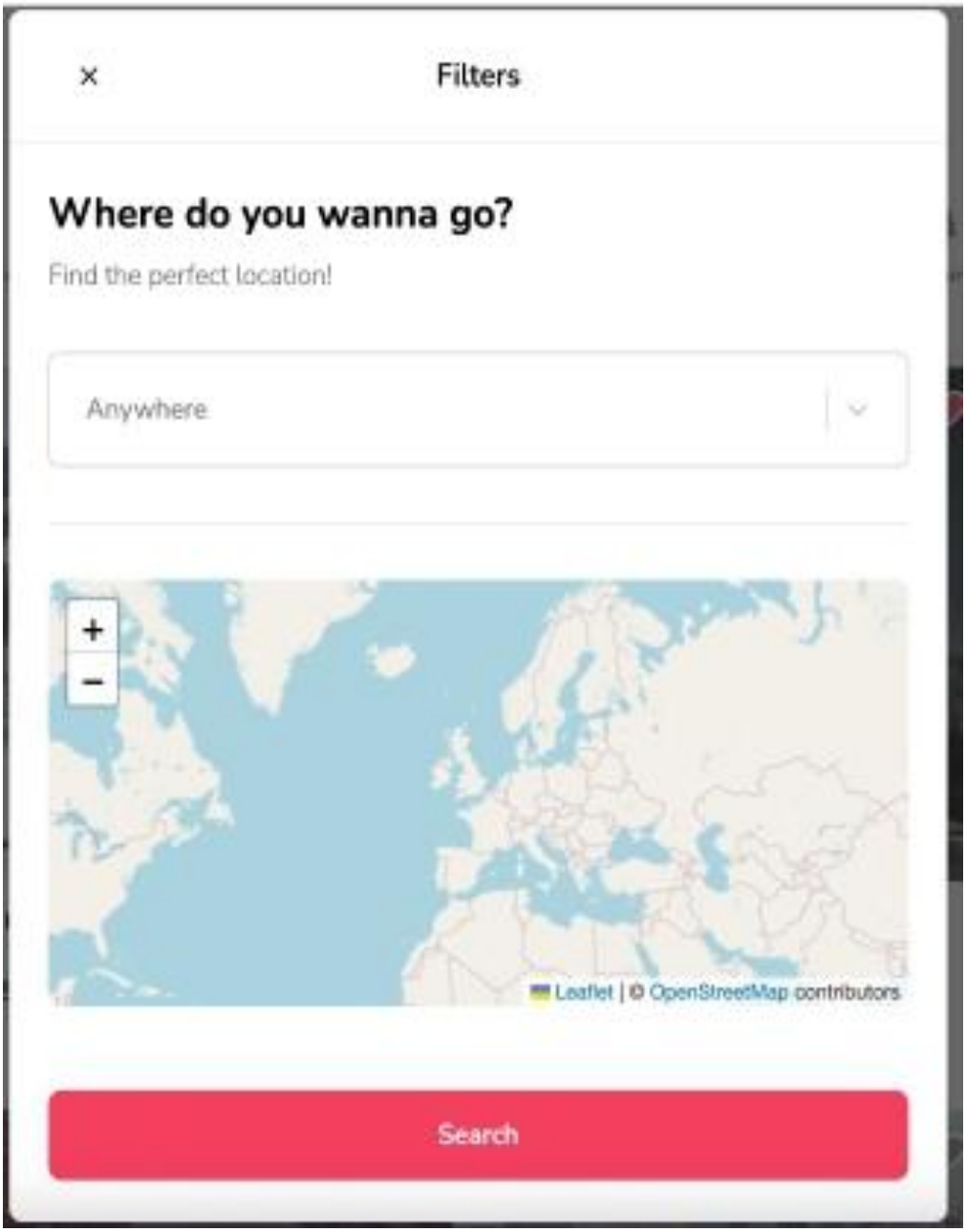


Figure 4.4

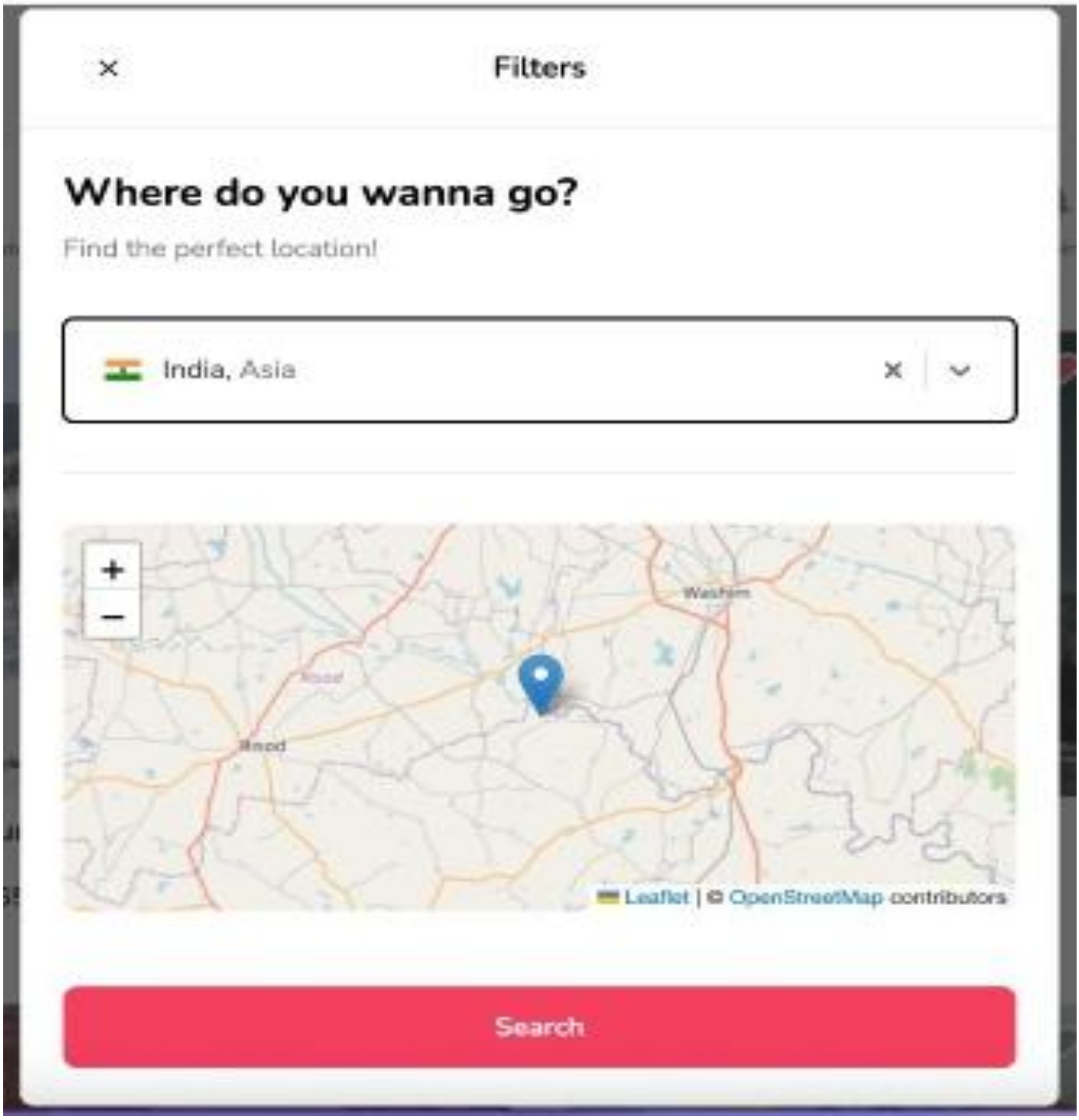

Filters
Where do you wanna go?
Find the perfect location!
India, Asia
Leaflet | © OpenStreetMap contributors
Search

Figure 4.5

