

**Property Rentals Management System Requirements  
Specification**

**Version 1.0**

**May 24, 2024**

# Table of Contents

<b>PROPERTY RENTALS MANAGEMENT SYSTEM REQUIREMENTS SPECIFICATION</b>	<b>1</b>
<b>VERSION 1.0</b>	<b>1</b>
<b>MAY 24, 2024</b>	<b>1</b>
<b>1. Error! Bookmark not defined.</b>	
1.1 <b>Error! Bookmark not defined.</b>	
1.2 <b>Error! Bookmark not defined.</b>	
<b>2. Error! Bookmark not defined.</b>	
2.1 <b>Error! Bookmark not defined.</b>	
2.2 <b>Error! Bookmark not defined.</b>	
2.3 <b>Error! Bookmark not defined.</b>	
2.4 <b>Error! Bookmark not defined.</b>	6
2.5 <b>Error! Bookmark not defined.</b>	
<b>3. Error! Bookmark not defined.</b>	
3.1 <b>Error! Bookmark not defined.</b>	
3.2 <b>Error! Bookmark not defined.</b>	
3.2.1 <b>Error! Bookmark not defined.</b>	
3.2.1.1    11	
3.2.1.2    12	
3.2.1.3 <b>Error! Bookmark not defined.</b>	
3.2.1.3.1    13	
3.2.1.3.2 <b>Error! Bookmark not defined.</b>	
3.2.1.4 <b>Error! Bookmark not defined.</b>	
3.2.1.5 <b>Error! Bookmark not defined.</b>	
3.2.2 <b>Error! Bookmark not defined.</b>	
3.2.2.1 <b>Error! Bookmark not defined.</b>	
3.2.2.2 <b>Error! Bookmark not defined.</b>	
3.2.2.3 <b>Error! Bookmark not defined.</b>	
3.2.3 <b>Error! Bookmark not defined.</b>	
3.2.3.1 <b>Error! Bookmark not defined.</b>	
3.2.3.2 <b>Error! Bookmark not defined.</b>	
3.2.3.3 <b>Error! Bookmark not defined.</b>	
3.2.3.3.1 <b>Error! Bookmark not defined.</b>	
3.2.3.3.2 <b>Error! Bookmark not defined.</b>	
3.3 <b>Error! Bookmark not defined.</b>	
<b>4. Error! Bookmark not defined.</b>	
<b>5. DIAGRAMS</b>	
Entity Relationship Diagram (ERD)	49
BPMN Diagrams	50
DFD Diagrams	51
Activity Diagrams	53
Use Case Diagram	83
State Diagrams	84
Class Diagram	88
Sequence Diagrams	89
Collaboration Diagrams	115
Interaction Overview Diagram	128
<b>6. DESIGN PATTERNS</b>	<b>129</b>
<b>APPENDIX</b>	<b>132</b>

## 1. Executive Summary

### 1.1 Project Overview

The Property Rental Management System (PRMS) is designed to streamline the management of rental properties, facilitating efficient and effective operations for property owners, managers, and guests. This system will integrate various functionalities to address the needs of all stakeholders involved in property rental processes, such as property listings management, rent collection and financial tracking, communication platforms, and analytics and reporting.

Property Listings Management ensures that the processes of listing and managing properties are handled correctly. This functionality allows property owners to list available properties with detailed descriptions and images, including essential details such as property type, amenities, and rental conditions. It enables property managers to update property listings, ensuring the information remains current and accurate. Additionally, it tracks occupancy rates to identify trends and optimize property utilization, making sure that properties are used to their fullest potential.

Financial Tracking ensures that every financial transaction is conducted properly and securely. Guests make payments through a highly secure financial infrastructure on the internet called Stripe, which ensures secure and encrypted transactions to protect financial information. This functionality also provides comprehensive financial reporting tools for property owners and managers, allowing them to monitor revenue, expenses, and profitability. By maintaining a robust and secure financial tracking system, PRMS supports the financial health and transparency of property rental operations.

The Communication Platform functionality facilitates communication between hosts and guests through notification systems. It allows for seamless interaction, notifying hosts of any updates or changes to the property status as they happen, ensuring that hosts are always informed about the current state of their properties and any actions that need to be taken.

Analytics and Reporting functionality is responsible for providing accurate and up-to-date reports when requested. This includes financial reports, which can be generated upon the admin's request, offering insights into revenue, expenses, and overall financial performance. The system also generates admin performance reports, which provide metrics on the efficiency and effectiveness of system management. These reports are essential for making informed decisions and improving the management of rental properties.

The primary users of the Property Rental Management System include:

Property Owners

Hosts

Guests

# Property Rentals Management System Requirements Specification

## Purpose and Scope of this Specification

The purpose of this specification is to define the functional and non-functional requirements for the Property Rental Management System (PRMS). It aims to provide a thorough and understandable roadmap for the development team to implement the system complying with the stakeholders' needs. This document is purposefully created to be used by project managers, developers, testers, and other stakeholders involved in the design, development, and deployment of the PRMS. Project Managers to plan and oversee the project. Developers to understand the detailed requirements and build the system. Testers to design and execute test cases ensuring the system meets the requirements. Stakeholders to ensure their needs and expectations are accurately captured and addressed.

### In scope:

#### Property Listings Management:

- Allowing property owners to list available properties with detailed descriptions and images.
- Enabling property managers to update property listings, ensuring current and accurate information.
- Tracking occupancy rates to identify trends and optimize property utilization.

#### Financial Tracking:

- Providing online payment options through secure and encrypted transactions via Stripe.
- Managing financial records.
- Offering comprehensive financial reporting tools for monitoring revenue, expenses, and profitability.

#### Communication Platform:

- Facilitating communication between hosts and guests through notification systems.
- Notifying hosts of updates or changes to property status in real-time.

#### Analytics and Reporting:

- Generating financial reports and admin performance reports as requested.
- Providing accurate and up-to-date insights into various aspects of property management.

### Out of Scope:

#### Advanced Tenant Management Features:

- Features such as detailed tenant profiles, background checks, and lease agreement management.

#### Maintenance and Service Requests:

- Functionality for tenants to submit maintenance requests and track their status.

#### Integration with External Property Management Tools:

- Synchronizing property listings and availability with external platforms and tools not specified in this phase.

## **2. Product/Service Description**

In this section, describe the general factors that affect the product and its requirements. This section should contain background information, not state specific requirements (provide the reasons why certain specific requirements are later specified).

### **2.1 Product Context**

The Property Rental Management System (PRMS) is a comprehensive system created for managing rental properties efficiently. Although this product is essentially independent and self-contained, it can be made more functional by integrating with a number of related systems. Important external interfaces and connections consist of:

Payment Gateways: Stripe integration enables safe online payments.

Tools for Communication: Inbuilt messaging tools to let guests and hosts communicate.

Reporting Tools: Systems for internal analytics and reporting that produce performance and financial reports.

External Property Listings: To synchronize availability and occupancy data, there is a possibility of interface with external property listing websites.

Through these connections, PRMS is able to operate seamlessly as part of a larger ecosystem, offering a smooth experience to all parties involved in property rental management.

### **2.2 User Characteristic**

#### **Hosts**

Type: Experts or businesses that look after rental properties for owners.

Experience: Moderate to high; familiarity with procedures related to property management

Technical Expertise: Moderate to high; at ease using internet resources and management software.

Other Characteristics: A focus on tenant happiness, maintenance coordination, and effective property operations.

#### **Guests**

Type: People looking to rent out homes alone or with family

Experience: Widely varied, ranging from new tenants to long-term tenants.

Technical Expertise: often low to moderate; basic knowledge of navigating web pages.

Other Characteristics: looking for an easy-to-use and secure rental property search and acquisition platform.

#### **Admins**

Type: System administrators responsible for managing the PRMS.

Experience: Very high level; deep understanding of system upkeep and operation.

Technical Expertise: High; strong in user support, troubleshooting, and software management.

Other Characteristics: Guarantees system security, performance, and integrity.

## Property Rentals Management System Requirements Specification

### **2.3 Assumptions**

Operating System: A particular operating system (such as Windows Server or Linux) will be used on the servers where the PRMS is installed. Availability of this OS is assumed.

Internet Access: In order to communicate with the PRMS, users will have dependable internet access.

User Expertise: In order to navigate the system, users must possess fundamental computer abilities.

Payment Integration: For safe transactions, Stripe will be accessible and supported.

### **2.4 Constraints**

For security and accountability reasons, the system needs to keep track of all transactions and user activity through log files and a thorough audit trail.

To safeguard user data and guarantee that only authorized access is granted, strong security measures must be put in place. This covers data encryption, user authentication, and frequent security audits.

Criticality: High availability and dependability are essential for the PRMS, which is a crucial application for property management operations.

Within specified hardware parameters, such as memory, processor power, and disk space, the system must function effectively.

Following certain design guidelines, including utilizing a specific framework or programming language, may be required to ensure compatibility and maintainability.

### **2.5 Dependencies**

Data Downloads: Every day, the PRMS could need to download updated property listings from third-party websites, among other types of external data.

Module Dependencies: Before other PRMS modules may be produced, some must first be finished. For example, before deploying full reporting features, the financial tracking module needs to be operational.

Third-Party Services: The business will keep running and integrate with other services like external property listing platforms and Stripe for payment processing.

User Feedback: In order to improve the system's features and functionality over time, ongoing user feedback will be required.

## Property Rentals Management System Requirements Specification

### 3. Requirements

The following definitions are intended as a guideline to prioritize requirements.

- Priority 1 – The requirement is a “must have” as outlined by policy/law
- Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits

- Priority 3 – The requirement is a “nice to have” which may include new functionality

It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS must be either A or I" or "It would be nice if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category

#### 3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_LR_01	Different UI For each account type	There will be 3 roles, Admin, Host, Guest	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_02	Authentication and Authorization	All modification and queries of data must only be allowed by their corresponding roles	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_03	Hosts should be able to create listings for their properties	Hosts can create listings for their properties (entire place, private room, shared room) with descriptions, photos, amenities, location, pricing, and availability. Property Management: Enables hosts to effectively showcase their properties, attracting potential guests	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_04	The system should allow hosts to manage their booking requests	Hosts can manage booking requests, communicate with guests, and confirm reservations. Booking Management: Allows hosts to manage their bookings efficiently and communicate directly with guests.	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_05	Hosts can track their properties	Hosts can track their earnings, view transaction history, and manage payouts. Financial Management: Provides hosts with financial transparency and control over their earnings on the platform	2	10/04/24	Alvi Hysa, Indrit Ferati

## Property Rentals Management System Requirements Specification

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_LR_07	Guests should be able to browse properties	Guests can search for listings based on location, amenities, price range, and availability. Search Functionality: Enables guests to find properties that meet their needs and preferences.	1	10/04/24 05/23/2024	Alvi Hysa, Indrit Ferati
BR_LR_08	Guests can view property details	Guests can view detailed listing information, including photos, descriptions, house rules, and host reviews. Property Information: Provides guests with comprehensive information to make informed booking decisions.	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_09	Guests can book a property	Guests can send booking requests to hosts and communicate with them regarding their stay. Booking Functionality: Allows guests to easily initiate reservations and communicate directly with hosts.	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_10	Guests can leave reviews for their property	Guests can leave reviews for their stays, providing feedback for hosts and future guests. Review System: Enables guests to share their experiences and helps build trust within the platform.	2	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_11	The system should process transactions securely	The system implements a secure payment processing system for handling booking transactions. Secure Payment Processing: Ensures safe and reliable transactions for both hosts and guests.	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_12	The system should enable users to efficiently find properties that meet their criteria.	The system offers a user-friendly search and filtering system for finding listings. Search Functionality:	2	10/04/24	Alvi Hysa, Indrit Ferati

## Property Rentals Management System Requirements Specification

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_LR_13	The system should allow for a rating and review system for hosts and guests	The system implements a review and rating system for hosts and guests. Review System: Encourages accountability and trust within the platform by allowing users to evaluate each other's experience.	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_14	The system should allow for the host to set dynamic rental prices	The system implements a way to edit rental prices in real time in response to increased demand, change in season ect.	2	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_15	The system should allow the host to set availability	The system should allow the host to set the times in which the software is available	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_16	The system should generate reports for the admin	The system should be able to generate reports regarding finance, geographical data, user data ect.	3	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_17	The system should allow for a method to review properties by the admin	The system shouldn't allow any property to be entered into the system, it should make them reviewable by the admin	1	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_18	The system should allow malicious admins, reviews, guests,host to be reported and dealt with	There needs to be a mechanism for any malicious review, posting or admin to be reported by any user	2	10/04/24	Alvi Hysa, Indrit Ferati
BR_LR_19	The system should allow for favorite properties by the user	The user should be able to mark a property as favorite and have it be saved by the system	3	10/04/24	Alvi Hysa, Indrit Ferati

### 3.2 Non-Functional Requirements

## Property Rentals Management System Requirements Specification

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
R_01	Integration	The website should be able to integrate with other tools such as student information systems, video conferencing, plagiarism detectors, etc.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_02	Usability	The website should be user-friendly and the options should be organized and intuitive. The website should be easy to use by every user.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_03	Reliability	The website should have an almost uninterrupted uptime. It should have little to none errors and it should be able to handle large amount of traffic.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_04	Security	The website should be able to protect user data and encrypt sensitive information so unauthorized users cannot access it.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_05	Performance	The website should be fast and responsive. It should handle data and not slow down for any reasons.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_06	Compatibility	The website should be compatible with a wide range of screen sizes, browsers and operating systems.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_07	Scalability	The website should be modular and be very easy to expand in the future with new features.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_08	Accessibility	The site should be accessible to the impaired users with accessibility standards.	1	28.05.2023	Alvi Hysa, Indrit Ferati
R_09	Privacy	The website should respect user privacy and be compliant with data protection laws. It should seek user consent regarding data collection and use and be transparent about all the actions with them.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_10	Support	The website should have a helpful and very fast responding support team to help the user with any problem they encounter.	2	10/04/24	Alvi Hysa, Indrit Ferati

## Property Rentals Management System Requirements Specification

R_11	Searchability	The website should have a robust search function that is returns relevant results quickly and accurately.	2	10/04/24	Alvi Hysa, Indrit Ferati
R_12	Performance metrics	The website should have performance metrics that allows admins to monitor the performance and engagement of the website. This could include metrics such as page views, user activity, and user feedback.	3	10/04/24	Alvi Hysa, Indrit Ferati
R_13	Compliance	The website should comply with all the relevant laws and regulations to the specific countries. It should also adhere to industry best practices and guidelines for online learning platforms.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_14	Internationalization	The website should be able to support international users, currencies, time zones and cultural norms.	2	10/04/24	Alvi Hysa, Indrit Ferati
R_15	User authentication and authorization	The website should have a secure authentication system that can include 2-factor authentication.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_16	Payment processing	The website should have a secure processing system for users to pay for the premium services.	1	10/04/24	Alvi Hysa, Indrit Ferati
R_17	Data Backups	The website should have regular data backups to ensure data redundancy.	1	10/04/24	Alvi Hysa, Indrit Ferati

## Property Rentals Management System Requirements Specification

### 3.2.1.1 User Interface Requirements

#### 1. Guest Interface

The logged-in interface serves as the default interface for all users after they login . It provides access to various features and functionalities, including:

- Explore: A dashboard where you can explore all the property rentals
- Profile: A section where users can manage their profile information, including profile picture, bio, and privacy settings.
- Search: A search bar and filters to discover courses based on topic, skill level, instructor, and other criteria.
- Bookings : A way to manage properties you have already booked as well as to view ongoing bookings

#### 2. Host Interface:

- Listing Management: Create, edit, and manage property listings with descriptions, photos, availability, and pricing.
- Booking Management: Review and approve booking requests, communicate with guests, and manage cancellations.
- Calendar Management: Control booking availability and track reservations across different properties (if applicable).
- Analytics & Performance: See insights on listing performance, guest demographics, and potential revenue streams.

#### 3. Admin Panel:

Provides administrative functionalities for managing the platform:

- User Management: Manage user accounts (guests, hosts) and enforce platform policies.
- Listing Management: Review and potentially approve new listings, ensure adherence to platform guidelines.
- Analytics & Reporting: Monitor platform usage, track key metrics (bookings, listings), and generate reports.
- Security & Maintenance: Manage system security, user access controls, and perform system maintenance tasks.

### 3.2.1.2 Usability

#### User Documentation and Help

Complete and Comprehensive: The user documentation should cover all functionalities of the software, including

Listing creation and management (including property details, photos, availability calendar, pricing)

Guest booking and reservation process

Payment processing and refunds

Communication tools for hosts and guests (messaging, messaging history)

User account management (settings, profile information, etc.)

Troubleshooting guides for common issues

Easy to Find and Access: The documentation should be readily available within the software, ideally through a dedicated "Help" section or a searchable knowledge base.

## **Property Rentals Management System Requirements Specification**

**Varied Formats:** Offer the user documentation in multiple formats to cater to different learning styles. This includes text-based guides, video tutorials, or even interactive walkthroughs.

### **Context-Sensitive Help**

**Dynamic and Relevant:** Provide help that automatically pops up or becomes easily accessible depending on the specific task a user is trying to complete. For instance, if a host is editing their listing, context-sensitive help could explain how to add high-quality photos or optimize their listing description.

**Actionable Tips:** Help content should go beyond simply explaining features. It should offer step-by-step instructions and actionable tips to help users achieve their goals efficiently.

### **Overall System Learnability**

**Intuitive Interface:** Design a user interface (UI) that is clear, consistent, and uses common design patterns that users are already familiar with. This will minimize the learning curve and allow users to navigate the software with ease.

#### **Simple Navigation:**

Organize the functionalities of the software in a logical and hierarchical manner.

Implement a clear and consistent navigation bar or menu system that allows users to find what they need quickly.

**Search Functionality:** Include a robust search function that allows users to easily find specific information or functionalities within the software.

### **3.2.1.3 Efficiency**

#### **3.2.1.3.1 Performance Requirements**

##### **Static Requirements: Setting the Stage**

**Number of Listings:** We've designed the system to handle at least 1000 active listings. This number can be adjusted based on our target market size and projected growth. We can easily scale up as needed to accommodate more properties.

**User Accounts:** The software can comfortably store information for 100,000 user accounts, including both hosts and guests. This number factors in potential future growth and keeps us prepared for a bustling user base.

**Data Storage Capacity:** We've ensured the system has ample storage for all the information it needs to handle, including:

Listing details (text, photos, videos) with a minimum capacity of 10 terabytes.

User data (profiles, communication history) sized at an estimated 5000 gigabytes.

##### **Dynamic Requirements: Keeping Up with the Flow**

**Speedy Searches:** Under normal usage conditions, at least 80% of search queries should return results within 0.8 seconds. This ensures a fast and responsive experience for users browsing listings. No one wants to wait ages for search results!

**Swift Booking Transactions:** During peak usage times (weekends, holidays), we aim for 95% of booking transactions to process in less than 3 seconds. This minimizes wait times and frustration for users finalizing their reservations. A smooth booking experience is key to happy users.

**High System Availability:** We've designed the software to be highly available, with an uptime of at least 99%. This translates to minimal downtime for essential functionalities. We want the platform to be reliable and accessible whenever users need it.

**Scalable User Onboarding:** The system can handle 500 new user registrations per day during peak periods. This ensures a smooth onboarding experience for new users joining the platform. We want everyone to feel welcome and have a seamless start on our platform.

##### **Reliability: Keeping the Platform Up and Running**

##### **Monitoring: Keeping a Watchful Eye**

## Property Rentals Management System Requirements Specification

To ensure our software is reliable, we've implemented robust monitoring systems that track various aspects of its health. Here's what we monitor:

**Product/Service Health:** We continuously monitor key performance indicators (KPIs) like system uptime, response times, and error rates. This allows us to identify potential issues before they impact users.

**Failure Conditions:** The system is designed to detect failures automatically. This includes monitoring for server crashes, database errors, and network disruptions.

**Error Detection and Logging:** All errors are logged in detail, providing valuable information for troubleshooting and identifying root causes. This helps us fix issues efficiently and prevent them from recurring.

**Correction Mechanisms:** We've built-in mechanisms for automated recovery in case of minor failures. Additionally, the monitoring system alerts us in case of critical issues requiring manual intervention.

**Maintenance:** Designed for Ease

We understand that maintaining a complex software system is essential. That's why we've prioritized maintainability from the start:

**Modularity:** The software is designed with modular components, making it easier to isolate issues, update specific parts, and avoid unintended consequences.

**Reduced Complexity:** We strive for a clean and well-documented codebase, minimizing complexity for future maintenance and troubleshooting.

**Intuitive Interface:** The system's architecture and code are designed with maintainability in mind. This ensures developers can easily understand the system and make modifications when needed.

Remember, these are not just good practices, they're crucial for a reliable platform. By proactively monitoring and maintaining the software, we can minimize downtime and ensure a smooth user experience.

**Integrity: Protecting Your Data**

**Security: Building a Fortress**

We take data security very seriously. Here are the measures in place to protect your information:

**Encryption:** All sensitive data, such as user passwords and financial information, are encrypted at rest and in transit. This makes it virtually impossible for unauthorized parties to access it.