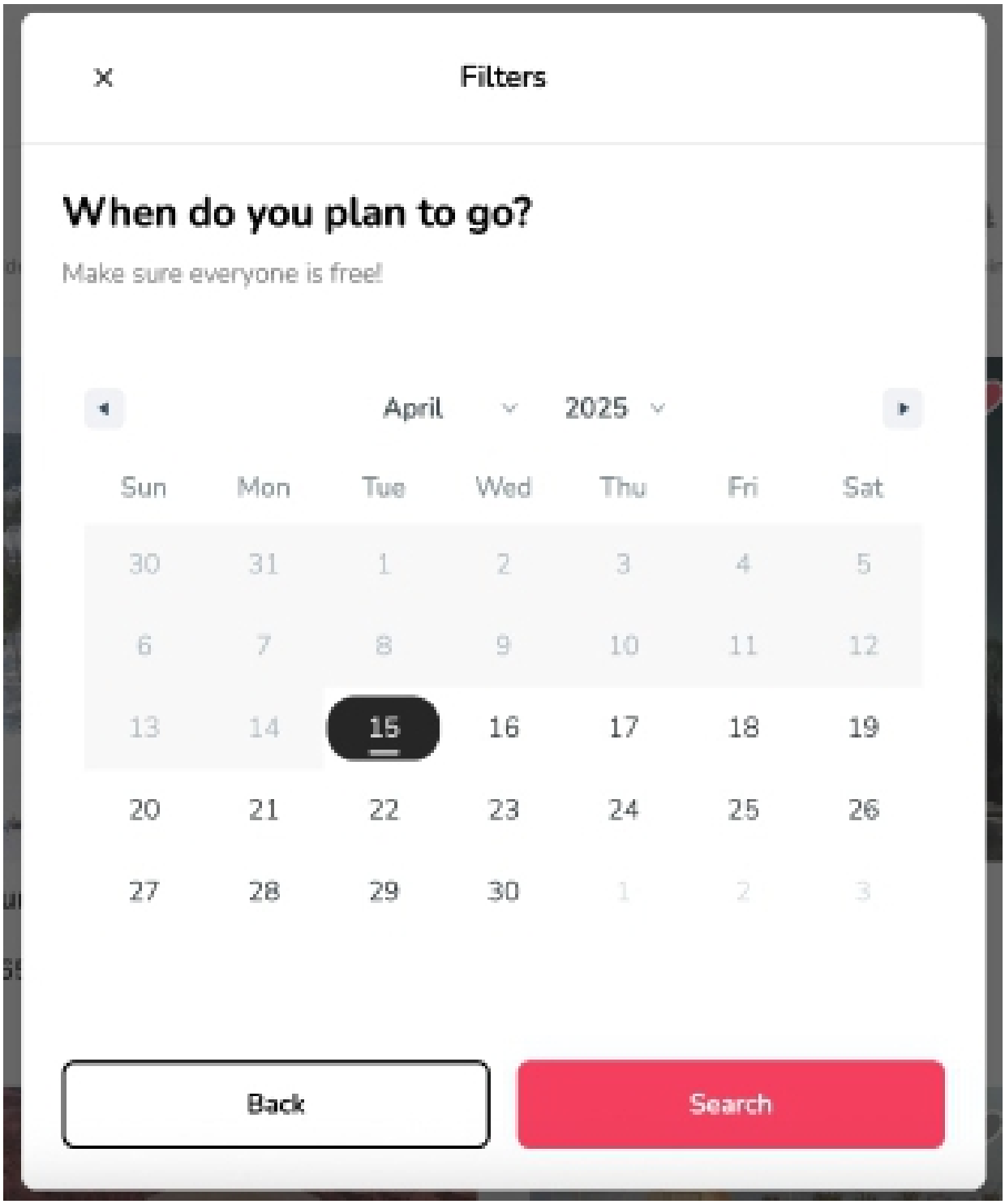

Filters
When do you plan to go?
Make sure everyone is free!
April
2025
Sun
Mon
Tue
Wed
Thu
Fri
Sat
30
31
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
1
2
3
Back
Search

Figure 4.6

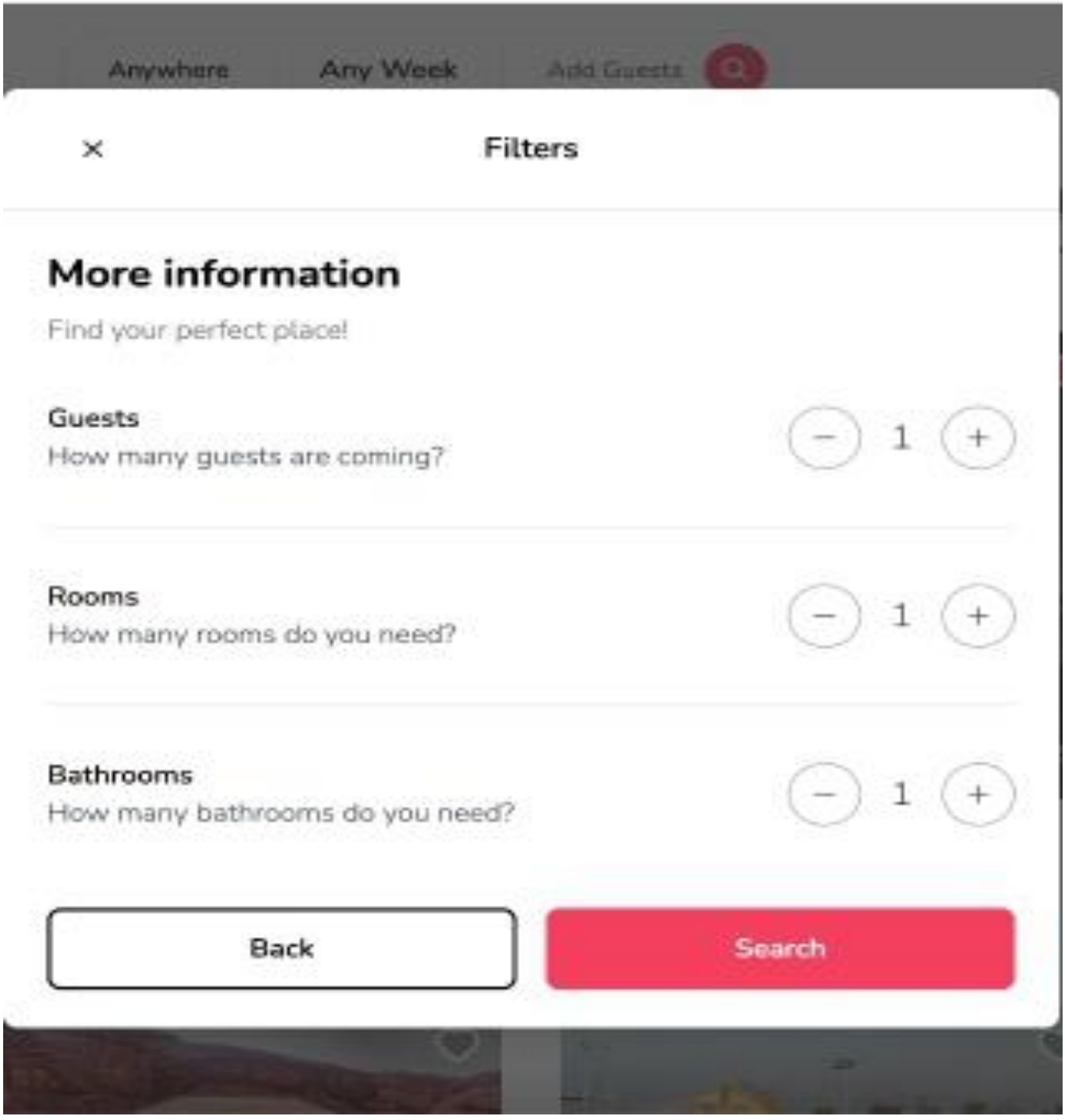

Anywhere
Any Week
Add Guests
Filters
More information
Find your perfect place!
Guests
How many guests are coming?
1
Rooms
How many rooms do you need?
1
Bathrooms
How many bathrooms do you need?
1
Back
Search