**Activity Logging:** We maintain detailed logs of all user activity. This allows us to track access attempts, identify suspicious behavior, and investigate potential security breaches.

**Historical Data Sets:** We retain historical data sets for security purposes. This allows us to analyze past activity and identify patterns that could indicate a security threat.

**Restrictions on Communication:** Access to sensitive data is strictly controlled. We implement restrictions on how different parts of the system can communicate with each other, minimizing the risk of unauthorized access.

**Data Integrity Checks:** The system continuously performs data integrity checks to ensure the accuracy and consistency of stored information. This helps us detect any unauthorized modifications or data corruption attempts.

**Authorization and Authentication: Who Gets In?**

We employ industry-standard tools and practices for user authorization and authentication. This ensures only authorized users can access specific features and information within the platform.

**Authentication:** Users are required to prove their identity before accessing the system. This is typically done through a username and password combination or other secure methods like multi-factor authentication.

**Authorization:** Once authenticated, users are granted specific access levels based on their role (host, guest, administrator). This ensures users can only access and modify information relevant to their role.

While tools like PubCookie can be considered, we may explore a broader range of secure authentication methods to best suit the platform's needs.

By implementing these security measures, we strive to create a safe and secure environment for both hosts and guests using our platform.

### **3.2.2 Organizational Requirements: Aligning with Internal Processes**

Our software development process adheres to established organizational policies and procedures. Here's how we ensure a smooth fit:

**Process Standards:** We follow industry best practices for software development methodologies like Agile or Waterfall, depending on project needs. This ensures efficient development, testing, and deployment processes.

**Implementation Requirements:** We have defined clear guidelines for system implementation, including infrastructure setup, user training, and data migration plans. This minimizes disruption and ensures a smooth transition for the organization.

**Change Management Procedures:** We adhere to established change management procedures to manage system updates and modifications. This minimizes risk and ensures all stakeholders are informed about upcoming changes.

#### **3.2.2.1 Environmental Requirements: Considering the System's Surroundings**

The software's operational environment plays a crucial role. Here's what we consider:

**Hardware and Software Compatibility:** The software is designed to be compatible with a defined range of hardware and software configurations. This ensures smooth operation within the existing IT infrastructure.

**Performance Requirements:** We establish performance benchmarks that consider factors like server capacity, network bandwidth, and user load. This ensures the software meets performance expectations within its operational environment.

**Scalability:** The system architecture is designed to be scalable, allowing us to accommodate future growth in terms of users, listings, and data volume. This ensures the platform can adapt to the organization's evolving needs.

#### **3.2.2.2 Operational Requirements: Streamlining Day-to-Day Use**

Our software is designed to be operationally efficient:

**System Monitoring and Alerting:** Monitoring systems are integrated to track system health, performance, and security. They generate alerts for potential issues, allowing for proactive intervention by the IT team.

**Backup and Recovery Procedures:** We establish comprehensive backup and recovery procedures to ensure minimal data loss and downtime in case of failures. This includes regular data backups and disaster recovery plans.

**User Support:** We provide user support channels such as documentation, FAQs, and a ticketing system. This allows users to resolve issues independently or get assistance from the support team.

#### **3.2.2.3 Development Requirements: Building a Maintainable System**

From the start, we prioritize maintainability:

**Code Quality:** Our developers follow coding standards and best practices to ensure clean, well-documented, and maintainable code. This facilitates future modifications and reduces the risk of errors.

**Modular Design:** The software is designed with modular components, making it easier to isolate issues, update specific parts, and avoid unintended consequences during maintenance.

**Version Control:** We employ a version control system to track code changes and facilitate rollbacks in case of issues. This allows for efficient development and collaboration.

### **3.2.3 External Requirements: Playing by the Rules**

The software needs to comply with various external factors:

#### **3.2.3.1 Regulatory Requirements**

## **Property Rentals Management System Requirements Specification**

We ensure the software adheres to relevant regulations depending on the target market and industry. This may include data privacy regulations like GDPR (General Data Protection Regulation) or financial regulations.

### **3.2.3.2 Ethical Requirements**

We uphold ethical principles in the design and development of the software. This includes protecting user privacy, promoting fair treatment of all participants in the rental marketplace, and avoiding discriminatory practices.

### **3.2.3.3 Legislative Requirements**

The software complies with relevant laws governing areas like consumer protection, data security, and fair housing practices. This may involve specific requirements for data reporting, recordkeeping, and user authentication.

#### **3.2.3.3.1 Accounting Requirements**

The software facilitates accurate and transparent financial transactions:

**Transaction Recording:** All financial transactions (rent payments, fees, refunds) are recorded in detail, ensuring accurate financial reporting and compliance with accounting standards.

**Audit Trail:** We implement an audit trail that tracks all financial activity within the system. This allows for reconciliation, fraud detection, and meeting any specific audit requirements.

#### **3.2.3.3.2 Security Requirements**

Security is paramount, and the software adheres to industry best practices:

**Data Encryption:** Sensitive data like user passwords and financial information are encrypted at rest and in transit. This ensures data confidentiality and minimizes the risk of unauthorized access.

**Access Controls:** We implement robust access control mechanisms that restrict access to sensitive data based on user roles and permissions. This ensures only authorized users can modify or view specific information.

## **3.3 Domain Requirements**

Here, we delve into the non-functional requirements specific to the rental property domain, ensuring our software caters to the unique needs of this market:

### **User Interface (UI) and User Experience (UX) Requirements**

**Intuitive Interface:** The user interface (UI) should be clean, intuitive, and user-friendly for both hosts and guests with varying levels of technical expertise. This includes clear navigation, easy-to-understand icons, and consistent design patterns.

**Mobile Responsiveness:** The software should be responsive and function seamlessly across different devices, especially mobile phones and tablets. This caters to users searching for rentals or managing their properties on the go.

**Multilingual Support:** The software should ideally support multiple languages to cater to a global user base. This can include UI translation, currency conversion tools, and the ability for hosts to list properties in their native language.

**Search Functionality:** A robust search function is crucial, allowing users to filter listings based on various criteria like location, price range, amenities, property type, and availability. This ensures users can efficiently find properties that meet their exact needs.

## **Property Rentals Management System Requirements Specification**

**Photo and Video Quality:** High-quality photos and videos are essential for showcasing rental properties effectively. The software should support uploading clear and visually appealing media content to attract potential guests.

### **Usability Requirements**

**Listing Creation and Management:** The process of creating, editing, and managing listings should be straightforward and efficient. This includes clear instructions, user-friendly forms, and the ability to upload photos and descriptions with ease.

**Booking and Reservation Management:** Both hosts and guests should be able to manage bookings and reservations conveniently. This includes tools for calendar synchronization, communication channels, and secure online payment processing.

**Review and Rating System:** A review and rating system allows users to share their experiences, fostering trust and transparency within the platform. It should be easy for both hosts and guests to leave reviews and ratings.

**Accessibility:** The software should be accessible to users with disabilities, adhering to accessibility guidelines like WCAG (Web Content Accessibility Guidelines). This ensures everyone can enjoy a seamless user experience.

### **Data Requirements**

**Data Security:** As mentioned previously, data security is paramount. We need to ensure user information, financial data, and property details are protected through encryption, access controls, and secure data storage practices.

**Data Privacy:** The software should comply with data privacy regulations. This includes transparent data collection practices, user consent for data usage, and mechanisms for users to access and control their personal information.

**Data Backup and Recovery:** Regular data backups are crucial for disaster recovery purposes. The software should have a robust backup and recovery plan to minimize data loss in case of system failures.

### **Performance Requirements**

While we covered general performance requirements earlier, here are some specific considerations for the rental marketplace:

**Search Performance:** Search queries should return results quickly, especially during peak seasons or high-demand periods. This ensures a smooth user experience for browsing listings.

**Calendar Synchronization:** Calendar synchronization between the software and external calendars (e.g., Google Calendar) should be seamless and reliable to avoid double bookings.

**Image and Video Loading:** Images and videos showcasing rental properties should load quickly to prevent frustration and keep users engaged.

## **4. User Scenarios/Use Cases**

➤ **Browse available rental properties to find a place to stay.**

1. Search Properties: Users search for rental properties based on various criteria.
2. View Property Details: Users search for rental properties based on various criteria.
3. Comparison: User enters criteria to get the best property.
4. Save Favorites: Users save preferred properties to a list for future reference.
5. Contact Property Owner/Agent: Users initiate communication with property owners or agents regarding rental properties.

➤ **View reviews and ratings for rental properties to make informed decisions.**

6. Browse Reviews: Users can browse through a list of reviews and ratings for various rental properties.
7. Filter Reviews by Rating: Users can filter reviews based on the rating given by previous tenants.
8. Sort Reviews by Date: Users can sort reviews based on the date they were posted.
9. Rate and Review Properties: Users can leave their own ratings and reviews for rental properties they have stayed in, contributing to the overall feedback pool.
10. Report Inappropriate Reviews: Users can flag reviews that they believe are inappropriate or violate community guidelines for moderation.

➤ **Listing properties for rent on the app to share it with potential guests.**

11. Create Property Listing: Hosts can easily create property listings through an intuitive interface
12. Edit Property Listing: Hosts can update their property listing
13. Update Account Information: Hosts can manage their personal and financial account details.
14. Delete Property Listing: Delete Property Listing

➤ **Manage the property's availability calendar to ensure double bookings are avoided.**

15. Process Secure Financial Transactions: Hosts should be able to securely process rental payments, security deposits, and additional fees through the platform.
16. Integrate Payment Gateways: The system should integrate with reputable payment gateways to facilitate secure financial transactions.
17. Display Transaction Fees: The system should transparently display transaction fees associated with rental payments, security deposits, and additional fees.
18. Manage Security Deposits: The system should provide an escrow service to hold security deposits until the end of the rental period.
19. Ensure Regulatory Compliance: The system should adhere to relevant financial regulations and standards to ensure legal compliance.

## **Property Rentals Management System Requirements Specification**

- **Reviewing and approving new properties before they become available to guests.**
- 20. Property Review Interface: The system must provide an interface for admins to review new properties.
- 21. Notification System for Property Status Update: The system should notify hosts about the status of their properties after admin review.
- 22. Admin Review Queue Management: The system should prioritize the review queue for admins based on certain criteria.
- 23. Feedback Mechanism for Hosts: The system should allow hosts to provide feedback or seek clarification on admin decisions regarding their properties.
- 24. Reporting System for Admin Performance: The system should provide reporting tools for evaluating admin performance in reviewing properties.

- **Generating financial reports to track the app's performance and user activity**

- 25. Report Generation Tool: The system provides an interface for admins to generate financial reports for various timeframes.
- 26. Revenue Analysis: The system generates financial reports that include specific financial metrics.
- 27. Report Download and Sharing: The system allows admins to download and share reports.
- 28. User Books Property: The user is able to create a booking for a certain property for a required time
- 29. Leaving Reviews: The user is able to leave a review after a booking

1.

UC Name	<i>Search Properties</i>
Summary	<i>Users search for rental properties based on various criteria.</i>
Dependency	<i>None</i>
Actors	<i>User</i>
Preconditions	<i>User is logged into the system.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• 1. User enters search criteria such as location, price range, and property type.</li> <li>• 2. System views search criteria.</li> <li>• 3. System displays search results to the user.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• None</li> </ul>
Non functional requirements	<i>Performance: Search results should be displayed within a reasonable time frame, typically under 5 seconds. Security: User data entered during the search process should be securely handled and stored.</i>
Postconditions	<i>User is presented with a list of rental properties matching the search criteria.</i>

## Property Rentals Management System Requirements Specification

UC Name	<i>View Property Details</i>
Summary	<i>Users search for rental properties based on various criteria.</i>
Dependency	<i>None</i>
Actors	<i>User</i>
Preconditions	<i>User has initiated a search for rental properties.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• 1. User selects a rental property for detailed information.</li> <li>• 2. System views selected property.</li> <li>• 3. System displays the property details to the user.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• None</li> </ul>
Non-functional requirements	<i>Performance: Property details should load quickly, typically within 2 seconds.</i>
Postconditions	<i>User is presented with a list of rental properties matching the search criteria.</i>

## Property Rentals Management System Requirements Specification

UC Name	Comparison
Summary	<i>User enters criteria to get the best property.</i>
Dependency	<i>None</i>
Actors	<i>User</i>
Preconditions	<i>User has initiated a search for rental properties.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• 1. User enters criteria such as no of rooms, amenities, price range etc.</li> <li>• 2. System compares properties based on the criteria.</li> <li>• 3. System displays the best property that complies with the criteria.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• None</li> </ul>
Non functional requirements	<i>Performance: Filtered search results should be generated within a reasonable time frame, typically under 3 seconds.</i>
Postconditions	<i>User is presented with search results that match the selected filters.</i>

**Property Rentals Management System Requirements Specification**

<b>UC Name</b>	Save Favorites
<b>Summary</b>	<i>Users save preferred properties to a list for future reference.</i>
<b>Dependency</b>	<i>View Property Details</i>
<b>Actors</b>	User
<b>Preconditions</b>	<i>User has viewed property details of at least one rental property.</i>
<b>Description of the Main Sequence</b>	<ul style="list-style-type: none"> <li>• 1. User selects the option to save a rental property as a favorite.</li> <li>• 2. System adds the selected property to the user's list of favorites.</li> </ul>
<b>Description of the Alternative Sequence</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Non functional requirements</b>	<i>None</i>
<b>Postconditions</b>	<i>Selected rental property is successfully saved to the user's list of favorites.</i>

UC Name	<i>Contact Property Owner/Agent</i>
Summary	<i>Users initiate communication with property owners or agents regarding rental properties.</i>
Dependency	<i>View Property Details</i>
Actors	<i>User, Property Owner/Agent</i>
Preconditions	<i>User has viewed property details of at least one rental property.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>● 1. User selects the option to contact the property owner/agent.</li> <li>● 2. System provides contact options such as email, phone number, or contact form.</li> <li>● 3. User initiates communication using the provided contact information.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>● None</li> </ul>
Non functional requirements	<i>None</i>
Postconditions	<i>User has successfully initiated communication with the property owner/agent.</i>

2.

UC Name	<i>Browse Reviews</i>
Summary	<i>Users can browse through a list of reviews and ratings for various rental properties.</i>
Dependency	<i>None</i>
Actors	<i>User</i>
Preconditions	<i>User is logged into the system.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: User navigates to the reviews section of the property listing.</li> <li>• Step 2: System displays a list of available reviews and ratings for rental properties.</li> <li>• Step 3: User browses through the reviews</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• Step 1: No reviews are available for the property.</li> <li>• Step 2: System displays a message to indicate that there are no reviews available</li> </ul>
Non functional requirements	<i>Performance: Reviews should load quickly, typically within 3 seconds.</i>
Postconditions	<i>User can view a variety of reviews and ratings for rental properties.</i>

UC Name	<i>Filter Reviews by Rating</i>
Summary	<i>Users can filter reviews based on the ratings given by previous tenants.</i>
Dependency	<i>Browse Reviews</i>
Actors	<i>User</i>
Preconditions	<i>User has browsed through available reviews.</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Step 1. User chooses "Filter by ratings".</li><li>• Step 2: User selects the specific rating range.</li><li>• Step 3: System retrieves the reviews that match the selected rating criteria.</li><li>• Step 4: System displays reviews.</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• None</li></ul>
Non functional requirements	<i>None</i>
Postconditions	<i>User can view reviews filtered by the selected rating range.</i>

**Property Rentals Management System Requirements Specification**

<b>UC Name</b>	<i>Sort Reviews by Date</i>
<b>Summary</b>	<i>Users can sort reviews based on the date they were posted.</i>
<b>Dependency</b>	<i>Browse Reviews</i>
<b>Actors</b>	<i>User</i>
<b>Preconditions</b>	<i>User has browsed through available reviews.</i>
<b>Description of the Main Sequence</b>	<ul style="list-style-type: none"> <li>• Step 1. User chooses “Filter by date”.</li> <li>• Step 2: User clicks the “Newest” button.</li> <li>• Step 3: System arranges them from most recent to oldest.</li> <li>• Step 4: System displays reviews.</li> </ul>
<b>Description of the Alternative Sequence</b>	<ul style="list-style-type: none"> <li>• Step 1: User clicks the “Oldest” button.</li> <li>• Step 2: System arranges them from oldest to newest.</li> <li>• Step 3: System displays reviews.</li> </ul>
<b>Non functional requirements</b>	<i>None</i>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User selects the option to sort reviews by date.</li> <li>2. System rearranges the reviews to display the most recent ones first.</li> </ol>

UC Name	<i>Rate and Review a Property</i>
Summary	<i>Users can leave their own ratings and reviews for rental properties they have stayed in, contributing to the overall feedback pool.</i>
Dependency	<i>Browse Reviews</i>
Actors	<i>User</i>
Preconditions	<i>User has rented the property, is a previous tenant.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: User clicks "Write a review."</li> <li>• Step 2: System shows a form.</li> <li>• Step 3: User writes in the textarea to provide feedback.</li> <li>• Step 4: User chooses stars to rate based on experience.</li> <li>• Step 5: Users submit form.</li> <li>• Step 6: System saves the review.</li> <li>• Step 7: System shows message that review was created.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• Step 1: User input is not valid (has left a field empty or text is too short)</li> <li>• Step 2: System shows error message.</li> <li>• Step 3: System shows the form again</li> </ul>
Non functional requirements	<i>None</i>
Postconditions	<i>User's rating and review are added to the feedback pool for the respective rental property.</i>

UC Name	<i>Report Inappropriate Reviews</i>
Summary	<i>Users can flag reviews that they believe are inappropriate or violate community guidelines for moderation.</i>
Dependency	<i>View Property Details</i>
Actors	<i>User, Property Owner/Agent</i>
Preconditions	<i>User has browsed through available reviews.</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li>• Step 1: User identifies a review that they believe is inappropriate or violates community guidelines.</li><li>• Step 2: User selects the option to report the review.</li><li>• Step 3: System prompts the user to provide a reason for reporting.</li><li>• Step 4: User fills out the form.</li><li>• Step 4: User submits the form.</li></ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"><li>• None</li></ul>
Non functional requirements	<i>None</i>
Postconditions	<i>Reported review is flagged for moderation by the system administrators.</i>

## Property Rentals Management System Requirements Specification

3.

UC Name	<i>Create Property Listing</i>
Summary	<i>Hosts can easily create property listings</i>
Dependency	<i>Manage property listings</i>
Actors	<i>Host</i>
Preconditions	<i>Host is logged into the system</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: Host navigates to the "Create Listing" section.</li> <li>• Step 2: Host uploads photos of the property.</li> <li>• Step 3: Host fills in detailed descriptions of the property.</li> <li>• Step 4: Host sets rental price and additional fees.</li> <li>• Step 5: Host specifies available dates and stay requirements.</li> <li>• Step 6: Host submits the form</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• Step 1: The system prompts the host to correct the errors and resubmit the listing.</li> <li>• Step 2: Host revises the listing information accordingly and resubmits.</li> </ul>
Non functional requirements	<i>User interface responsiveness, accessibility compliance</i>
Postconditions	<i>Property listings are successfully managed based on the host's actions.</i>

UC Name	<i>Edit Property Listing</i>
Summary	<i>Hosts can update their property listing</i>
Dependency	<i>Create Property Listing</i>
Actors	<i>Host</i>
Preconditions	<i>Host has created a property listing</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: Host has navigated to the property listing he wants to edit.</li> <li>• Step 2: Host enters the new data in the necessary fields.</li> <li>• Step 3: Host submits the form</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• Step 1: The system prompts the host to correct the errors and resubmit the listing.</li> <li>• Step 2: Host revises the listing information accordingly and resubmits</li> </ul>
Non functional requirements	<i>User interface responsiveness, accessibility compliance</i>
Postconditions	<i>Property listing is successfully modified and displayed on the app</i>

## Property Rentals Management System Requirements Specification

UC Name	<i>Update Account Information</i>
Summary	<i>Hosts can manage their personal and financial account details.</i>
Dependency	<i>None</i>
Actors	<i>Host</i>
Preconditions	<i>Host is logged into the system</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: Host navigates to the "Account Settings" section.</li> <li>• Step 2: Host selects the type of information they want to update:</li> <li>• Step 3: Host may update name, email, contact info, password, or modifies profile picture.</li> <li>• Step 4: Host may add a new bank account, updates existing bank account information</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• Step 1: The system validates the data put and detects it as invalid.</li> <li>• Step 2: The system displays an error message to the host.</li> <li>• Step 3: Host corrects the error and resubmits the update.</li> <li>• Step 4: The system successfully updates the account information with the corrected data.</li> <li>• Step 5: Alternatively, if the host cannot correct the error, they may choose to cancel the update and return to the previous account information.</li> </ul>
Non-functional requirements	<i>User interface responsiveness, accessibility compliance</i>
Postconditions	<i>Host successfully updates their account information.</i>

**Property Rentals Management System Requirements Specification**

<b>UC Name</b>	<i>Delete Property Listing</i>
<b>Summary</b>	<i>Host can delete their property listing</i>
<b>Dependency</b>	<i>Contact Property Owner</i>
<b>Actors</b>	<i>Host</i>
<b>Preconditions</b>	<i>Host is logged into the system</i>
<b>Description of the Main Sequence</b>	<ul style="list-style-type: none"> <li>• Step 1: Host has navigated to the property listing he wants to delete.</li> <li>• Step 2: Host clicks the button of deletion</li> <li>• Step 3: Host clicks "Yes" on the confirmation dialog</li> <li>• Step 4: Property listing is deleted</li> </ul>
<b>Description of the Alternative Sequence</b>	<ul style="list-style-type: none"> <li>• Step 1: Host clicks "Cancel" button</li> <li>• Step 2: Host is sent back to all property listings page</li> </ul>
<b>Non functional requirements</b>	
<b>Postconditions</b>	<i>Effective communication is maintained between hosts and guests, ensuring a positive guest experience.</i>

4.