Figure 4.7

- **Favorites**

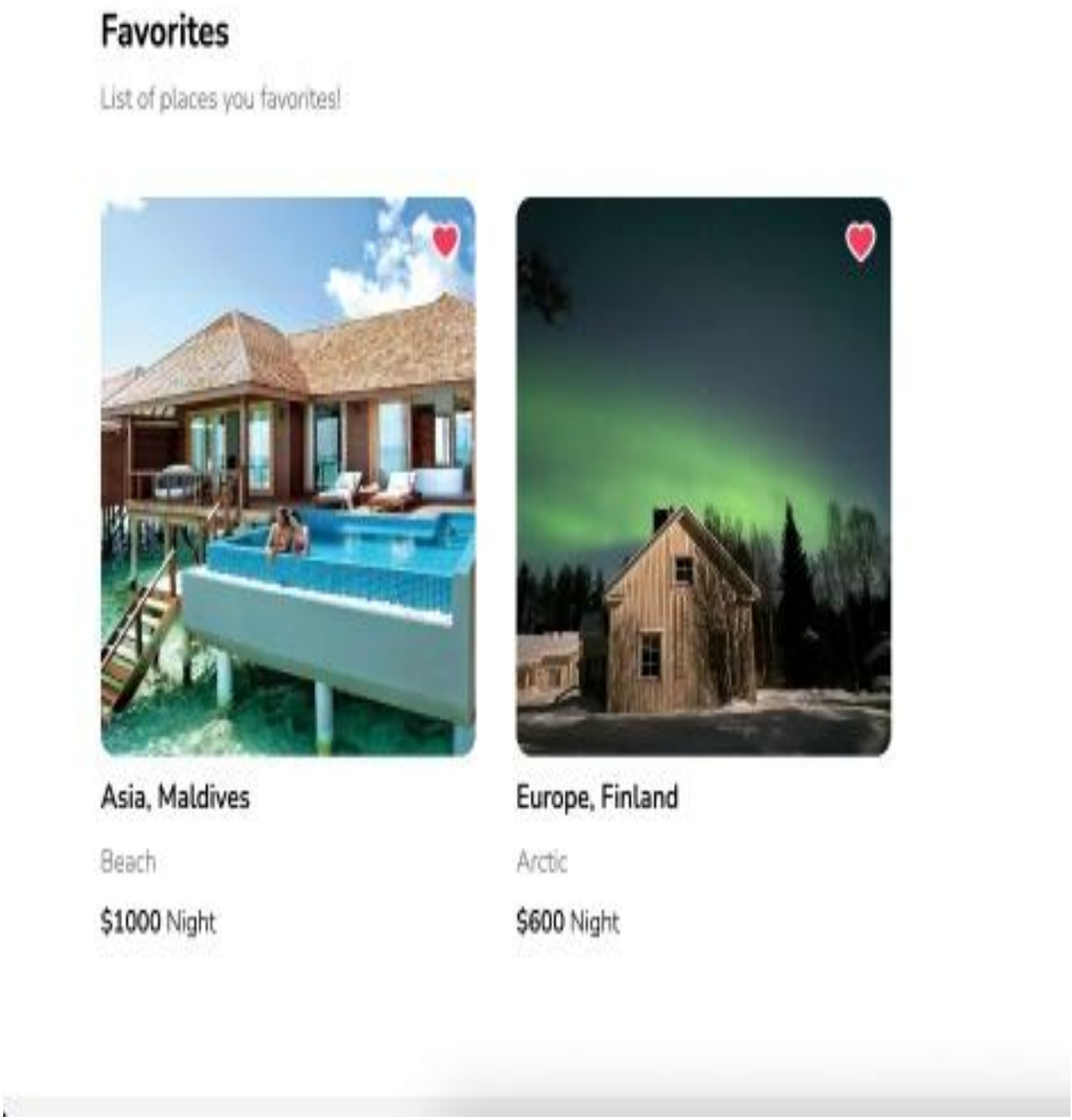


Figure 4.8

- **Properties**

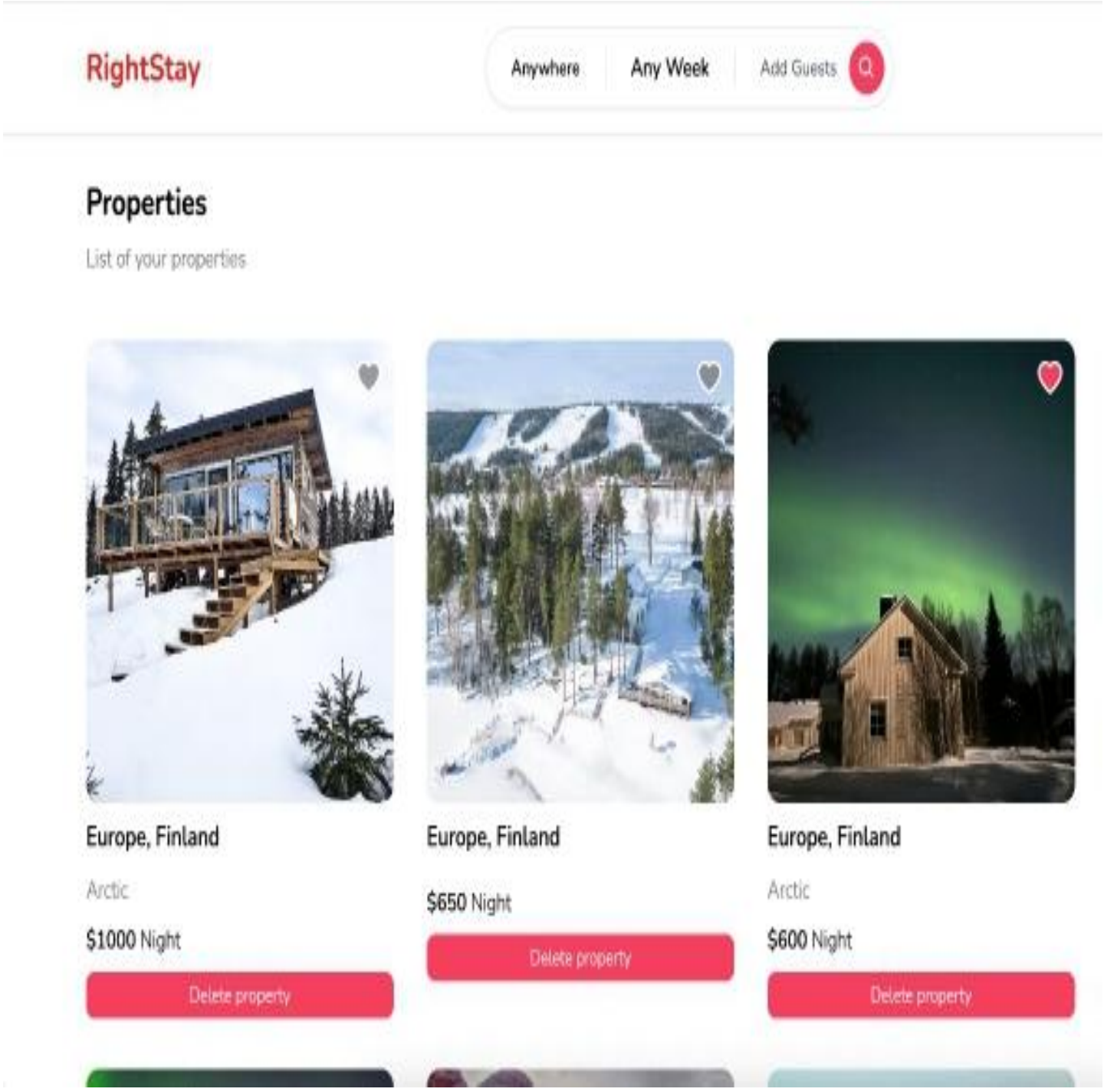


Figure 4.9

- **Reservation**

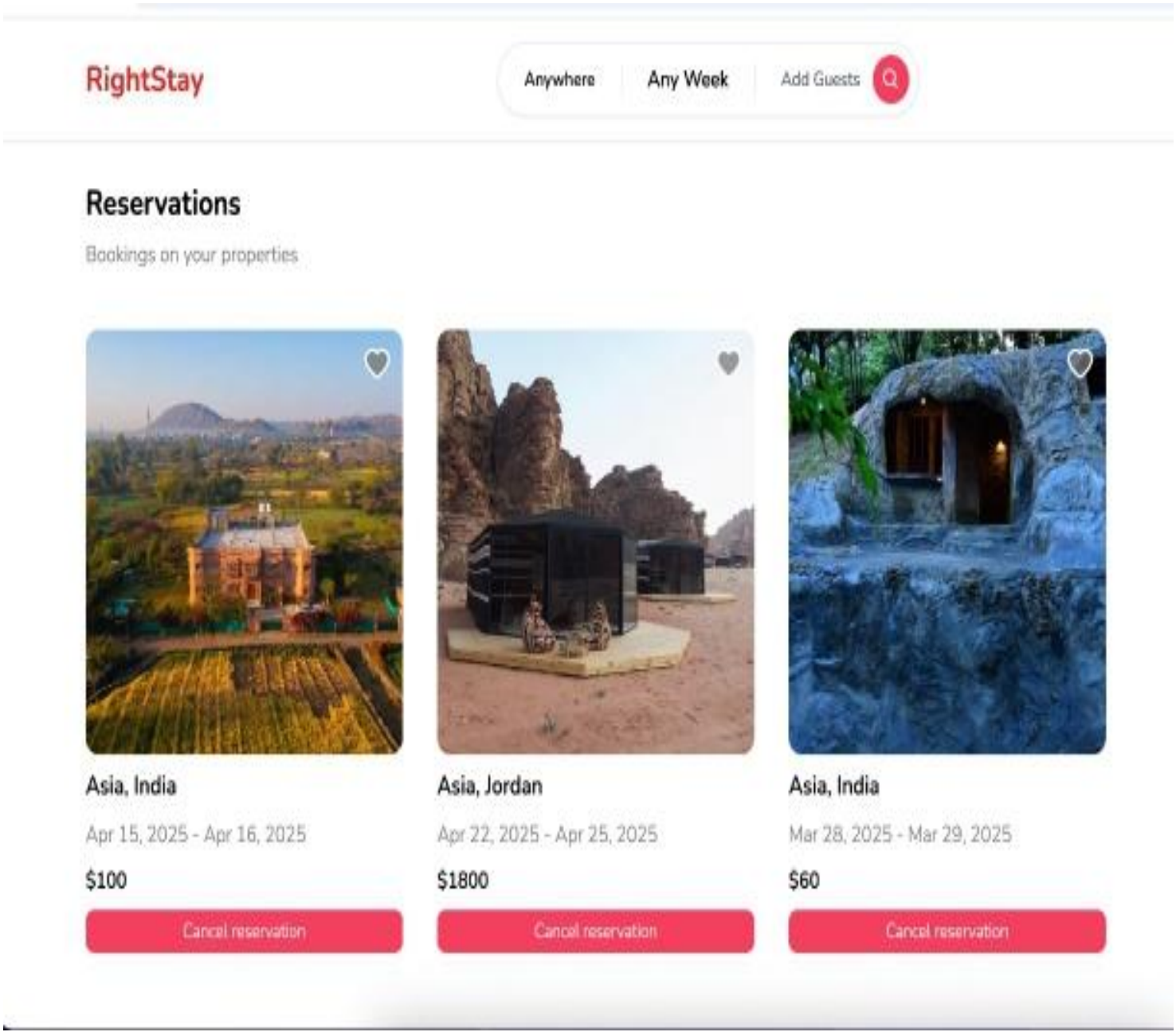


Figure 4.10

CHAPTER 5
TASK ANALYSIS AND SCHEDULE OF ACTIVITIES

5.1 Task Decomposition

The project is decomposed into the following tasks based on functional modules and development phases:

1. **Requirement Analysis**
   - Gather user and system requirements.
   - Define scope and objectives.

2. **System Design**
   - Create architectural diagrams (DFD, ERD, Class Diagrams).
   - Finalize technology stack and APIs.

3. **Module Development**
   - User Management Module
   - Property Listing & Search Module
   - Booking & Payment Module
   - Admin & Analytics Dashboard
   - Notification & Communication Module

4. **Integration & Testing**
   - API integration between frontend and backend.
   - Unit, integration, and user acceptance testing.

5. **Deployment**
   - Cloud setup (AWS/Google Cloud).
   - Security and scalability configurations.

6. **Documentation & Final Review**
   - Prepare user manuals and technical reports.
   - Final system validation.

5.2 Project Schedule

Task	Timeline	Duration
Requirement Analysis	June 2024	2 weeks
System Design	July 2024	3 weeks
Module Development	August – October 2024	10 weeks
Integration & Testing	October 2024	3 weeks
Deployment	November 2024	1 week
Documentation	November 2024	1 week

5.3 Task Specification

Task	Goals	Inputs	Outputs	Effort (Person-Hours)	Duration	Dependencies
User Management Module	Implement secure authentication.	User data, JWT tokens	Login/registration workflows	40	3 weeks	Database Module
Property Listing Module	Enable dynamic property listings.	Host inputs, pricing API	Host dashboard, search filters	60	3 weeks	Backend API Integration

Booking Module	Real-time booking & payment.	Availability data	Booking confirmation, receipts	50	3 weeks	Payment Gateway Integration
Admin Dashboard	Monitor platform metrics.	Analytics data	Reports, user activity logs	45	3 weeks	Database & Analytics Tools
System Testing	Validate end-to-end workflows.	Test cases	Test reports, bug fixes	35	3 weeks	All modules completed

CHAPTER 6
PROJECT MANAGEMENT

6.1 Major Risks and Contingency Plans

Risk	Contingency Plan
Technical Integration Failures	- Use mock APIs during development to simulate third-party services (e.g., payment gateways). - Maintain backup cloud configurations and fallback APIs.
Security Breaches	- Conduct regular penetration testing and code reviews. - Implement end-to-end encryption and OAuth/JWT for secure authentication.
Regulatory Compliance Issues	- Assign a legal advisor to monitor regional short-term rental laws. - Design modular compliance features for quick updates.
Team Coordination Delays	- Use Agile tools (Jira, Trello) for task tracking. - Schedule weekly sync-ups to address bottlenecks.
Low User Adoption	- Conduct usability testing during development. - Implement A/B testing for UI/UX improvements post-launch.