UC Name	<i>Process Secure Financial Transactions</i>
Summary	<i>Hosts should be able to securely process rental payments, security deposits, and additional fees through the platform.</i>
Dependency	<i>None</i>
Actors	<i>Host, Payment Gateway</i>
Preconditions	<i>Host must have an active listing, and guests must have provided payment information.</i>
Description of the Main Sequence	<ol style="list-style-type: none"> <li>1. Host initiates the payment process for rental payments, security deposits, or additional fees.</li> <li>2. The system securely communicates with the payment gateway to process the transaction.</li> <li>3. Payment gateway verifies payment details and processes the transaction securely.</li> <li>4. The system updates the transaction status and notifies the host and guest of the successful transaction.</li> </ol>
Description of the Alternative Sequence	<i>If the payment details are invalid or the transaction fails, the system provides appropriate error messages and prompts the host to retry or choose an alternative payment method.</i>
Non functional requirements	<i>Security: Payment processing must comply with industry-standard security protocols (e.g., PCI DSS) to protect financial data.</i>
Postconditions	<i>Transaction status is updated in the system, and both host and guest receive confirmation of the transaction.</i>

UC Name	<i>Integrate Payment Gateways</i>
Summary	<i>The system should integrate with reputable payment gateways to facilitate secure financial transactions.</i>
Dependency	<i>None</i>
Actors	<i>System Admin, Host, Payment Gateway</i>
Preconditions	<i>None</i>
Description of the Main Sequence	<ol style="list-style-type: none"> <li>1. <i>System admin configures integration settings for selected payment gateways.</i></li> <li>2. <i>Host selects preferred payment gateway for processing transactions.</i></li> <li>3. <i>Payment gateway securely handles payment transactions between guests and hosts.</i></li> </ol>
Description of the Alternative Sequence	<i>If the selected payment gateway is unavailable or encounters errors, the system prompts the host to select an alternative payment gateway.</i>
Non functional requirements	<i>Reliability: Payment gateway integration should be reliable and minimize downtime to ensure seamless transactions.</i>
Postconditions	<i>Successful integration with selected payment gateways, enabling hosts to process transactions.</i>

UC Name	<i>Display Transaction Fees</i>
Summary	<i>The system should transparently display transaction fees associated with rental payments, security deposits, and additional fees.</i>
Dependency	<i>None</i>
Actors	<i>Host, Guest</i>
Preconditions	<i>Host has initiated a transaction.</i>
Description of the Main Sequence	<ol style="list-style-type: none"> <li><i>System calculates applicable transaction fees based on the transaction amount and fee structure.</i></li> <li><i>Transaction fees are displayed to the host and guest before confirming the transaction.</i></li> </ol>
Description of the Alternative Sequence	<i>If there are no applicable transaction fees, the system indicates that no fees will be charged.</i>
Non functional requirements	<i>Transparency: Transaction fees must be clearly presented to ensure transparency and avoid surprises for hosts and guests.</i>
Postconditions	<i>Host and guest are aware of the transaction fees before completing the transaction.</i>

UC Name	<i>Manage Security Deposits</i>
Summary	<i>The system should provide an escrow service to hold security deposits until the end of the rental period.</i>
Dependency	<i>Secure Payment Processing</i>
Actors	<i>Host, Guest</i>
Preconditions	<i>Host has specified a security deposit requirement for the listing.</i>
Description of the Main Sequence	<ol style="list-style-type: none"> <li>1. <i>Guest submits the security deposit amount during the booking process.</i></li> <li>2. <i>System securely holds the security deposit amount in escrow until the end of the rental period.</i></li> <li>3. <i>Upon successful completion of the rental period, the system releases the security deposit to the host.</i></li> <li>4. <i>In case of damage or disputes, the system facilitates resolution and manages the release of the security deposit accordingly.</i></li> </ol>
Description of the Alternative Sequence	<i>If there are disputes regarding damages, the system may involve a mediation process or arbitration to resolve the issue.</i>
Non functional requirements	<i>Trustworthiness: The escrow service must be reliable and instill trust between hosts and guests by ensuring fair handling of security deposits.</i>
Postconditions	<i>Security deposit is released to the host upon completion of the rental period, or as per the resolution of any disputes.</i>

UC Name	<i>Ensure Regulatory Compliance</i>
Summary	<i>The system should adhere to relevant financial regulations and standards to ensure legal compliance.</i>
Dependency	<i>Secure Payment Processing, Integration with Payment Gateways</i>
Actors	<i>System Admin, Legal Compliance Team</i>
Preconditions	<i>None</i>
Description of the Main Sequence	<ol style="list-style-type: none"> <li>1. <i>System admin and legal compliance team review and ensure adherence to applicable financial regulations and standards.</i></li> <li>2. <i>System implements necessary measures and protocols to comply with regulations, such as data protection laws.</i></li> </ol>
Description of the Alternative Sequence	<i>If there are changes in financial regulations, the system promptly updates its processes and procedures to maintain compliance.</i>
Non functional requirements	<i>Compliance: The system must continuously monitor and adapt to changes in financial regulations to avoid legal risks and penalties.</i>
Postconditions	<i>System operations remain compliant with relevant financial regulations, minimizing legal risks for the platform and its users.</i>

## Property Rentals Management System Requirements Specification

5.

UC Name	<i>UR-5.1. Property Review Interface</i>
Summary	<i>The system must provide an interface for admins to review new properties.</i>
Dependency	<i>None</i>
Actors	<i>Admin</i>
Preconditions	<i>Admin is logged into the system.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: Admin accesses the admin dashboard.</li> <li>• Step 2: Admin navigates to the section for reviewing new properties.</li> <li>• Step 3: Admin views details of each property, including photos, descriptions, and pricing.</li> <li>• Step 4: Admin approves or rejects each property based on completeness and adherence to guidelines.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
Non-functional requirements	<p><i>The property review interface should load quickly to ensure efficient processing of new properties. The interface should be accessible from various devices and browsers to accommodate the admin's preferences.</i></p> <p><i>It should have a responsive design to ensure usability on different screen sizes.</i></p> <p><i>Admins should be able to sort and filter properties based on criteria like submission date or property status to manage them effectively.</i></p>
Postconditions	<i>All approved properties are made available for guests to view and book, while rejected properties are returned to the host for revisions.</i>

UC Name	<i>UR-5.2 Notification System for Property Status Update</i>
Summary	<i>The system should notify hosts about the status of their properties after admin review.</i>
Dependency	<i>UR-5.1</i>
Actors	<i>System, Hosts.</i>
Preconditions	<i>Admin has reviewed and decided on the property.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: After admin review, the system updates the status of the property.</li> <li>• Step 2: The system sends a notification to the host informing them of the review outcome.</li> <li>• Step 3: Host receives the notification and views the updated status of their property</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
Non-functional requirements	<p><i>Notifications should be delivered in real time to ensure timely communication with hosts.</i></p> <p><i>The system should support multiple channels for notifications, including email and app notifications.</i></p> <p><i>Notifications should include relevant information such as property approval status and any required actions from hosts.</i></p>
Postconditions	<i>Hosts are informed about the status of their property, allowing them to take necessary actions based on the review outcome.</i>

UC Name	<i>UR-5.3 Admin Review Queue Management</i>
Summary	<i>The system should prioritize the review queue for admins based on certain criteria.</i>
Dependency	<i>None</i>
Actors	<i>System, Admin</i>
Preconditions	<i>Admin is logged into the system.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: The system organizes new property submissions into a review queue.</li> <li>• Step 2: The system prioritizes the review queue based on factors like submission date, property completeness, and urgency(it may be indicated that a property is available for a limited time or a specific event).</li> <li>• Step 3: Admin accesses the review queue and begins reviewing properties according to the prioritization.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
Non-functional requirements	<p><i>The system should dynamically adjust the prioritization of the review queue based on time sensitivity(availability of the property, upcoming events, or special promotions). Admins should be able to manually rearrange the order of properties in the queue if needed.</i></p> <p><i>The system should provide estimated review times for hosts to manage their expectations.</i></p>
Postconditions	<i>Admins effectively manage the review process, ensuring timely approval or rejection of properties.</i>

UC Name	<i>UR-5.4 Feedback Mechanism for Hosts</i>
Summary	<i>The system should allow hosts to provide feedback or seek clarification on admin decisions regarding their properties.</i>
Dependency	<i>UR-5.1</i>
Actors	<i>Hosts, Admin</i>
Preconditions	<i>The host has received a decision on their property from the admin.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: Host receives notification of the admin decision on their property.</li> <li>• Step 2: The host has the option to provide feedback or seek clarification on the decision(expressing satisfaction or dissatisfaction with the decision, providing suggestions for improvement, or offering additional information relevant to their property.).</li> <li>• Step 3: Admin receives the feedback or clarification request and responds accordingly.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
Non-functional requirements	<p><i>The feedback mechanism should be easily accessible and intuitive for hosts to use.</i></p> <p><i>Admins should receive notifications for new feedback or clarification requests and respond promptly.</i></p> <p><i>The system should maintain a record of all communication between hosts and admins for transparency and dispute resolution purposes.</i></p>
Postconditions	<i>Hosts receive clarification or resolution regarding their concerns, ensuring transparency and satisfaction with the review process.</i>

UC Name	<i>UR-5.5: Reporting System for Admin Performance</i>
Summary	<i>The system should provide reporting tools for evaluating admin performance in reviewing properties.</i>
Dependency	<i>UR-5.1</i>
Actors	<i>System, Admin</i>
Preconditions	<i>Admin has reviewed properties</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Step 1: The system collects data on admin actions during the property review process.</li> <li>• Step 2: The system generates performance reports based on metrics such as review time, accuracy of decisions (The percentage of decisions made by admins that align with platform guidelines and host expectations), and feedback from hosts (The percentage of decisions made by admins that align with platform guidelines and host expectations).</li> <li>• Step 3: Admin accesses the performance reports to evaluate their efficiency and effectiveness in managing properties.</li> </ul>
Description of the Alternative Sequence	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
Non-functional requirements	<p><i>Reporting tools should provide real-time data to enable admins to track their performance continuously.</i></p> <p><i>Reports should be customizable, allowing admins to filter data based on specific periods or criteria (Analyze performance data for different periods (e.g., daily, weekly, monthly) or filter data based on specific criteria (e.g., by admin ID, property category)).</i></p> <p><i>The system should ensure the security and confidentiality of admin performance data to maintain trust and integrity.</i></p>
Postconditions	<i>Admins have access to performance insights to identify areas for improvement and maintain quality standards in the property review process.</i>

## Property Rentals Management System Requirements Specification

6.

UC Name	<i>SR-1: Report Generation Tool</i>
Summary	<i>The system provides an interface for admins to generate financial reports for various timeframes.</i>
Dependency	<i>None</i>
Actors	<i>Admin</i>
Preconditions	<i>Admin is logged into the system.</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• Admin accesses the reporting section in the admin dashboard.</li> <li>• The interface displays options to select the desired reporting timeframe (e.g., daily, weekly, monthly, custom date range).</li> <li>• Admin selects the timeframe and any additional filter options (if applicable).</li> <li>• The system generates a report based on the selected criteria.</li> <li>• Admin can view the report within the interface.</li> </ul>
Description of the Alternative Sequence	<i>N/A</i>
Non functional requirements	<p><i>The interface should be user-friendly and easy to navigate for admins.</i></p> <p><i>Report generation should be efficient and produce results within a reasonable timeframe.</i></p> <p><i>The system should be secure and restrict access to financial reports to authorized admins.</i></p>
Postconditions	<i>Admin has access to a financial report for the chosen timeframe, enabling analysis of the app's performance.</i>

UC Name	<i>SR-2: Revenue Analysis</i>
Summary	<i>The system generates financial reports that include specific financial metrics.</i>
Dependency	<i>SR-1 (Financial Reporting Tool)</i>
Actors	<i>System</i>
Preconditions	<i>Admin has generated a report (using SR-1).</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• <i>The system retrieves relevant financial data from the database.</i></li> <li>• <i>The system aggregates data based on the selected timeframe in the report (SR-1).</i></li> <li>• <i>The report includes the following financial metrics:</i></li> <li>• <i>Total number of bookings</i></li> <li>• <i>Total number of cancellations</i></li> <li>• <i>Revenue generated from rentals</i></li> <li>• <i>Revenue generated from fees (e.g., platform fees, service charges)</i></li> </ul>
Description of the Alternative Sequence	<i>N/A</i>
Non functional requirements	<i>Financial data calculations should be accurate and reliable. The report should clearly present each metric with proper labels and units.</i>
Postconditions	<i>The financial report provides admins with key financial insights into the app's performance.</i>

UC Name	<i>SR-3: Report Download and Sharing</i>
Summary	<i>The system allows admins to download and share reports.</i>
Dependency	<i>SR-1 (Financial Reporting Tool)</i>
Actors	<i>Admin</i>
Preconditions	<i>Admin has generated a report (using SR-1).</i>
Description of the Main Sequence	<ul style="list-style-type: none"><li><i>Admin views the generated financial report.</i></li><li><i>The interface provides</i></li></ul>
Description of the Alternative Sequence	<i>N/A</i>
Non functional requirements	<i>N/A</i>
Postconditions	<i>N/A</i>

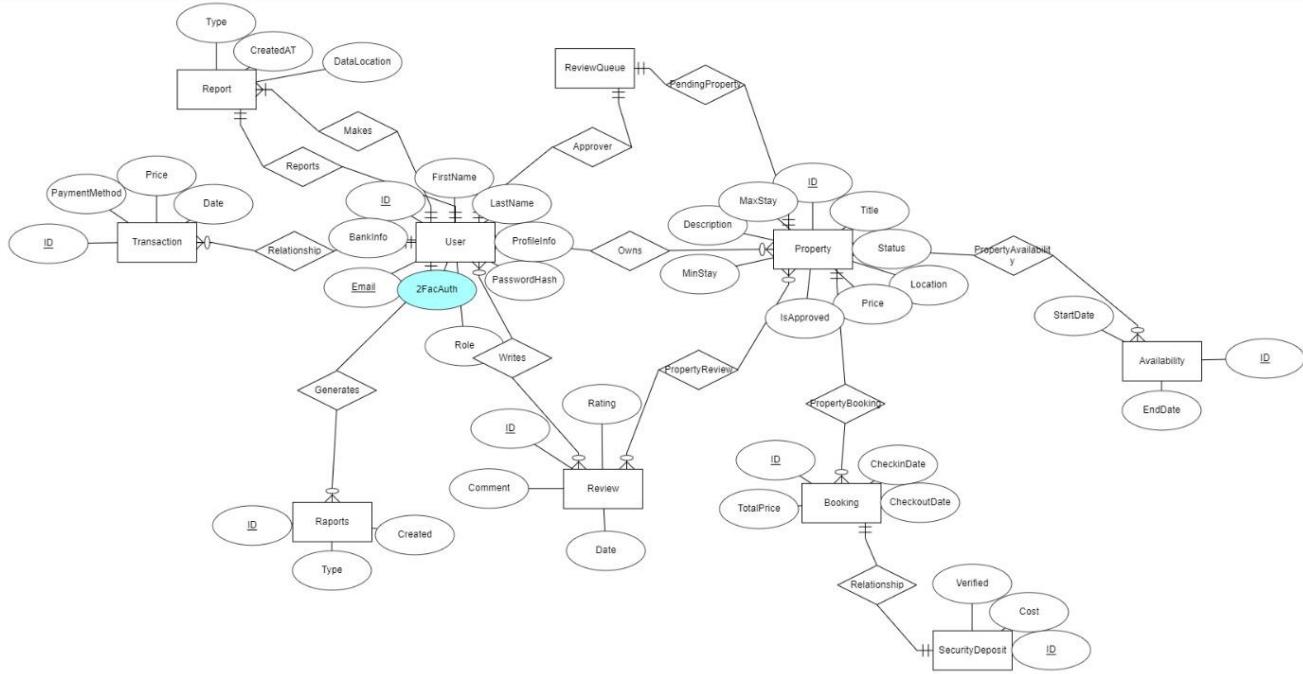
UC Name	SR-4: User Books Property
Summary	<i>The user is able to create a booking for a certain property for a required time</i>
Dependency	None
Actors	User
Preconditions	Mone
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• <i>The user selects a property</i></li> <li>• <i>User is able to view pictures, details, reviews</i></li> <li>• <i>User completes payment</i></li> <li>• <i>User is able to book property</i></li> </ul>
Description of the Alternative Sequence	N/A
Non functional requirements	None
Postconditions	<i>The system should verify that every transaction is secure and valid</i>

**Property Rentals Management System Requirements Specification**

UC Name	<b>SR5 - Leaving Reviews</b>
Summary	<i>User is able to leave review after a booking</i>
Dependency	<i>None</i>
Actors	<i>User</i>
Preconditions	<i>None</i>
Description of the Main Sequence	<ul style="list-style-type: none"> <li>• User finishes their booking</li> <li>• User is asked to leave a review</li> <li>• User leaves review under booking</li> </ul>
Description of the Alternative Sequence	<i>N/A</i>
Non functional requirements	<i>None</i>
Postconditions	<i>Review is checked to make sure it is valid</i>

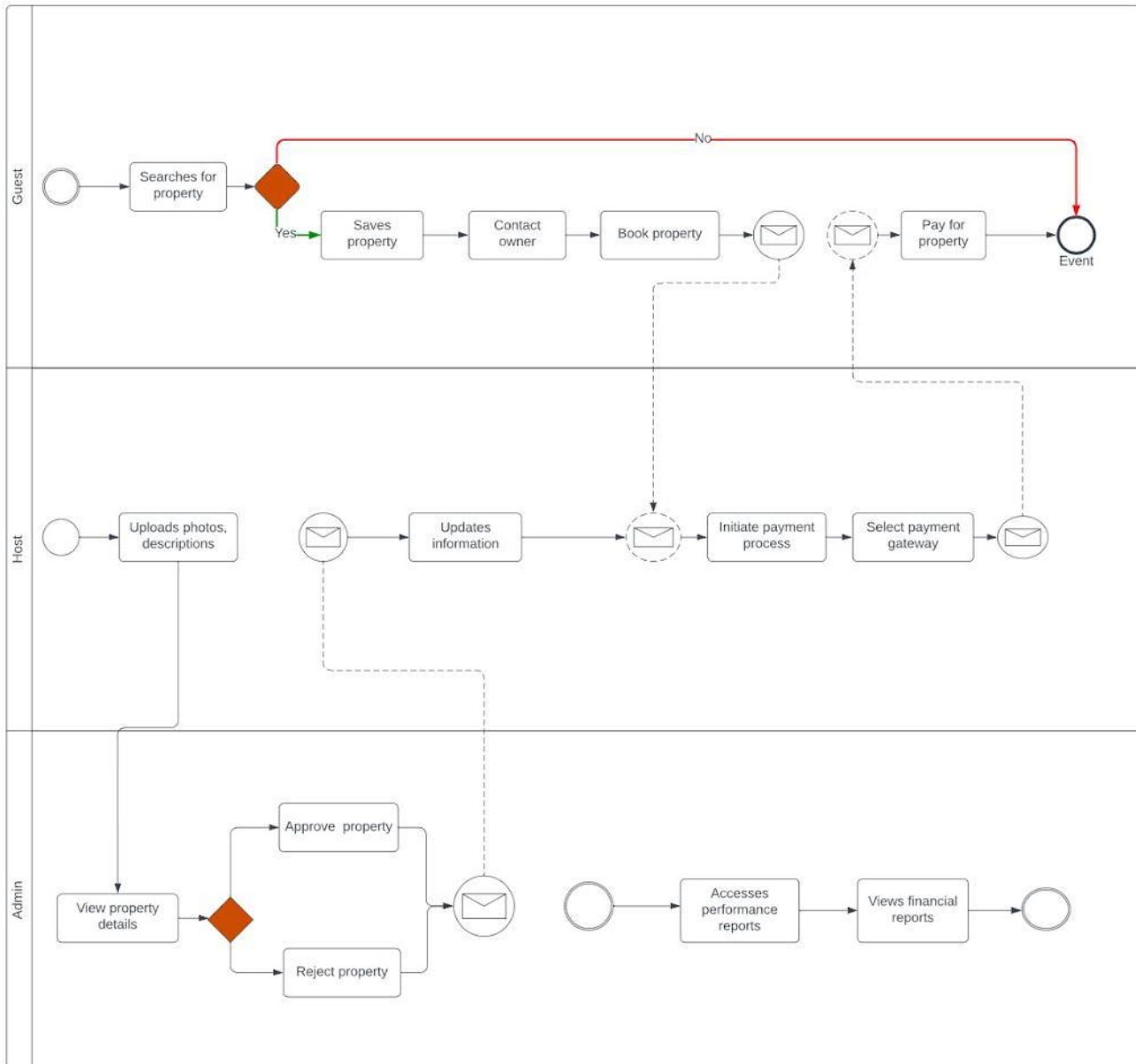
## 5. DIAGRAMS

### Entity Relationship Diagram (ERD)

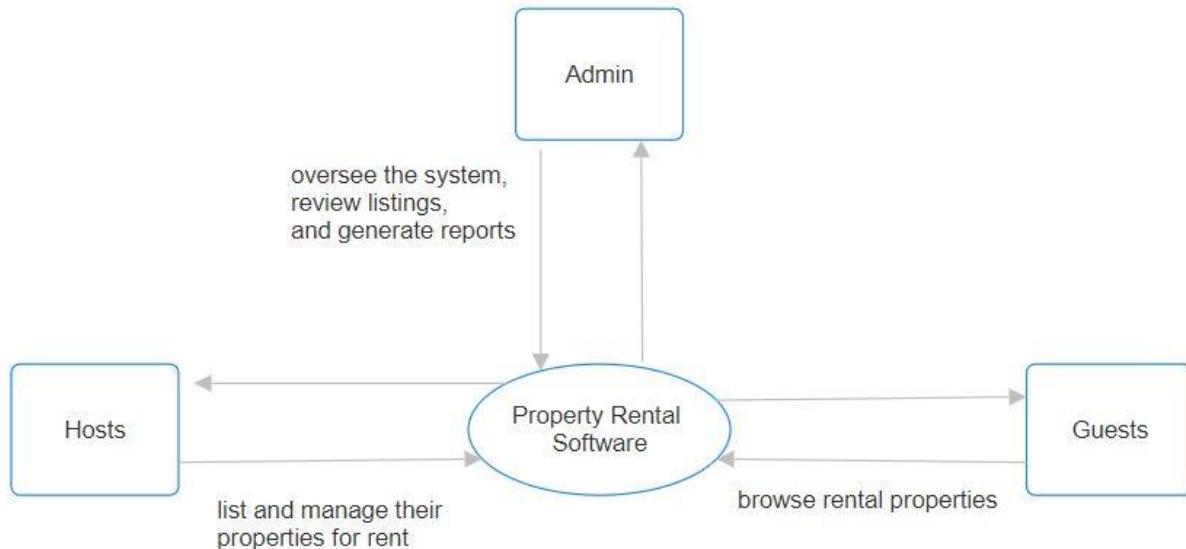


# Property Rentals Management System Requirements Specification

## BPMM

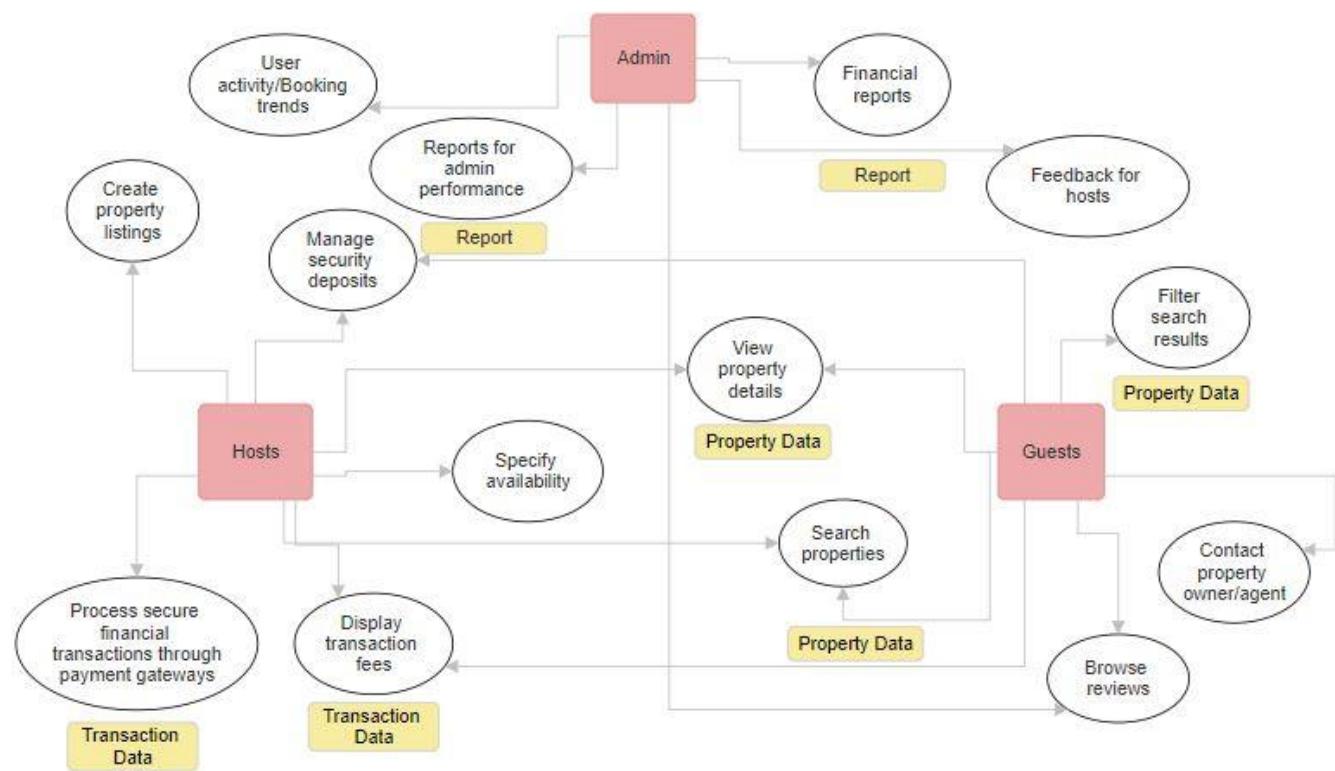


## DFD Level 1



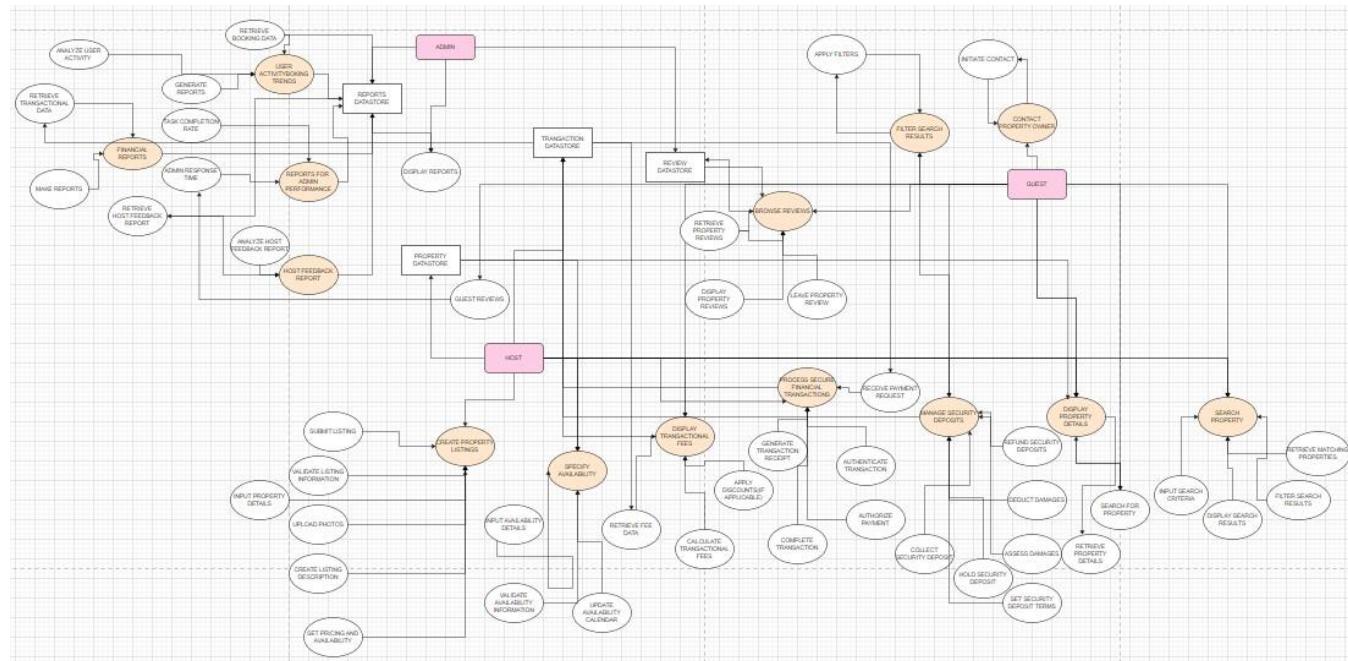
# Property Rentals Management System Requirements Specification