6.2 Principle Learning Outcomes

1. **Technical Proficiency:** Mastered full-stack development (React, Node.js, RESTful APIs) and cloud deployment (AWS/Google Cloud).

2. **Agile Management:** Improved skills in sprint planning, risk mitigation, and resource allocation using Agile methodologies.

3. **Security Best Practices:** Learned to implement robust encryption, secure payment gateways, and GDPR-compliant data handling.
4. **User-Centric Design:** Recognized the importance of intuitive UI/UX through iterative prototyping and feedback loops.

5. **Cross-Functional Collaboration:** Enhanced teamwork through roles like backend integration, frontend design, and DevOps coordination.

6. **Regulatory Adaptability:** Gained insights into navigating dynamic legal frameworks for global platforms.

PLAGIARISM REPORT

DrillBit

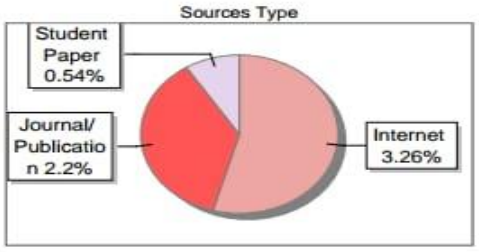
The Report is Generated by DrillBit Plagiarism Detection Software

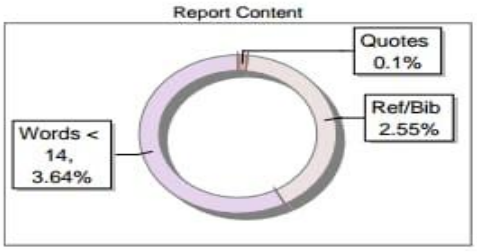
Submission Information

Author Name	Anurag Sharma
Title	Right Stay
Paper/Submission ID	3519677
Submitted by	library@coet.in
Submission Date	2025-04-17 21:51:52
Total Pages, Total Words	49, 7053
Document type	Thesis

Result Information

Similarity **6 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

6	32	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	medium.com	1	Internet Data
2	pmc.ncbi.nlm.nih.gov	<1	Internet Data
3	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
4	athena.ecs.csus.edu	<1	Publication
5	repository.ibs.ac.id	<1	Publication
6	docplayer.net	<1	Internet Data
7	www.linkedin.com	<1	Internet Data
8	statewatch.org	<1	Publication
9	academicjournals.org	<1	Publication
10	jmiiqac.ac.in	<1	Publication
11	open.uct.ac.za	<1	Publication
12	egu.eu	<1	Internet Data
13	en.nsd.pku.edu.cn	<1	Internet Data
14	fastercapital.com	<1	Internet Data

15	sportdocbox.com	<1	Internet Data
16	Submitted to YBN University on 2024-01-05 13-11	<1	Student Paper
17	www.finder.com	<1	Internet Data
18	www.freepatentsonline.com	<1	Internet Data
19	www.ijcaonline.org	<1	Publication
20	bidtostay.net	<1	Internet Data
21	build.saint-gobain.co.uk	<1	Internet Data
22	dannyboston.blogspot.com	<1	Internet Data
23	digitalcommons.library.umaine.edu	<1	Publication
24	docview.dlib.vn	<1	Publication
25	fastercapital.com	<1	Internet Data
26	gredos.usal.es	<1	Publication
27	industry.co.in	<1	Internet Data
28	moam.info	<1	Internet Data
29	moam.info	<1	Internet Data
30	qdoc.tips	<1	Internet Data
31	www.easemytrip.com	<1	Publication
32	www.skillsforemployment.org	<1	Publication

IJARESM

ISSN: 2455-6211, New Delhi, India

International Journal of All Research Education & Scientific Methods

An ISO & UGC Certified Peer-Reviewed/ Refereed Journal

UGC Journal No. : 7647

Certificate of Publication

Anurag Sharma

MGM COET, Noida, U. P. 201301, India

TITLE OF PAPER

Building the Future of Vacation Rentals: A Comparative Analysis of Right Stay

has been published in

IJARESM, Impact Factor: 8.536, Volume 13 Issue 3, March-2025

Certificate Id: IJ-2203251205

Date: 22-03-2025

Website: www.ijaresm.com Email: editor.ijaresm@gmail.com

Authorized Signatory

RESEARCH PAPER

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 3, March-2025, Available online at: www.ijaresm.com

Building the Future of Vacation Rentals: A Comparative Analysis of Right Stay

Dr. Karamjeet Kaur[1], Mayank Kumar Singh[2], Mohit Sharma[3], Dinesh Singh Bohra[4], Anurag Sharma[5]

[1,2,3,4,5]MGM COET, Noida, U. P. 201301, India

ABSTRACT

The emergence of digital platforms like RightStay and Airbnb has revolutionized the vacation rental industry, providing scalable, user-centric, and flexible alternatives to traditional lodging. This paper explores these platforms' technological, business, and regulatory aspects, offering insights from the perspectives of investors, users, and stakeholders. A comprehensive analysis highlights RightStay's cultural and service-driven innovations, Airbnb's SaaS expansion, market impacts, and the industry's challenges. This work aims to provide an inclusive understanding of how digital transformation aligns economic, social, and environmental imperatives for sustainable growth.

Keywords—Vacation Rental Industry; Rightstay; Airbnb; Digital Transformation; Peer-To-Peer Platforms; Sustainability; Saas; Regulatory Compliance.

INTRODUCTION

The rise of digital platforms has transformed various industries, and the vacation rental sector is no exception.

Platforms such as RightStay have introduced a new paradigm for travelers seeking short-term accommodations, shifting the focus from traditional hotels to personalized, peer-topeer lodging experiences. This paper reviews the key technological advancements and market strategies adopted by leading vacation rental platforms like RightStay and Airbnb, assessing their impact on both the industry and consumers.

Airbnb, in particular, has played a pivotal role in reshaping the vacation rental landscape. The platform connects hosts with travelers, offering unique, affordable, and flexible lodging options. As a leader in the industry, Airbnb has faced significant challenges, including regulatory concerns and competition from traditional hotels. Despite these obstacles, Airbnb continues to innovate, especially by expanding into the B2B and SaaS sectors.

THE LITERATURE REVIEW

The article explains how Airbnb used its branding and SEO to maintain traffic, even without paid marketing during the pandemic. It highlights the role of Airbnb's PR strategy in boosting its brand presence and mentions how their SEO was mainly driven by branded searches. Programmatic SEO

strategies, like targeted pages for specific locations, also contributed to its success. The article emphasizes that Airbnb's strong brand identity, which has become a verb (like "Kleenex"), was crucial in its sustained traffic and recognition.

The Airbnb India page features a variety of holiday rental options across the country, highlighting top-rated stays such as treehouses, luxury cabins, private villas, and unique farm stays. These properties offer scenic views, peace, and comfort, and cater to different types of travelers.

Amenities include kitchens, pools, and air conditioning. Popular options also emphasize local experiences, such as village walks and cooking classes, connecting travelers with nature and local culture. For more details, visit Airbnb India.

This guide provides insight into the traditional Japanese inn known as "Ryokan." It explains their cultural importance, unique features such as tatami flooring and futon bedding, and how they cater to tourists in search of an authentic Japanese experience. This type of accommodation is different from modern hotels and platforms like Airbnb.

The article explores the historical and cultural perspectives of Ryokan. It explains how they evolved from serving traveling samurai to modern-day tourists. It also talks about the challenges these traditional inns face, especially because of competition from global and modern accommodation services.

The study explores the impact of Airbnb on Japan's hotel industry. It shows that the emergence of Airbnb has resulted in a marked shift in the market, affecting pricing, occupancy rates, and customer preferences. The research also touches on how traditional accommodations, such as Ryokan, have been affected by this competitive disruption.

This paper examines regional disparities in how Airbnb affects Japan's hotel sector. It identifies that urban areas with high tourist volumes experience greater competition, while rural regions see varying impacts. The study also considers Japan's legal framework for private lodging and its effect on industry dynamics.

The authors analyze the collision between the traditional economy of Japan and the emerging sharing economy, emphasizing Airbnb and Uber. Paper discusses how regulatory frameworks and society norms in Japan influence the adoptions and working of those platforms, reflecting cultural and economic tensions. This is yet another long study from the Journal of Destination Marketing & Management, and likely also a duplicate of the above citation. It drives the point again that Airbnb has revolutionarily changed Japan's hospitality scene by underlining regional, economic, and cultural impacts.

Problems & Solutions

The vacation rental and lodging industry faces great challenges in meeting the diverse needs of modern travelers while ensuring operational efficiency and sustainability. This section formalizes these issues by integrating quantitative insights and mathematical modeling to underscore the complexities and potential areas for optimization.

1.Operational Complexity in Property Management: Managing vacation rentals is akin to operating (n)-independent hotel rooms, where each room represents a distinct property with unique operational requirements. This involves the interaction of (m)stakeholders through a variety of third-party applications. The complexity can be calculated as:

$$C = n.m.f(a, p)$$

Where: n: Number of properties (e.g., 50). m: Number of third-party applications (e.g., 5–10).
f(a,p): A failure probability function dependent on availability a and performance p of applications.

High C values correlate with greater risks of operational failures, such as missed updates (e.g., door codes), directly impacting the guest experience.

Challenges in Meeting Traveler Demands:

The traditional lodging industry often struggles with high costs, inflexible offerings, and a lack of personalized guest experiences.

To measure the gap in service:

A. Cost Elasticity of Demand

The demand D for lodging services is inversely proportional to price \(P\), modeled as: The demand D for lodging services is inversely proportional to price P, modeled as:

$$D = k. P^{-e}$$

Whereas k is a proportionality constant, and e>1represents the elasticity of demand. High costs (P) deter travelers, lowering D.

B. Diversity of Accommodation Options

Given \(o_t\) as the number of accommodations options provided by traditional models and \(o_m)\ as by modern platforms like RightStay, the diversity index I is expressed as:

$$I = \frac{o_m}{o_t}$$

Higher \(I\) indicates greater diversity and adaptability in modern platforms.

C. Proposed Solution Framework

To address these challenges, a scalable business model leveraging technology is proposed. The objective is to optimize \(C\), maximize \(E\), and minimize \(F\). This can be achieved through:

1.Integrated Systems: A unified tech stack to reduce (C). 2.Personalized Offerings: Dynamic pricing (P) and diverse accommodations (I) to increase (D).
3.Economic Empowerment: Enhancing \(u_i\) through peertopeer platforms to maximize \(E\).
4.Sustainability Goals: Reducing \(T\) and \(L\) to minimize \(F\) and \(H\), aligning with responsible tourism practices.

These formulations provide a mathematical perspective to evaluate and improve the operational, economic, and sustainability dimensions of the vacation rental industry.

METHODOLOGY

The methodology section outlines the approach used to analyze the unique cultural, operational, and customer service aspects of Japanese ryokan and Airbnb. This comparative analysis was conducted through qualitative and quantitative methods, focusing on understanding the differences and similarities between traditional and modern hospitality practices.

This study uses a qualitative methodology to explore the design and user experience of RightStay. Data collection involved examining the features of the platform through user feedback, expert reviews, and market analysis. A comparative study of Airbnb and similar platforms was conducted to understand the business model and technological tools utilized. Surveys were also distributed to property owners and users to assess their experiences and preferences. This comprehensive approach allows for an in-depth understanding of the platform's potential in a competitive landscape.

A mixed-method research design was employed to gain insights into the historical and cultural context of ryokan and the contemporary appeal of Airbnb. The study involved a combination of reviewing scholarly articles, books, and case studies on Japanese ryokan and the evolution of Airbnb.

Visiting selected ryokan and Airbnb properties in Japan to observe their operations, design elements, and customer interaction styles.

This study investigates the design, content, and user experience of Japanese ryokan websites to understand how they convey cultural authenticity and attract international tourists.