## DFD Level 2



# Property Rentals Management System Requirements Specification

## DFD Level 3

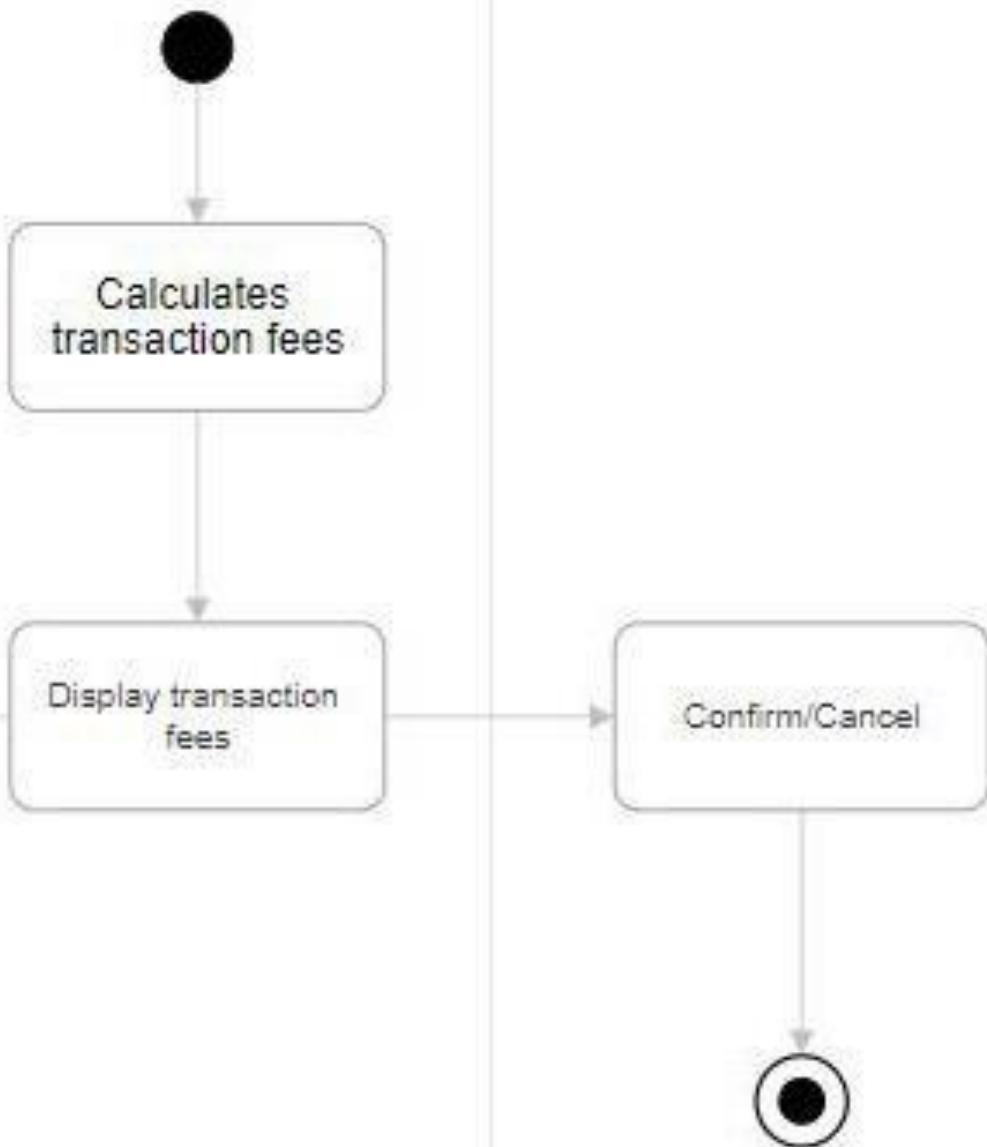


## ACTIVITY DIAGRAMS

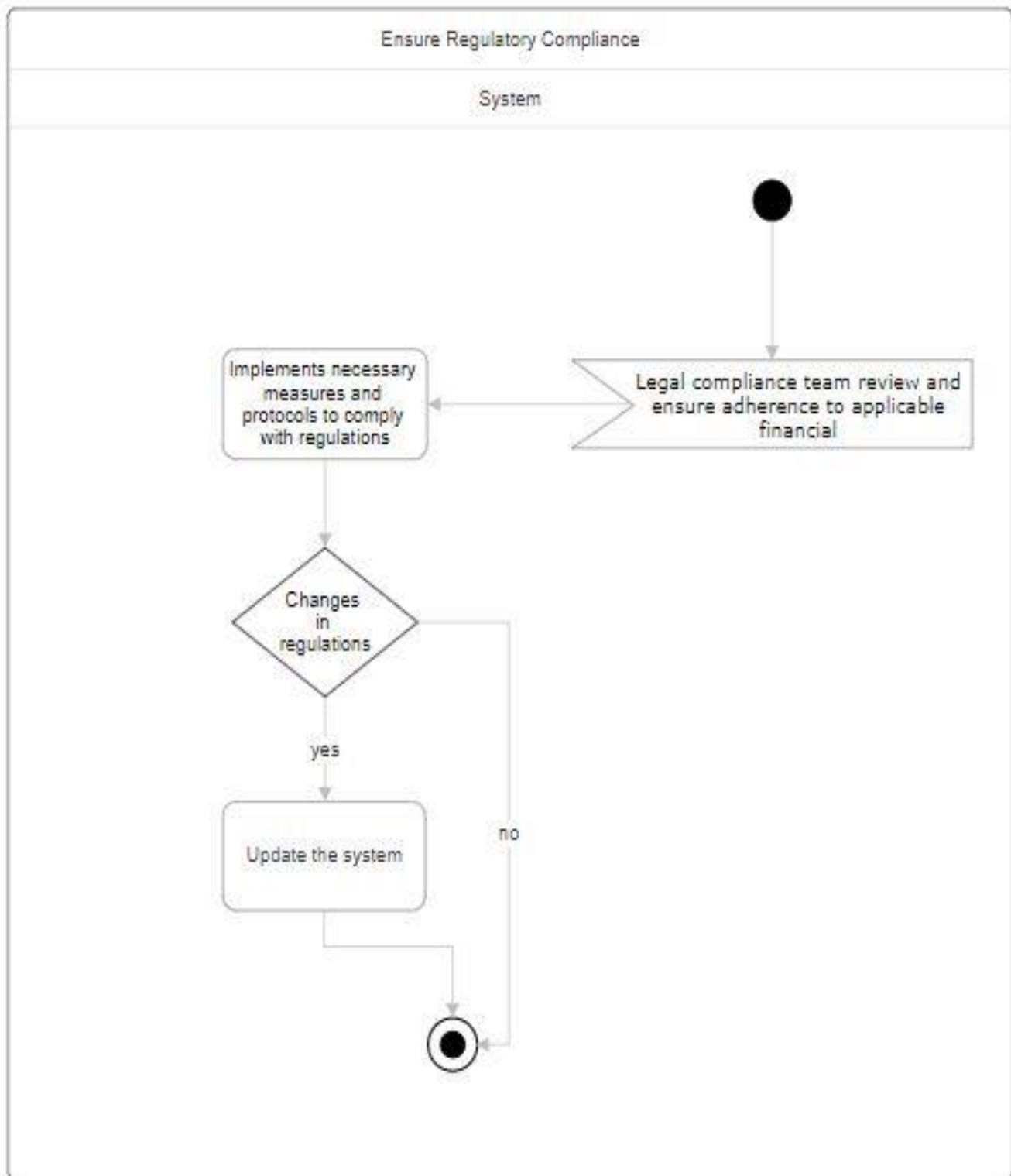
### Display Transaction Fees

System

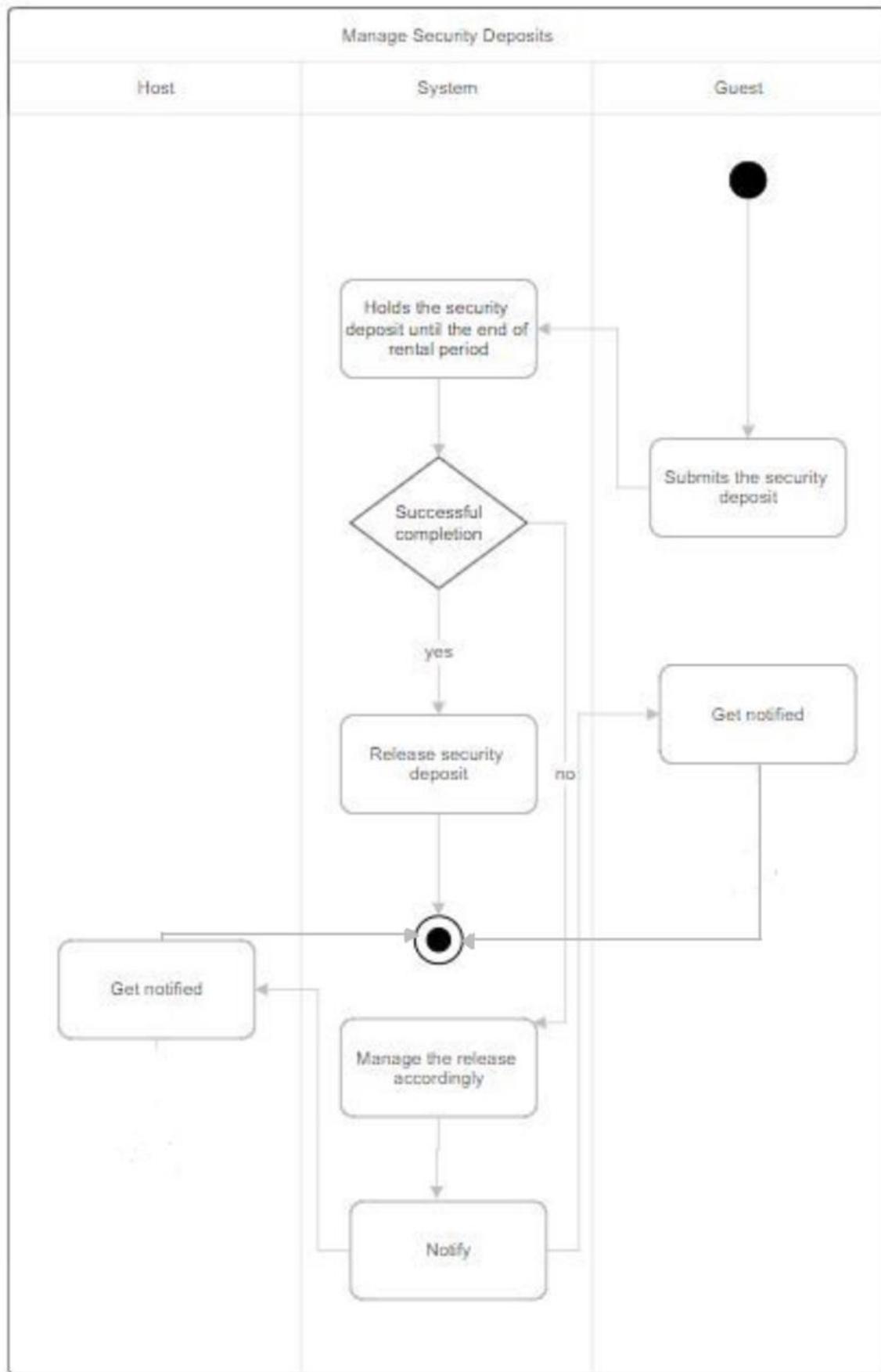
Guest



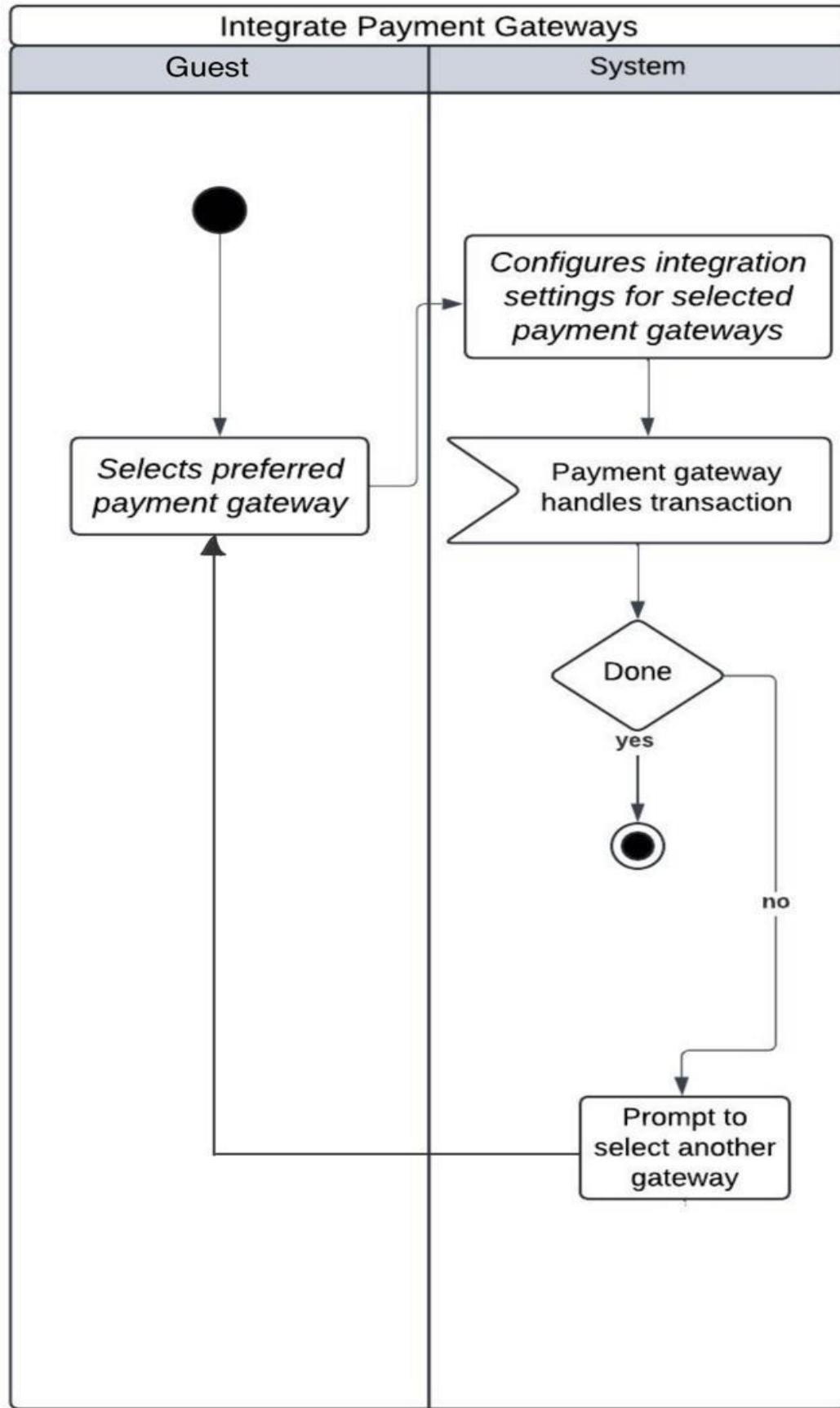
## Property Rentals Management System Requirements Specification



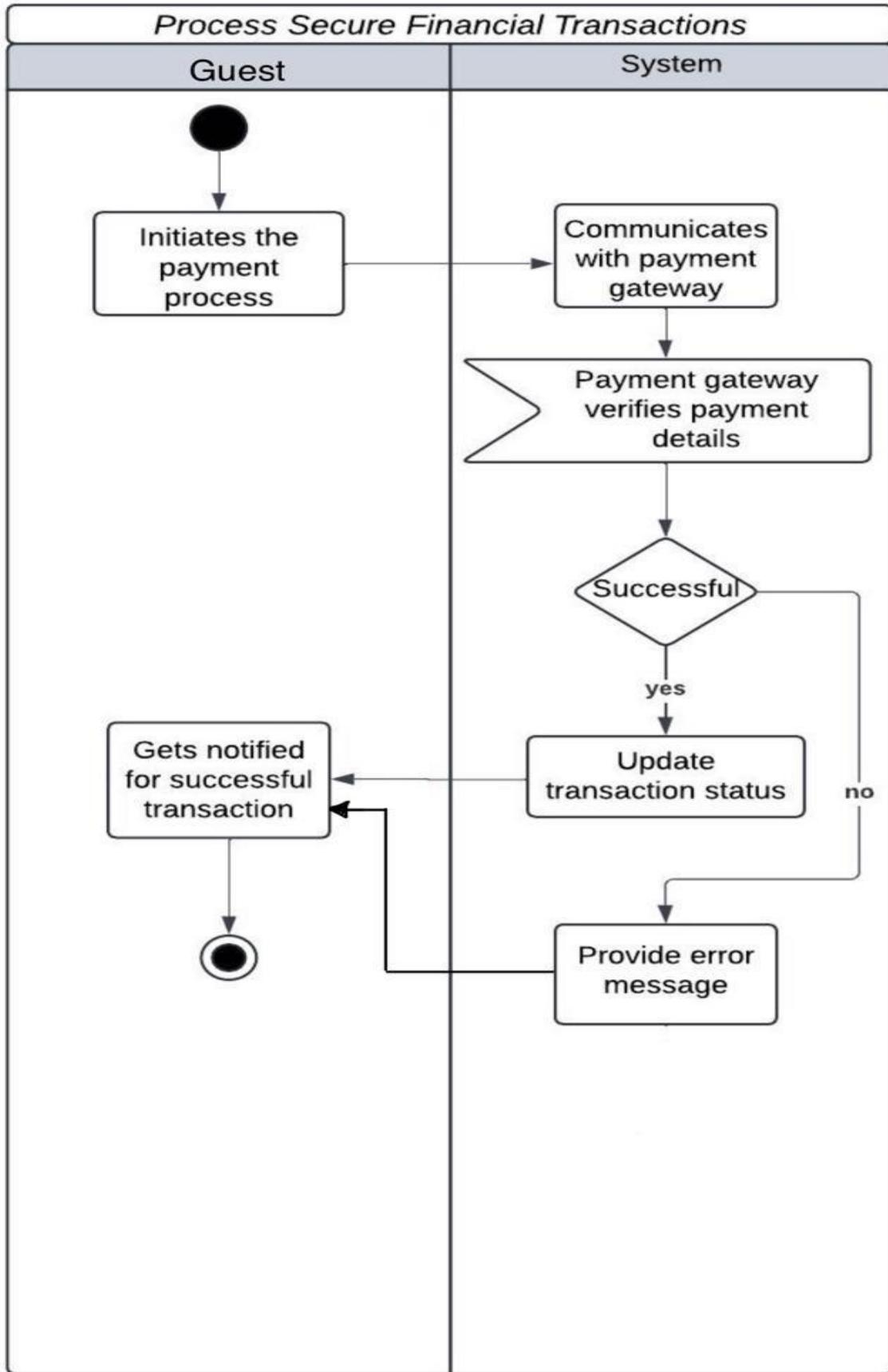
## Property Rentals Management System Requirements Specification



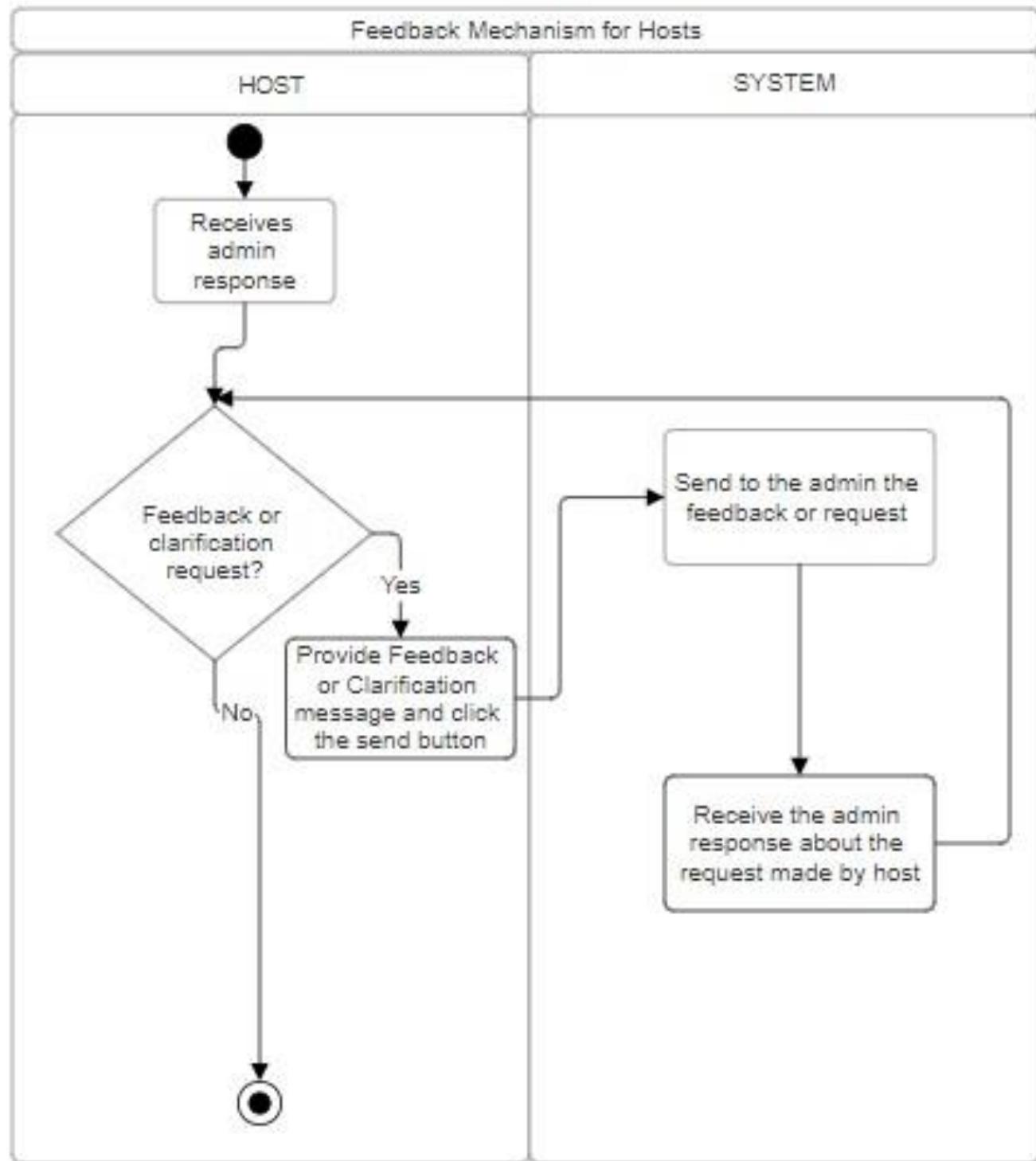
## Property Rentals Management System Requirements Specification



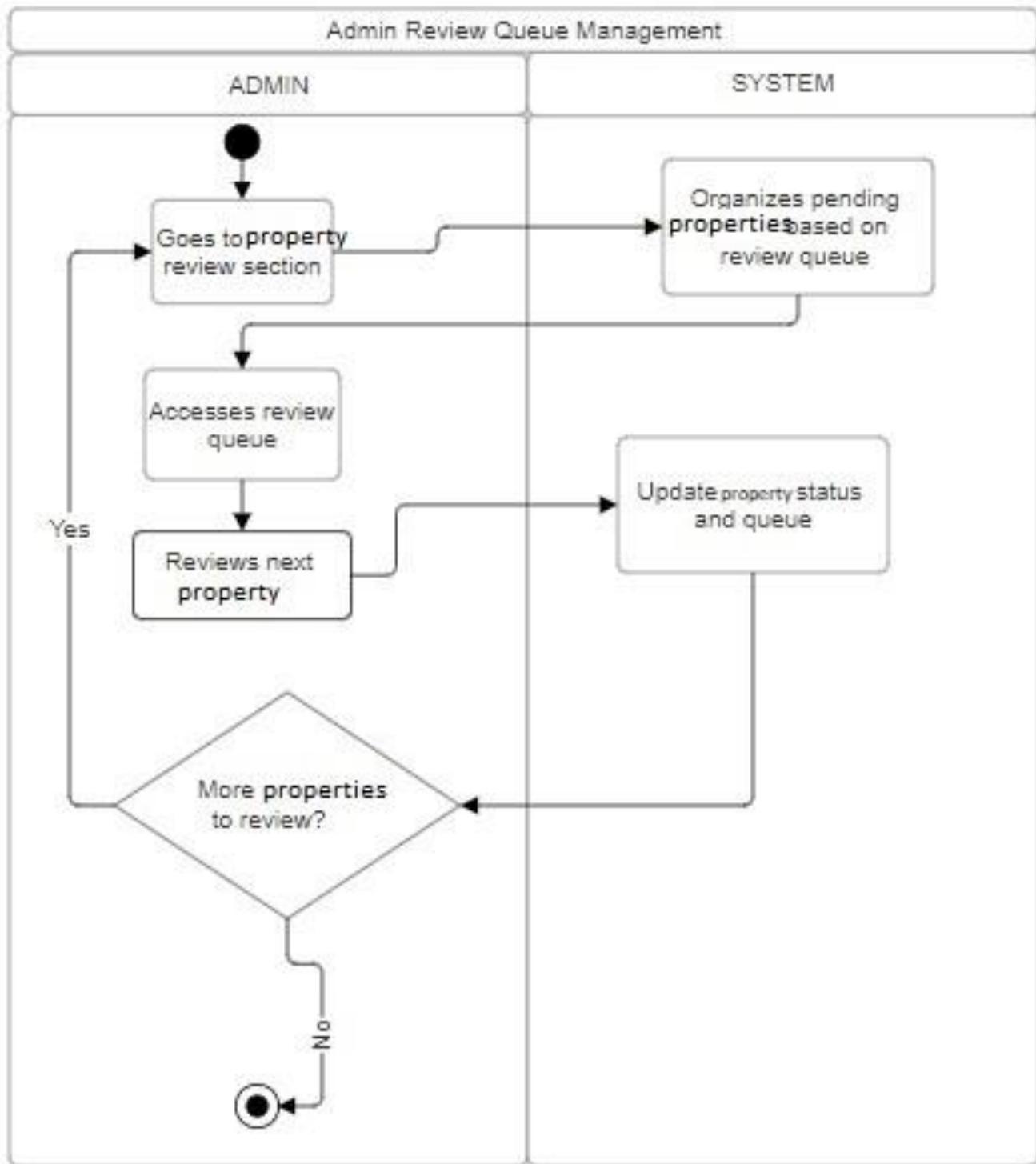
## Property Rentals Management System Requirements Specification



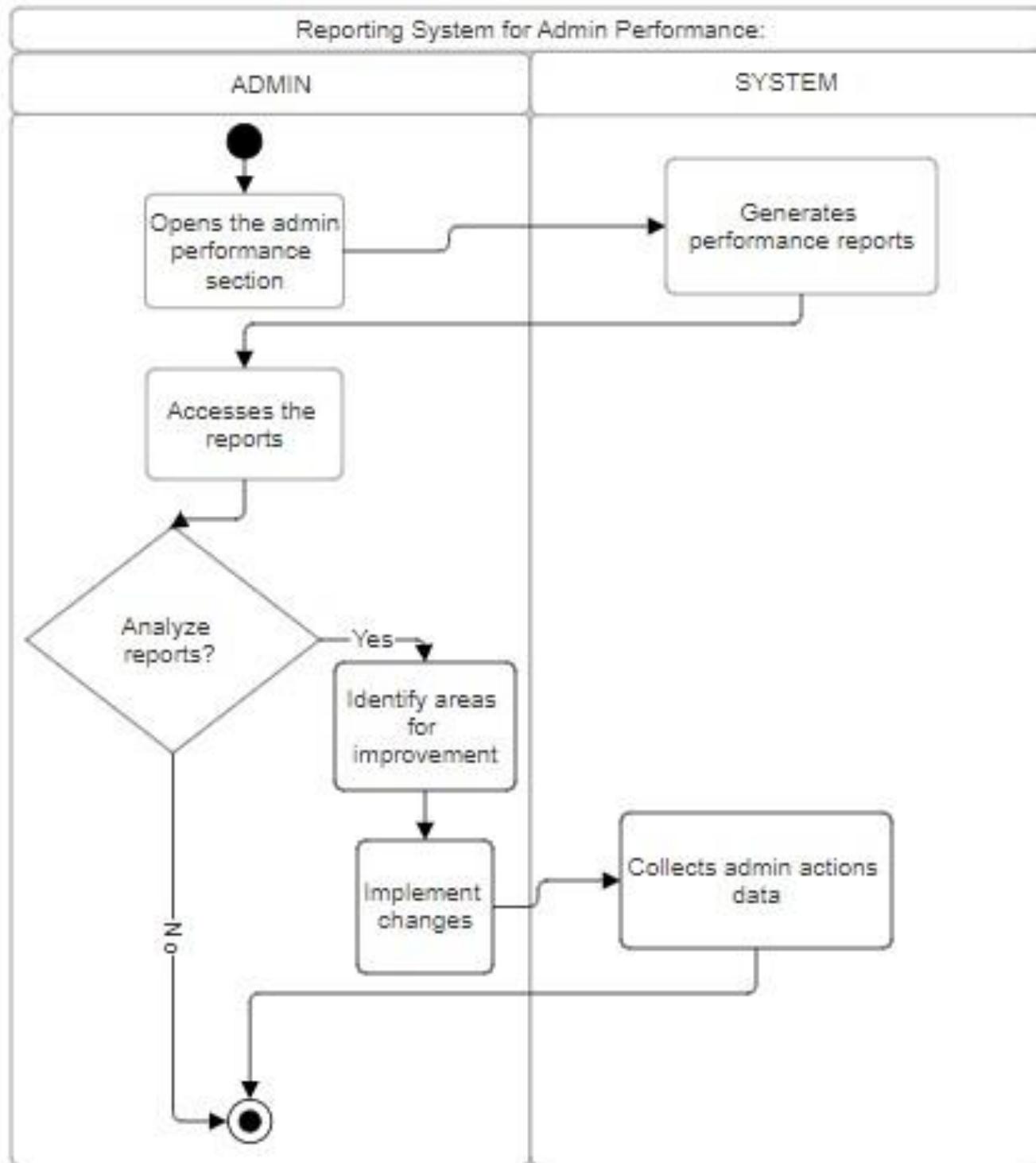
## Property Rentals Management System Requirements Specification



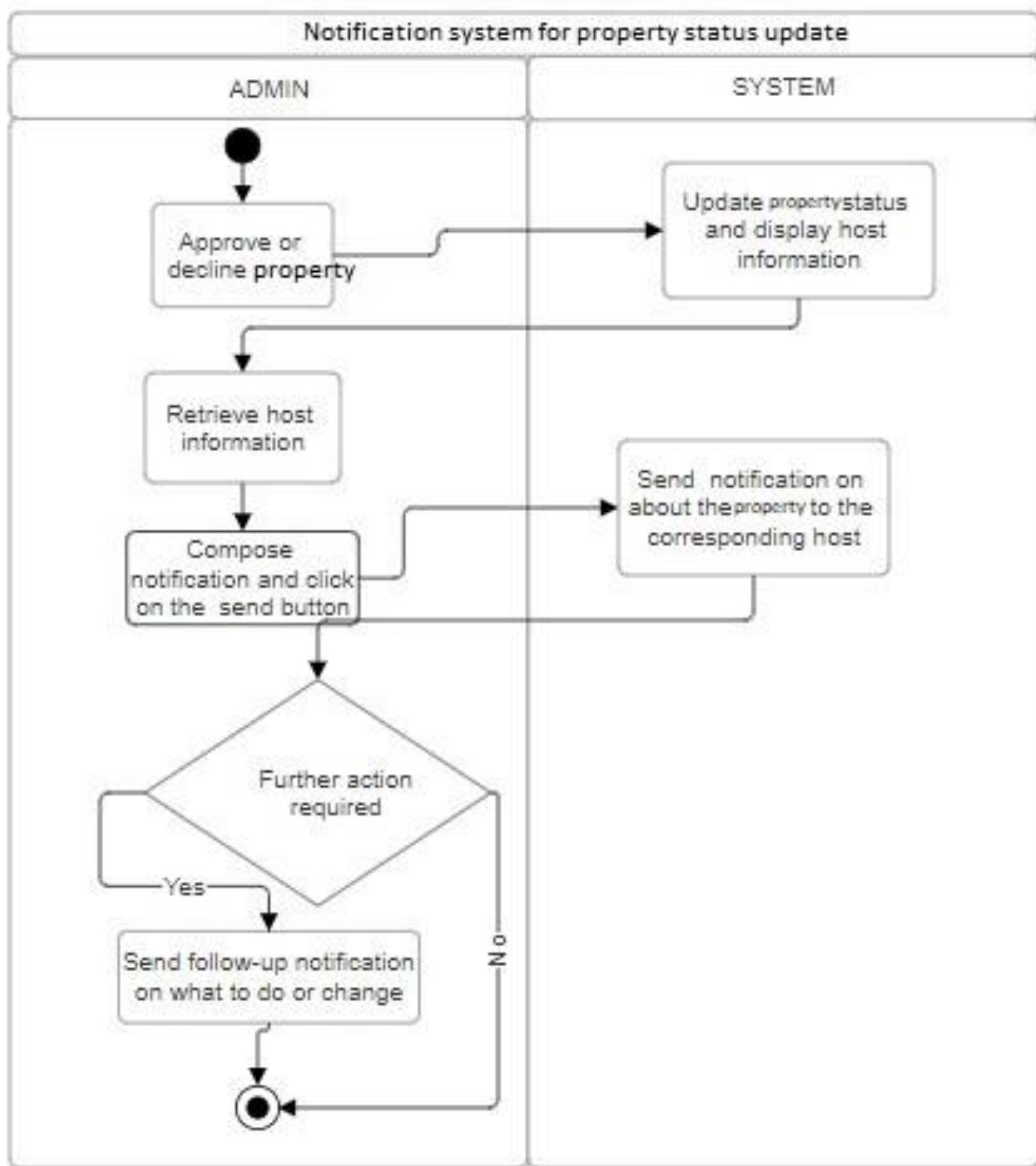
## Property Rentals Management System Requirements Specification