International Journal of All Research Education and Scientific Methods (IJARESM),
ISSN: 2455-6211, Volume 13, Issue 3, March-2025, Available online at: www.ijaresm.com

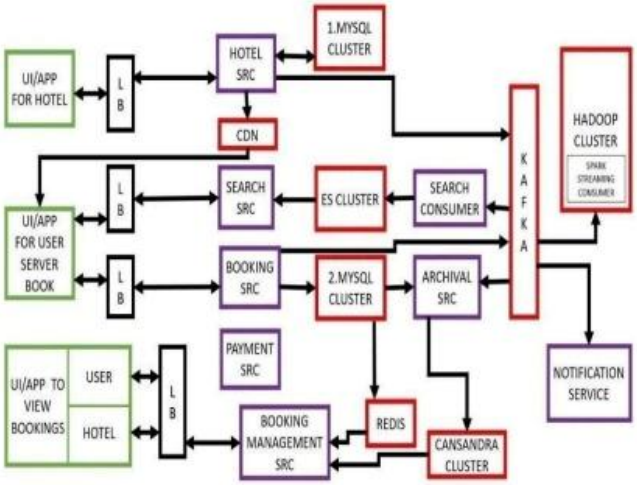
This study examines the features and design of the Airbnb website to understand how it caters to diverse audiences while promoting local and cultural experiences.

Feasibility Analysis

"RightStay" is a technically feasible platform that uses modern technologies to provide robust functionality and scalability. The technology stack, with React and Node.js/Express, provides a solid base for development, while cloud infrastructure services like AWS and Google Cloud ensure scalability to support growing user traffic. Integration of machine learning tools such as TensorFlow and PyTorch enables personalized recommendations and dynamic pricing strategies. Third-party APIs to pay (e.g., Stripe, PayPal), send messages (e.g., Twilio), and navigate (e.g., Google Maps) simplify core functions. A microservices architecture further enhances scalability through the ability to scale components independently, ensuring smooth and efficient operations (Singh & Patel, 2022).

The operational feasibility of "RightStay" is underscored by its emphasis on user-friendly design and automation, which enhance efficiency and user satisfaction. An intuitive user interface simplifies the experience for both hosts and guests, improving overall satisfaction (Martin et al., 2023). Automation of booking processes, payment systems, and communication channels, supported by AI-driven recommendations, reduces manual overhead and optimizes operations. Robust security measures, including data encryption, user verification, and fraud detection, strengthen platform trust and encourage user adoption. The basically cloud-based, modular architecture ensures minimum latency, meaning that the system is effectively supporting a growing userbase.

"RightStay" shows high economic feasibility by scaling up the revenue models and cost-efficient development strategies. It can generate revenues through booking fees, premium subscription, and advertisements. Its cost management strategy is achieved by open-source tools and cloud optimization that minimizes initial and operational costs. When the user base increases, it can enjoy economies of scale and increase its profitability and improve profit margins over time (Wang & Zhang, 2021).



Working Diagram

CONCLUSION

RightStay has the potential to make a significant impact on the online booking industry by offering a platform that caters to both property owners and guests. Through its technological infrastructure and customer-focused approach, it positions itself as a competitor to Airbnb. However, continued growth will depend on its ability to address challenges such as market penetration and differentiation. Further studies are needed to track its success and evolution in the competitive market.

References

1. Ding K, Niu Y, Choo WC. The evolution of Airbnb research: A systematic literature review using structural topic modelling. Helicon. 2023 June 8;9(6):e17090. doi: 10.1016/j.heliyon.2023.e17090. PMID: 37484274; PMCID: PMC10361235.

2. https://www.pymnts.com/in-depth/2015/airbnb-enters-the-b2b-saas-market/
3. https://www.researchgate.net/publication/333880155ProgressonAirbnba literaturereview
4. https://medium.com/@MustafaJamalNasser/revolutionizing-hospitality-airbnbs-success-story-powered-by-b2b-and-saas-innovation-eb5e44ae47cc
5. https://stars.library.ucf.edu/etd/6327/
6. https://www.linkedin.com/pulse/impact-airbnb-hotels-hospitality-industry-ahmed-mahmoud-1f/
7. https://foundationinc.co/lab/airbnb-branding-success
8. https://openai.com/index/chatgpt/
9. https://en.wikipedia.org/wiki/YouTube
10. https://www.airbnb.co.in/india/stays
11. https://www.japan-guide.com/e/e2029.html
12. https://en.wikipedia.org/wiki/Ryokan
13. Nakamura, Satoka& Baskaran, Angathevar& Kumari, Sonia. (2024). Impact of Airbnb on the hotel industry in Japan. Journal of Destination Marketing & Management. 31. 100841. 10.1016/j.jdmm.2023.100841.
14. Ito, Yoshihiro and Mitsumoto, Daiki and Yuasa, Kosaku, The Impact of Airbnb on the Hotel Industry: The Effect of Regional Differences and Private Lodging in Japan (January 8, 2024). Available at SSRN: https://ssrn.com/abstract=4686957 or http://dx.doi.org/10.2139/ssrn.4686957
15. ALTURA, Thomas & HASHIMOTO, Yuki & Jacoby, Sanford & KANAI, Kaoru & SAGUCHI, Kazuro. (2020). Japan Meets the Sharing Economy--Airbnb and Uber: Contending Frames. Social Science Japan Journal. 24. 10.1093/ssjj/jyaa041.
16. Satoka Nakamura, Angathevar Baskaran, Sonia Kumari Selvarajan, Impact of Airbnb on the hotel industry in Japan, Journal of Destination Marketing & Management, Volume 31,2024, 100841, ISSN 2212-571X, https://doi.org/10.1016/j.jdmm.2023.100841.(https://www.sciencedirect.com/science/article/p ii/S2212571X2300080X)

17. https://www.theguardian.com/travel/shortcuts/2017/dec/10/the-ryokan-the-ancient-japanese-inn-that-is-the-next-big-airbnb-thing
18. https://www.airbnb.co.in/help/article/2353
19. https://medium.com/airbnb-engineering
20. https://medium.com/@airbnbeng
21. https://www.ycombinator.com/companies/hostai