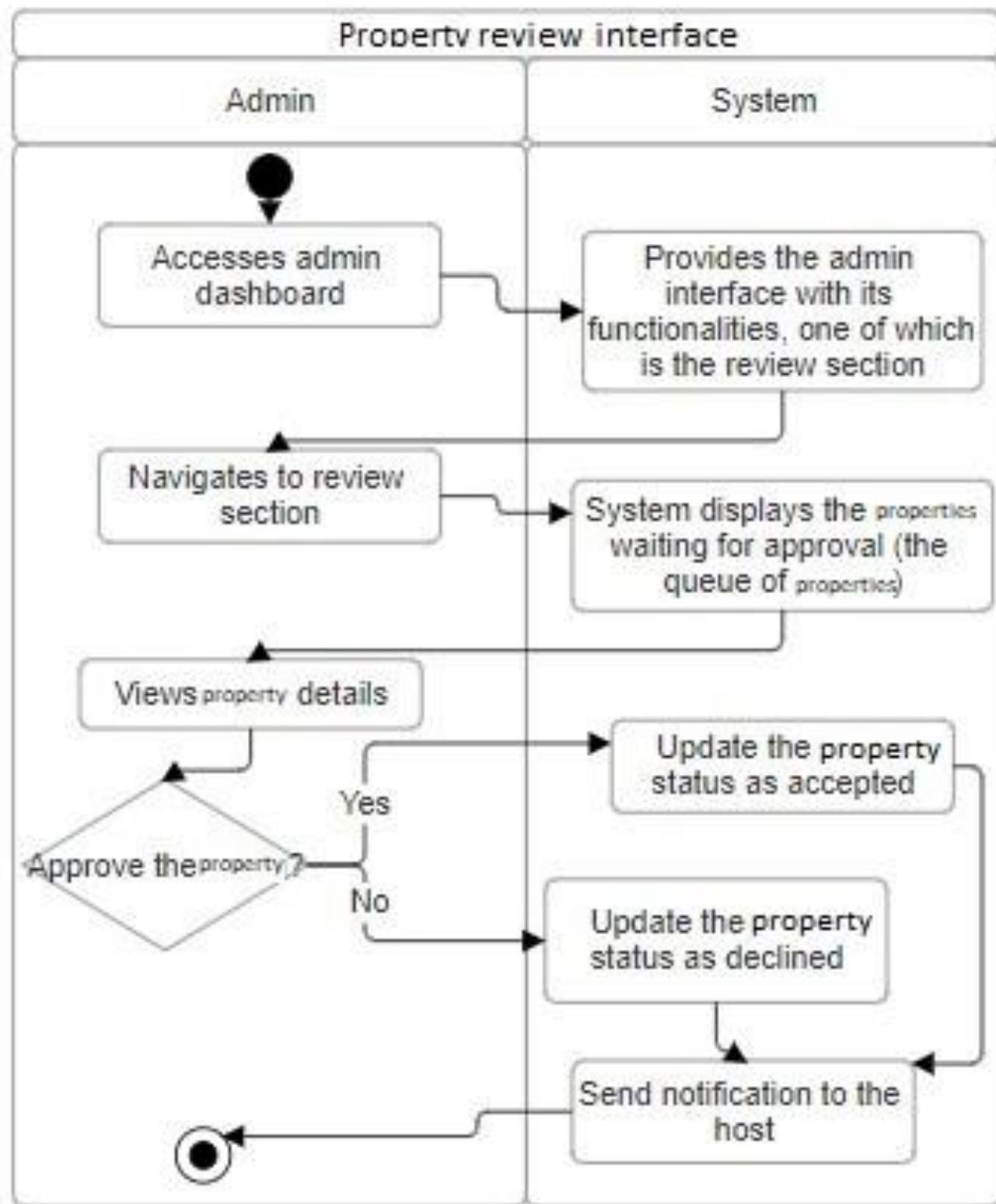


## Property Rentals Management System Requirements Specification

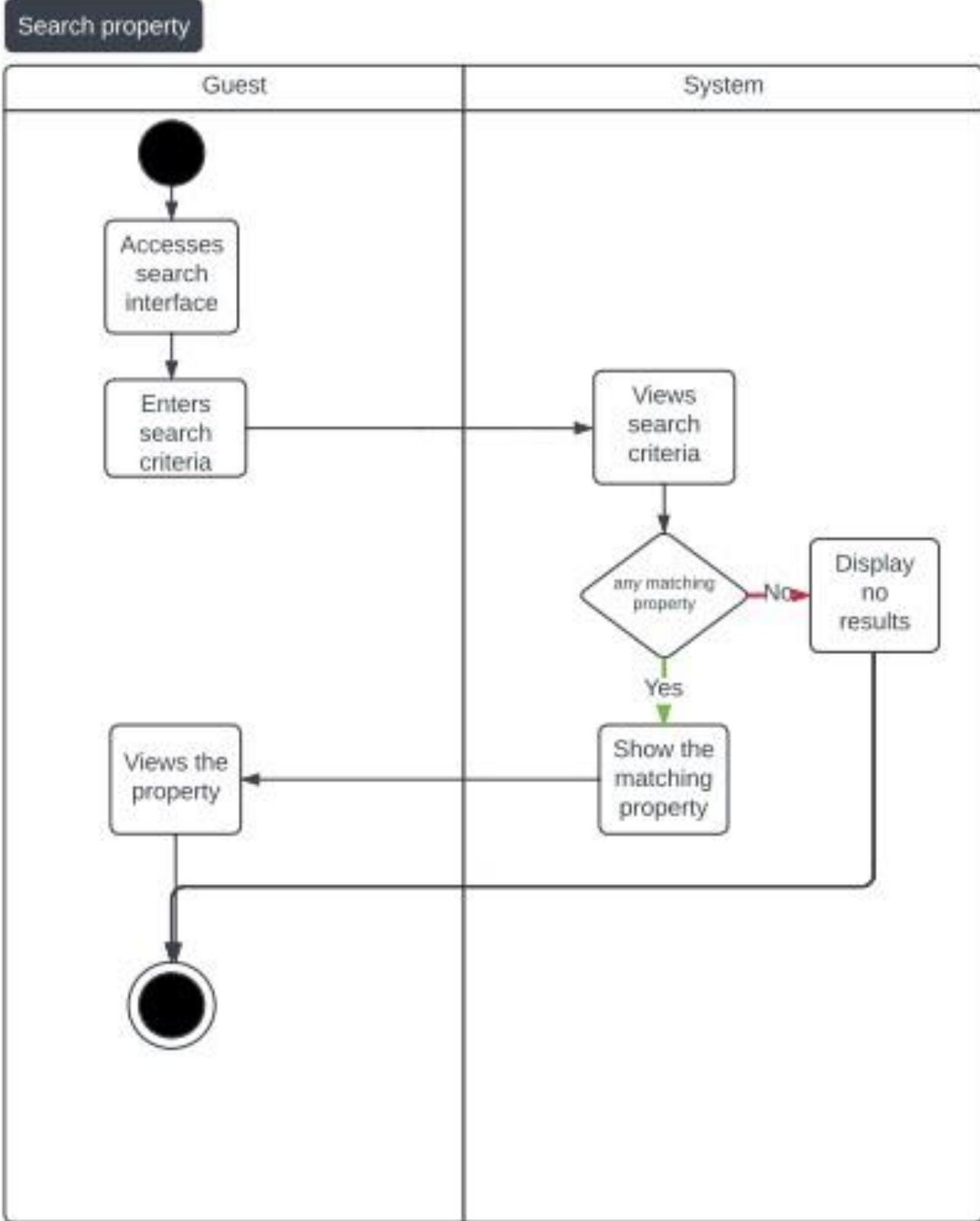


## Property Rentals Management System Requirements Specification



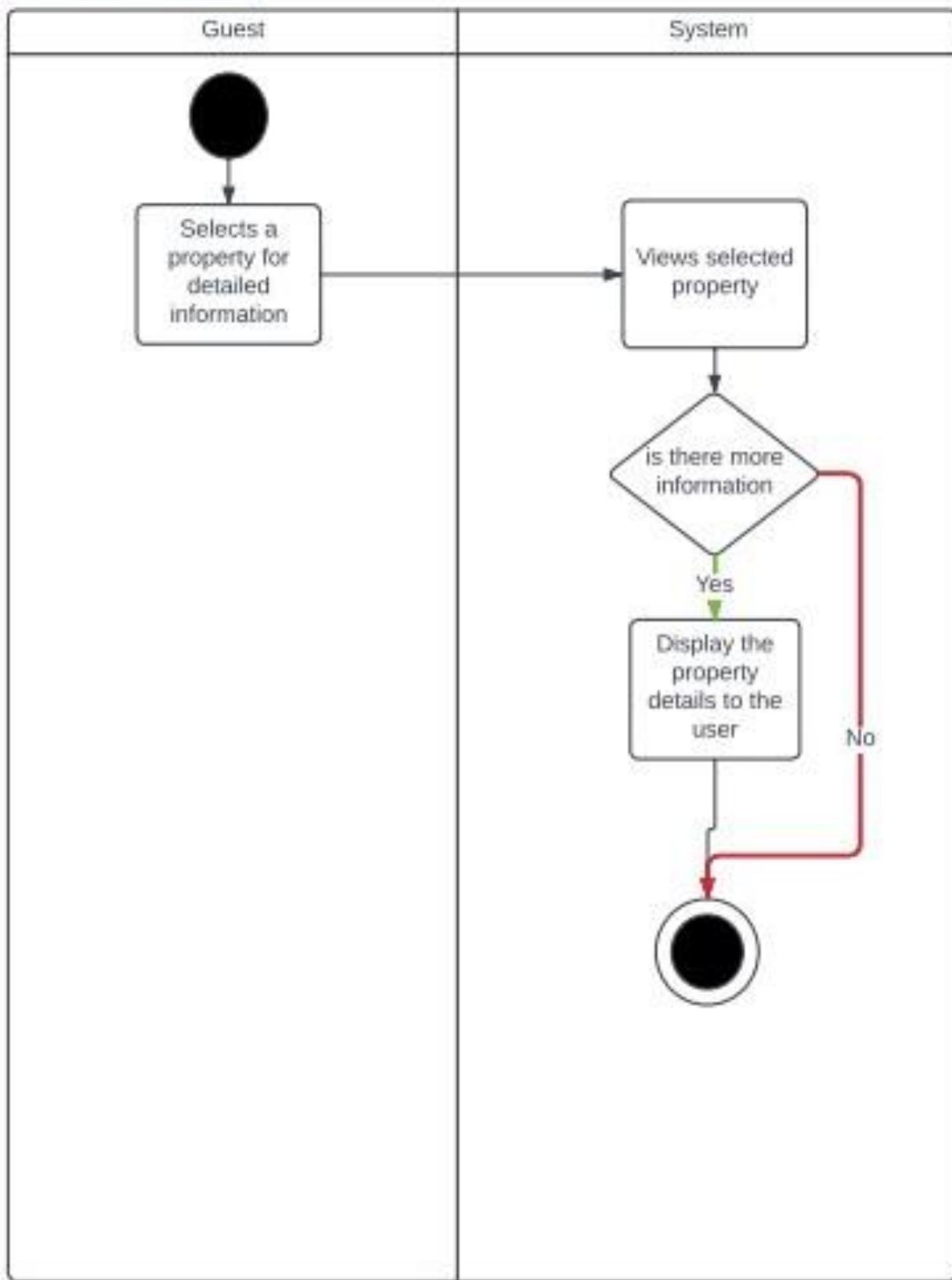


## Property Rentals Management System Requirements Specification



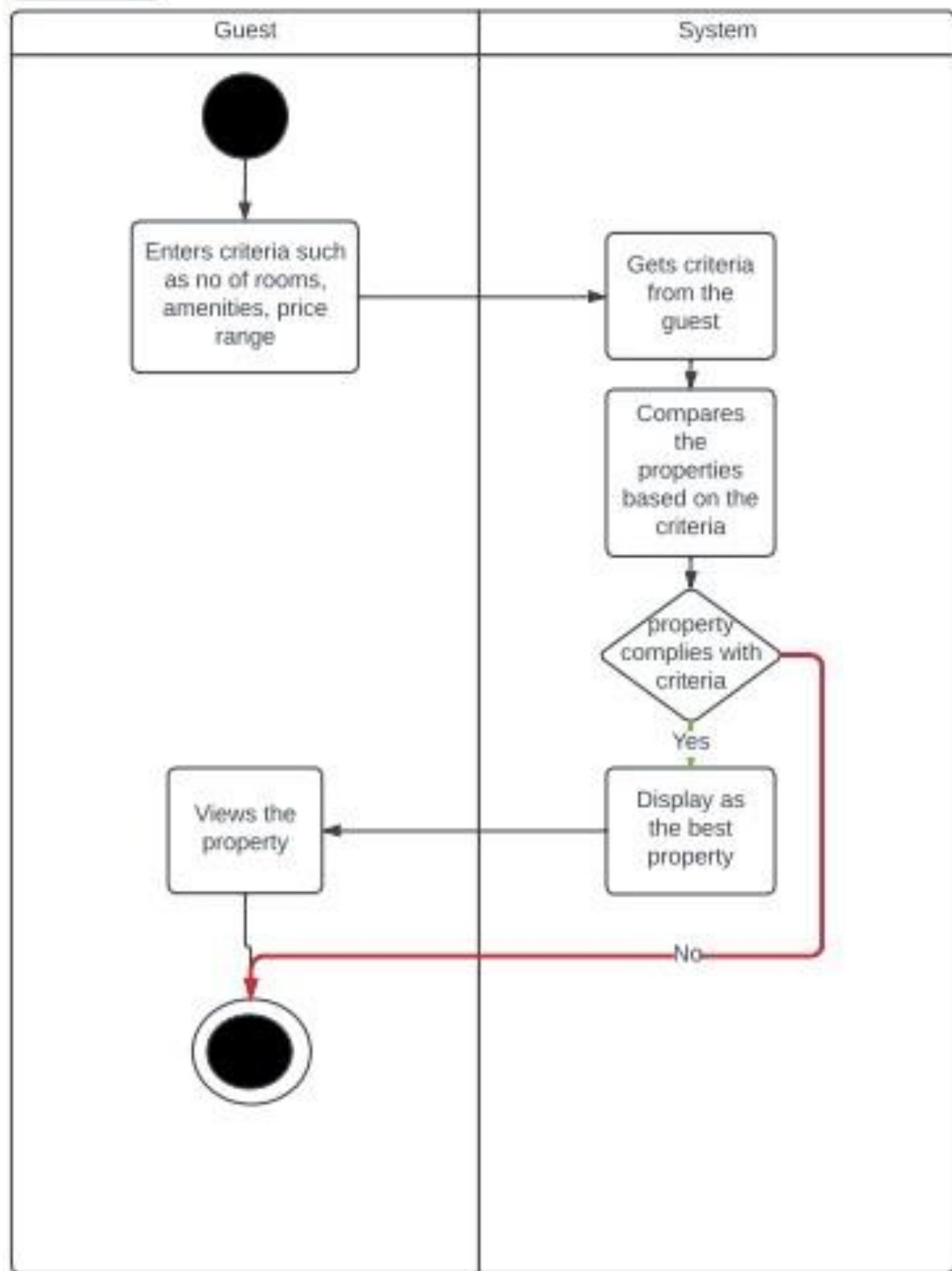
## Property Rentals Management System Requirements Specification

### View Property Details



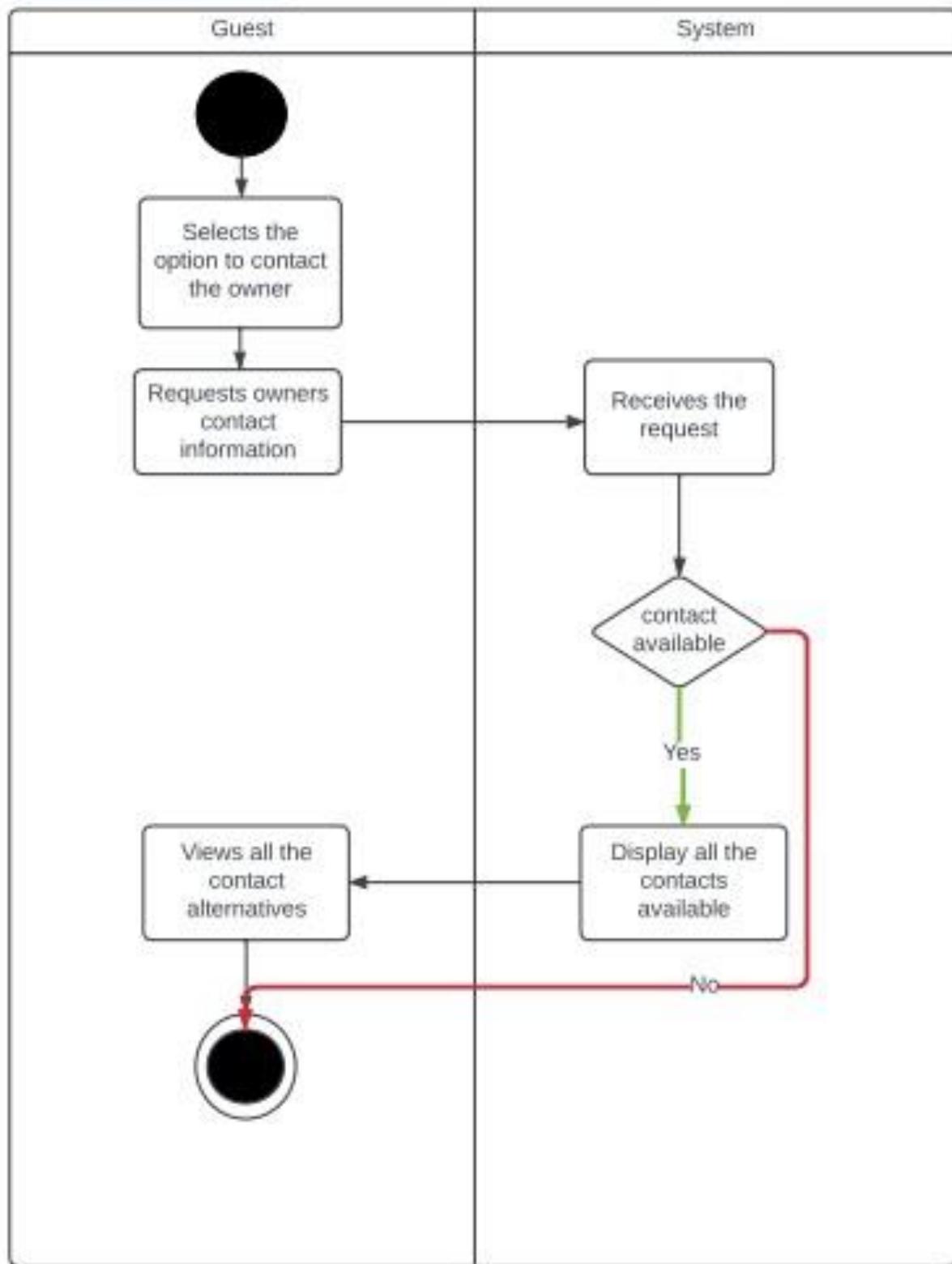
## Property Rentals Management System Requirements Specification

### Comparison

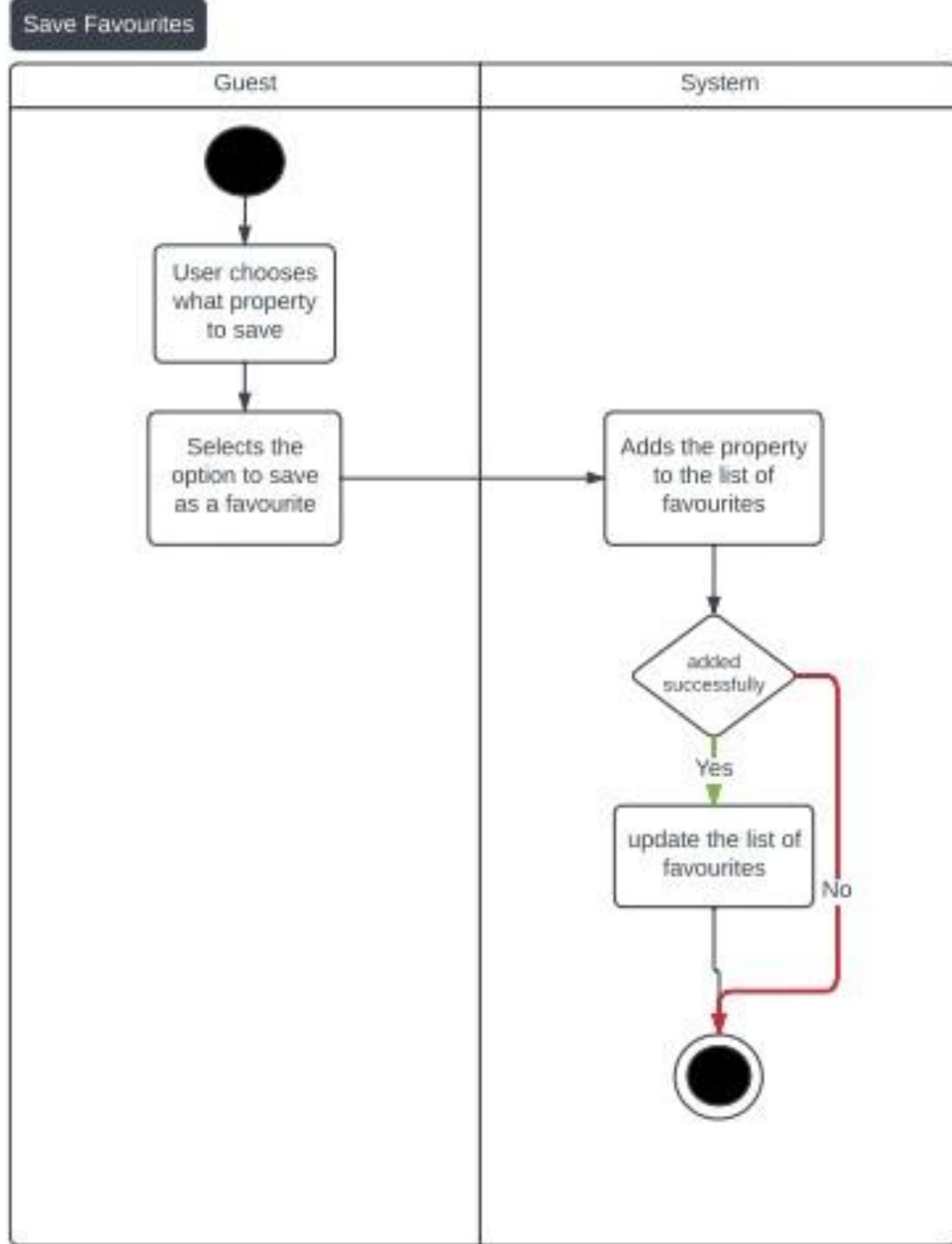


## Property Rentals Management System Requirements Specification

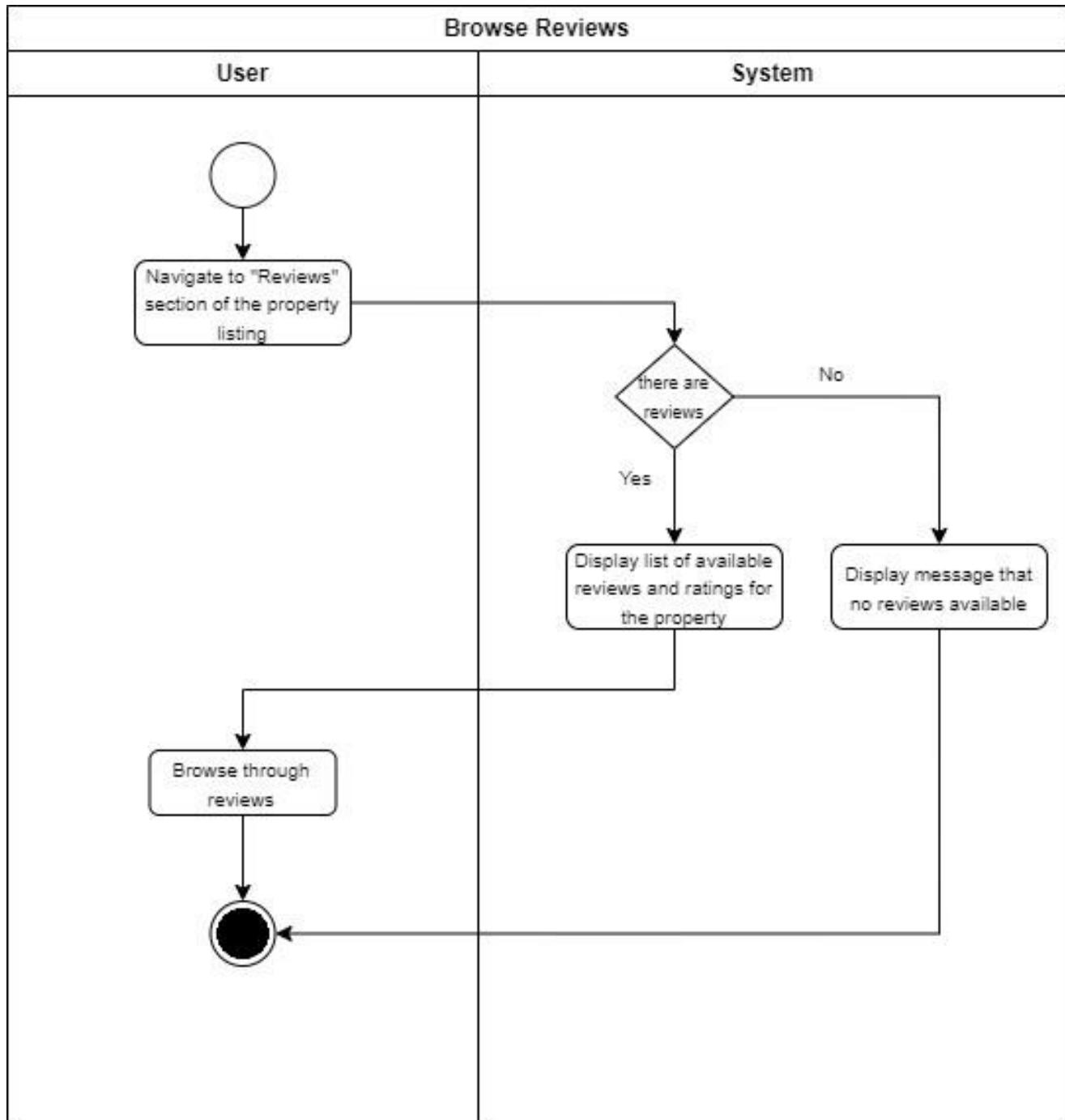
### Contact Property Owner



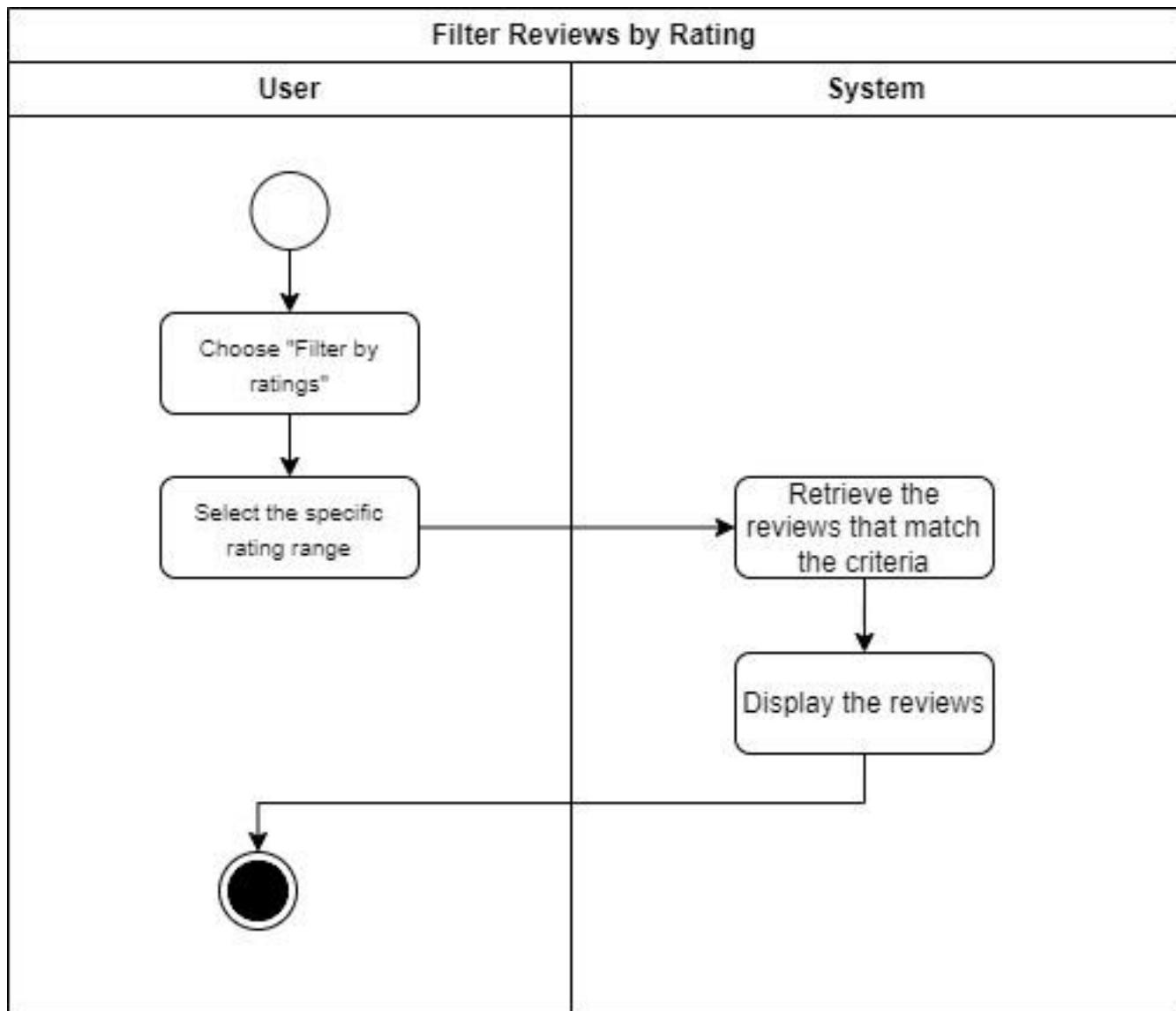
## Property Rentals Management System Requirements Specification



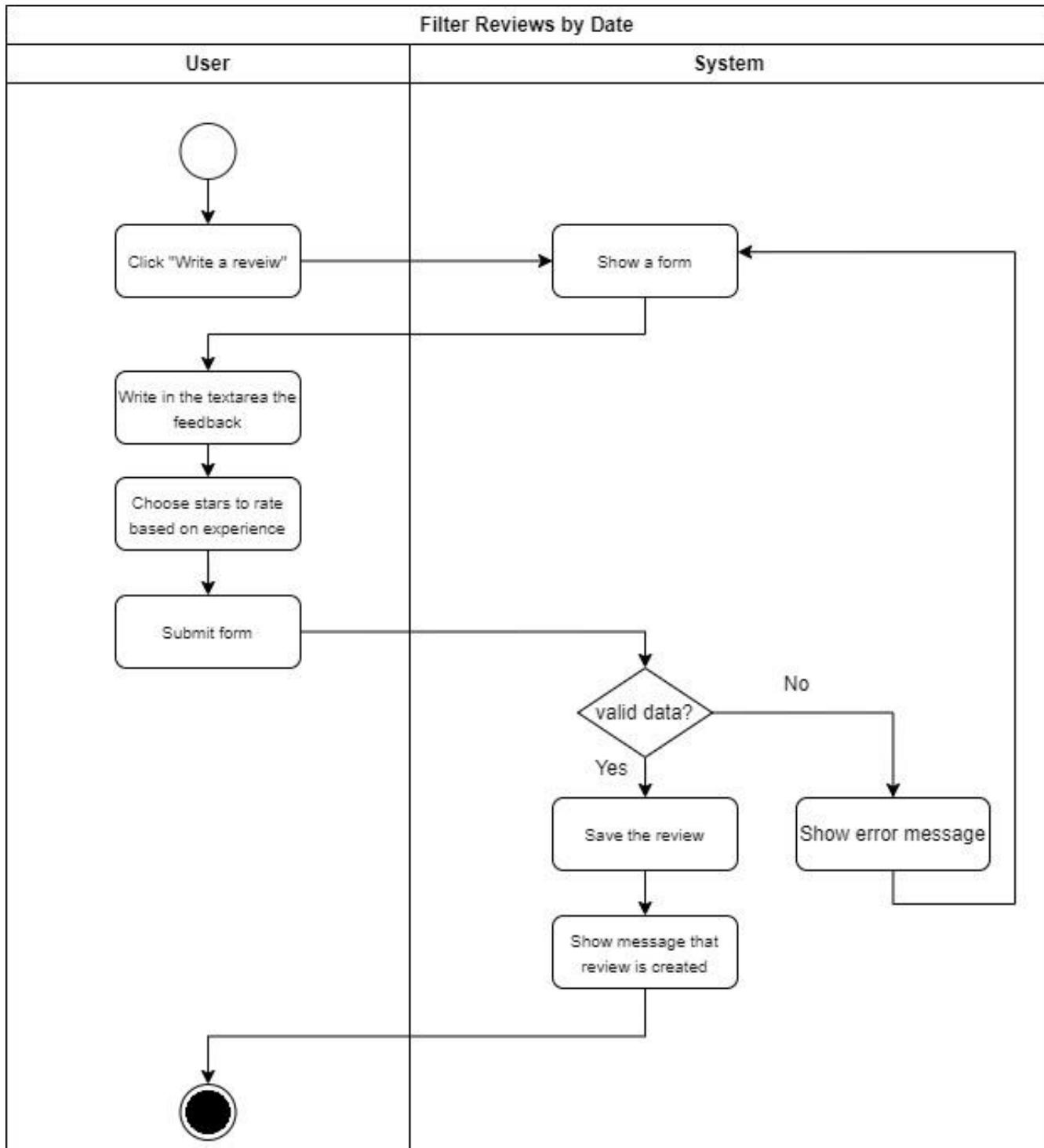
## Property Rentals Management System Requirements Specification



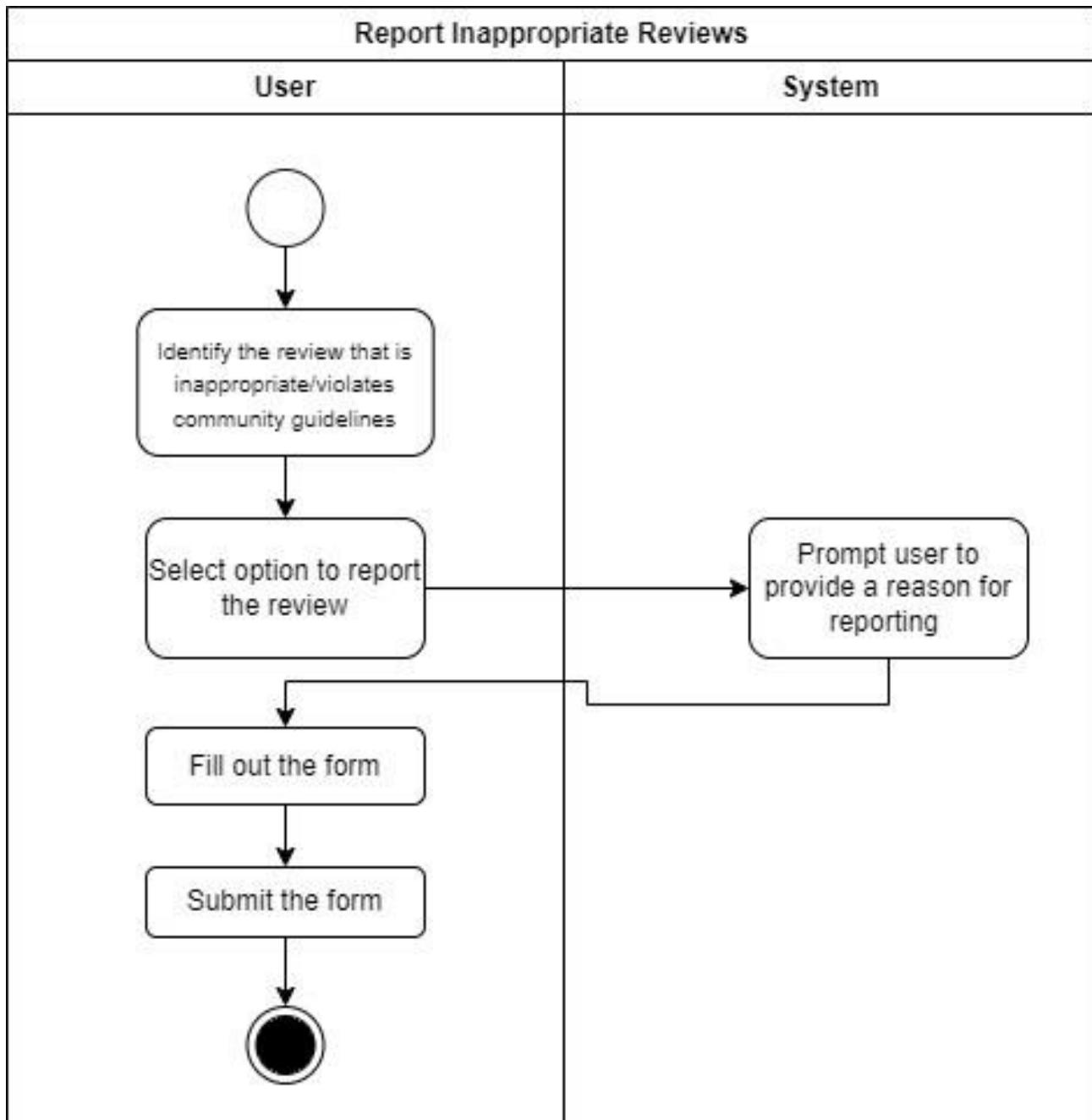
## Property Rentals Management System Requirements Specification



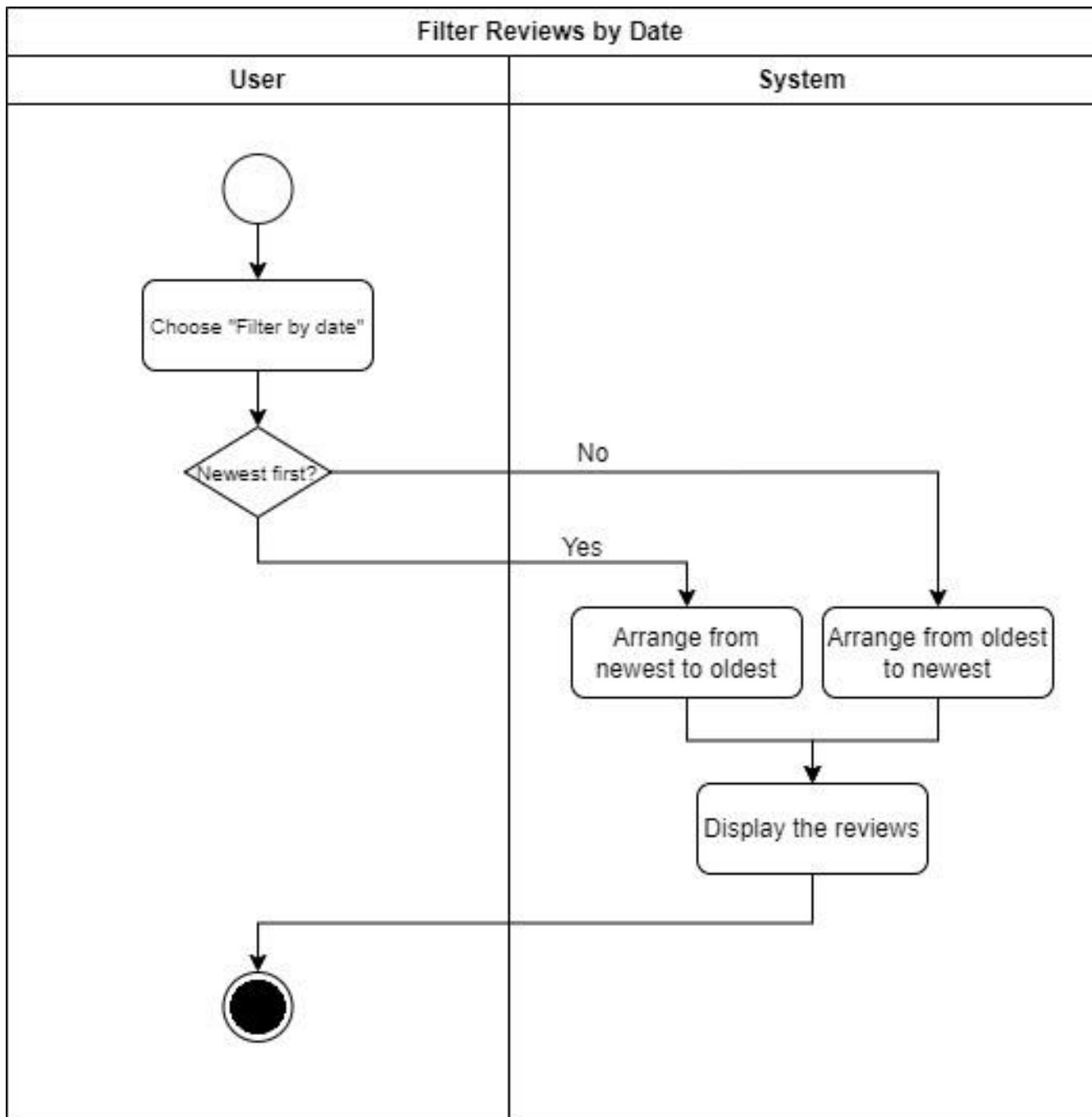
## Property Rentals Management System Requirements Specification



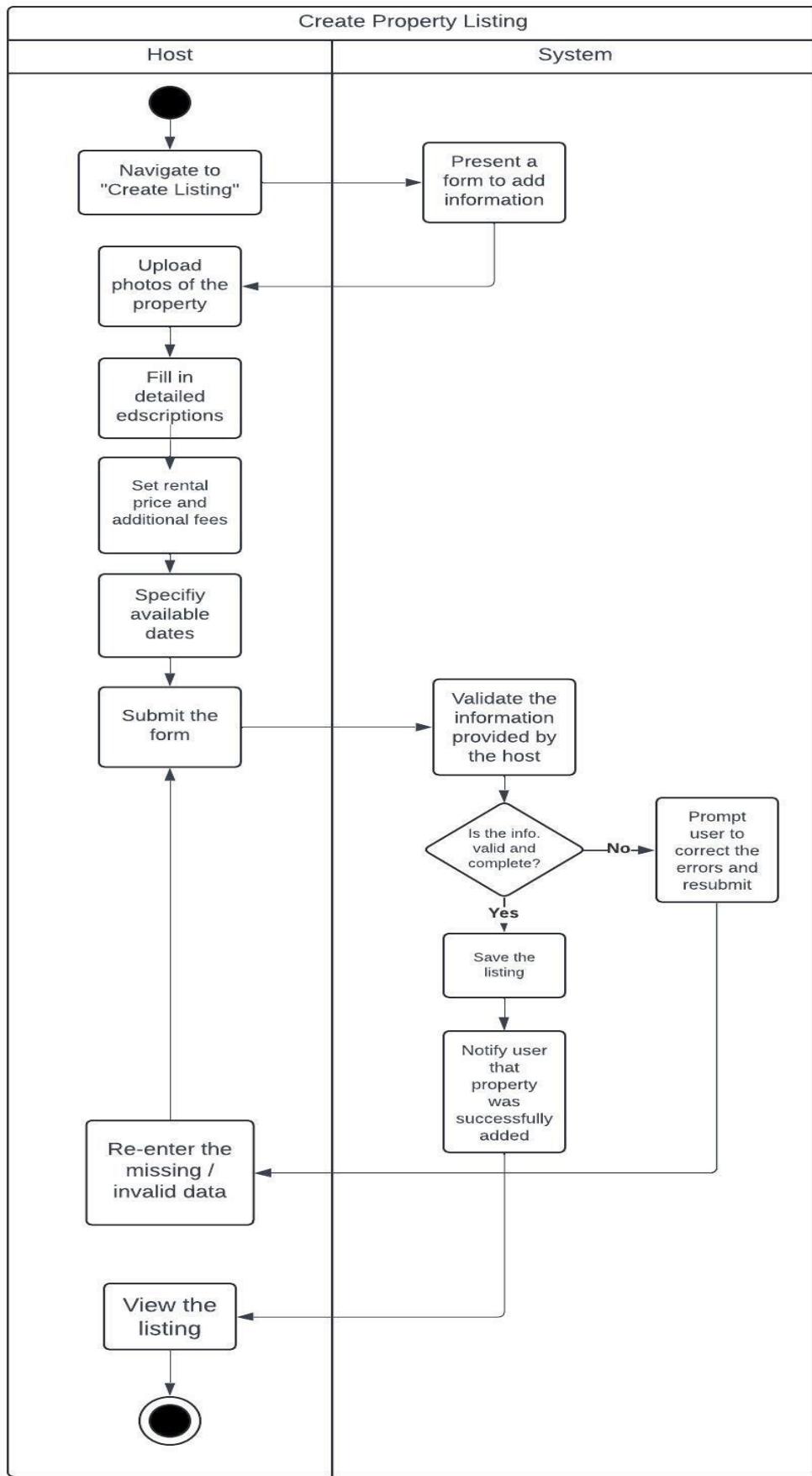
## Property Rentals Management System Requirements Specification



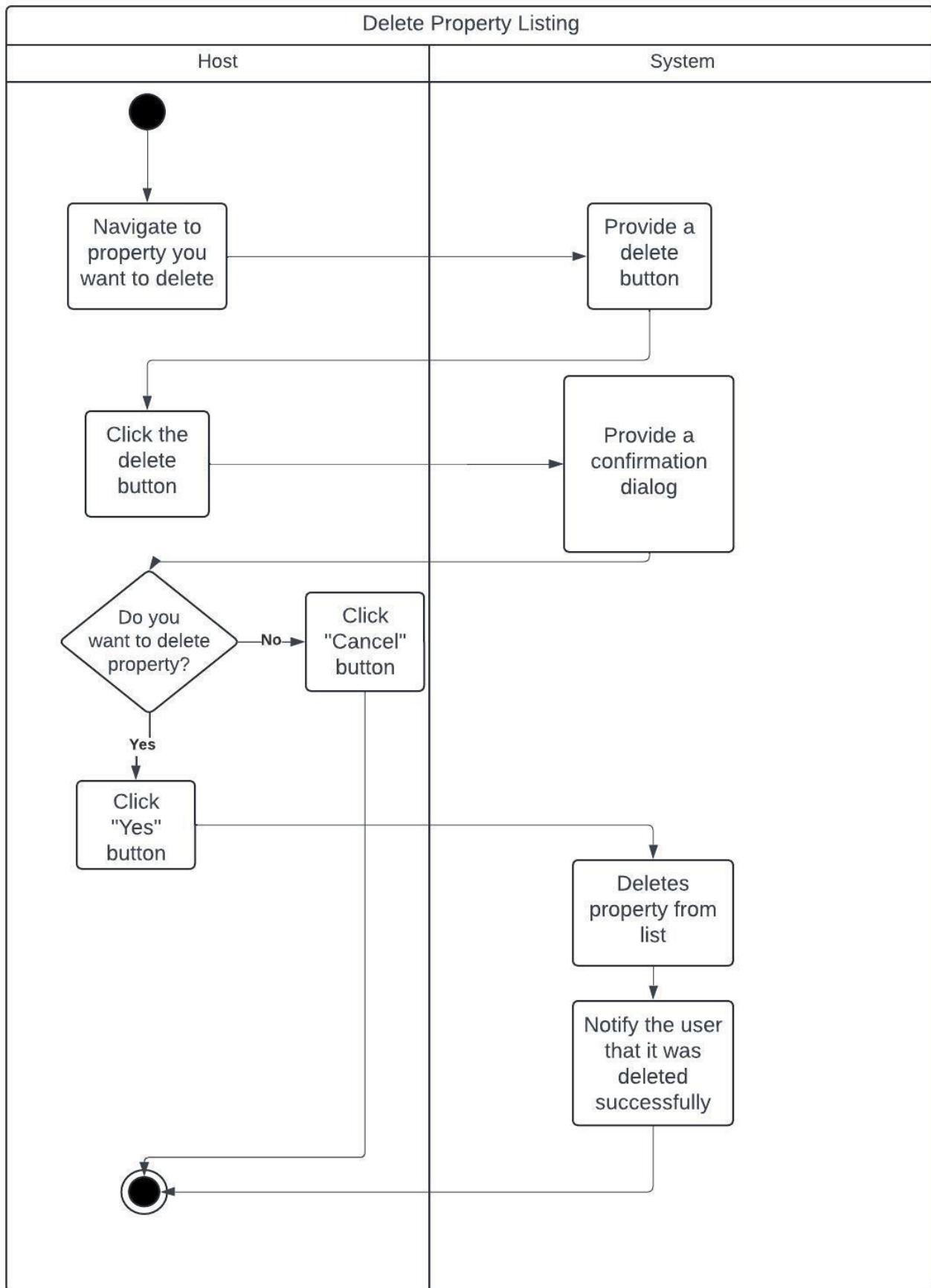
## Property Rentals Management System Requirements Specification



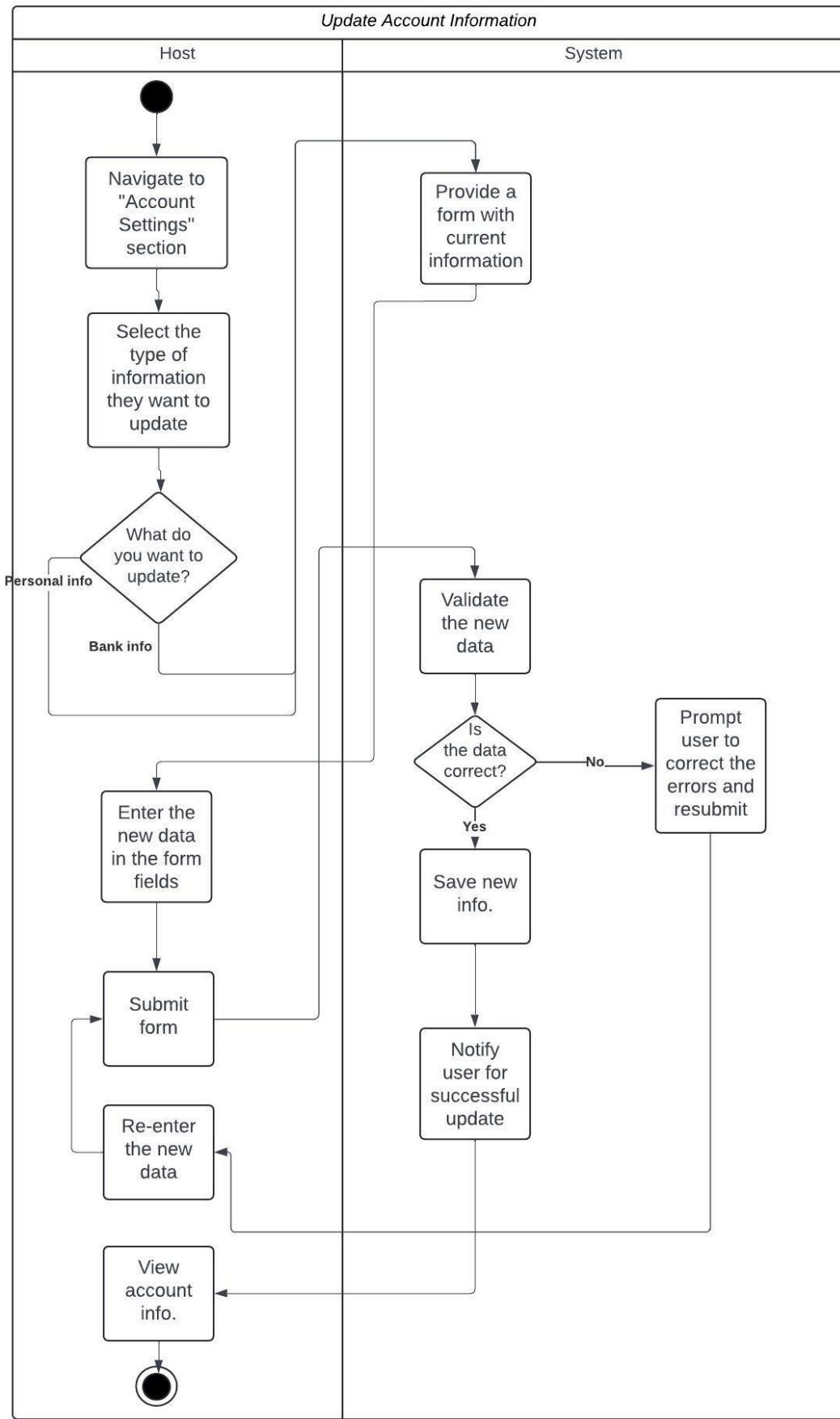
## Property Rentals Management System Requirements Specification



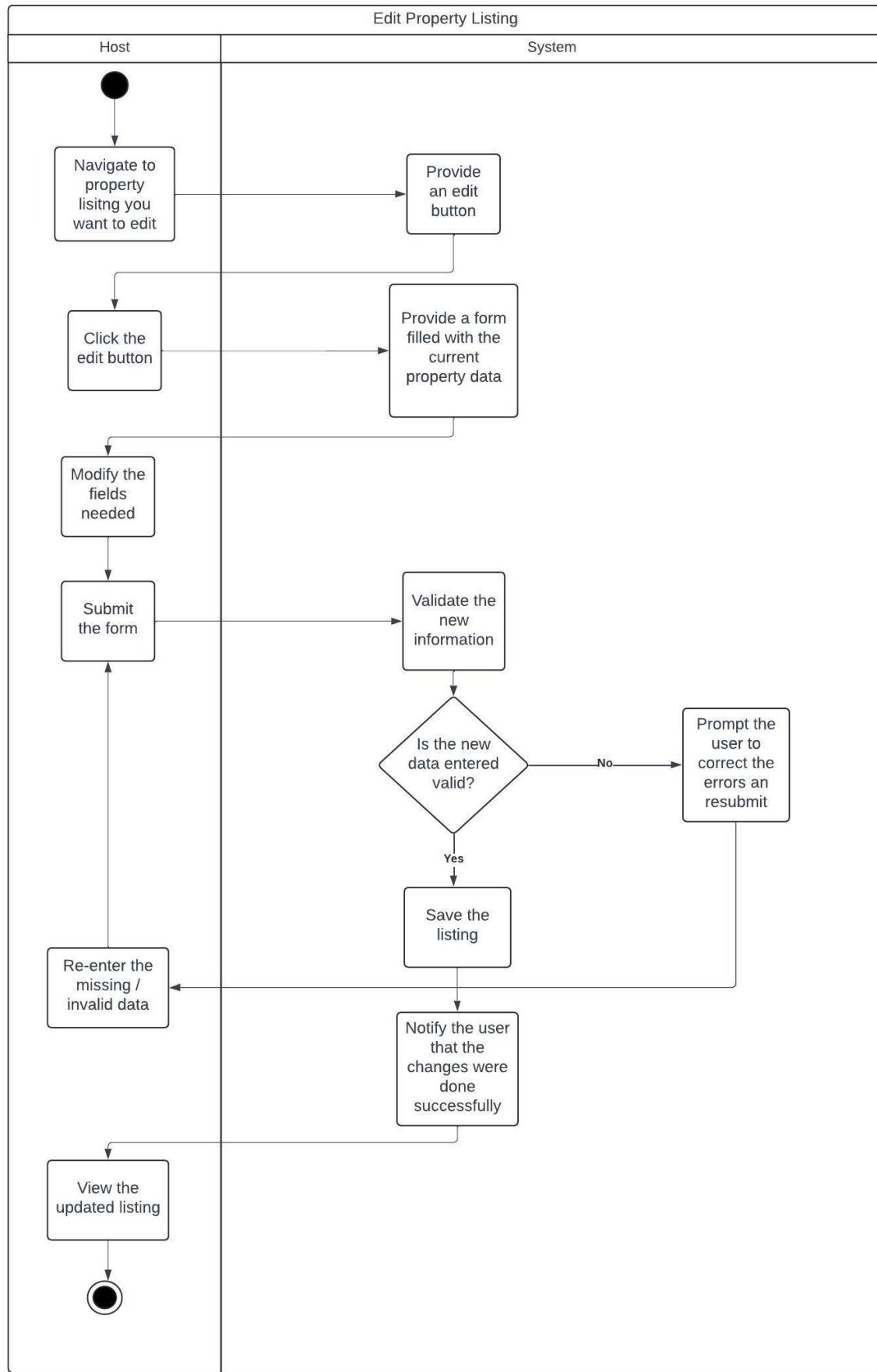
## Property Rentals Management System Requirements Specification



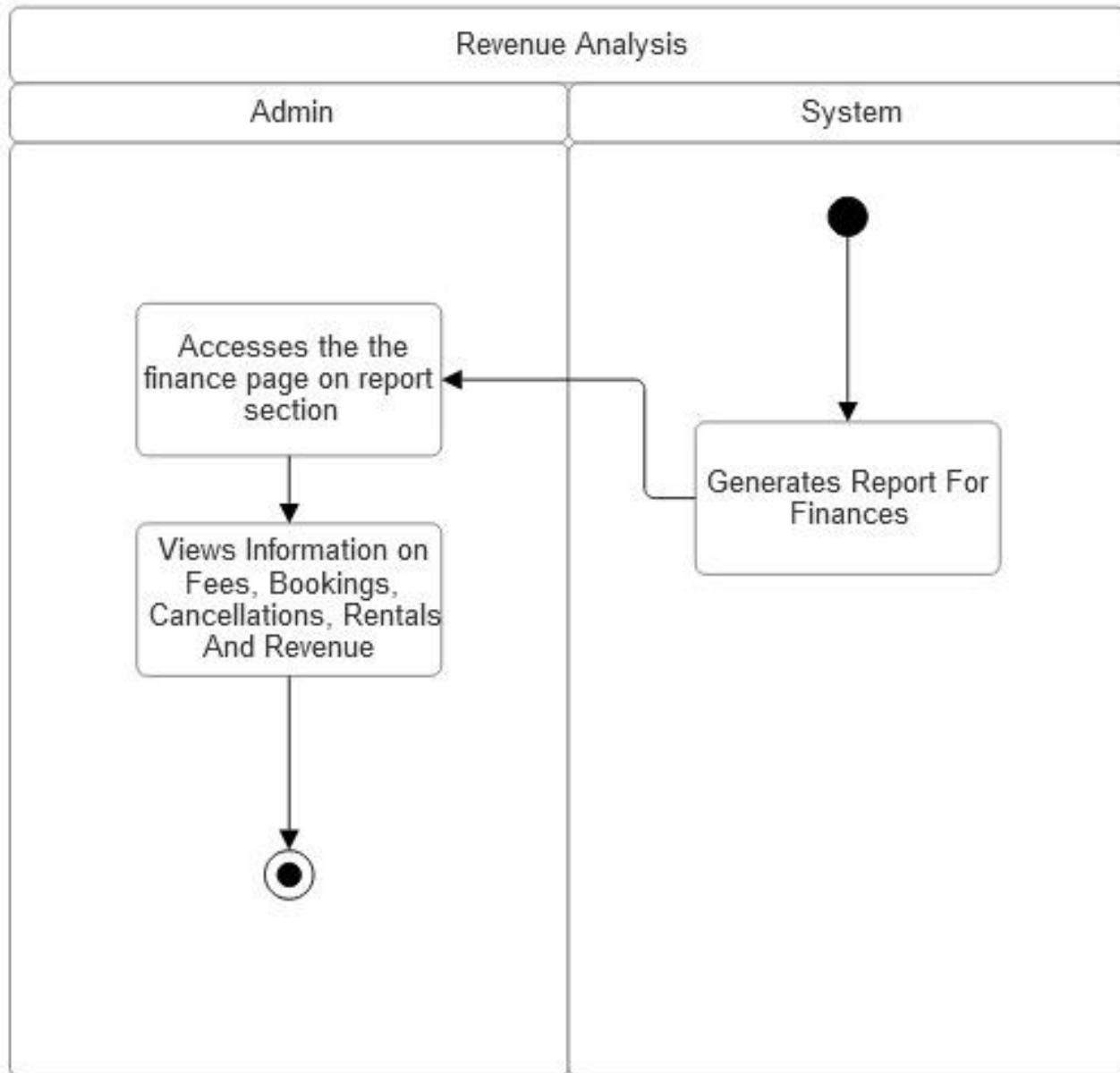
# Property Rentals Management System Requirements Specification



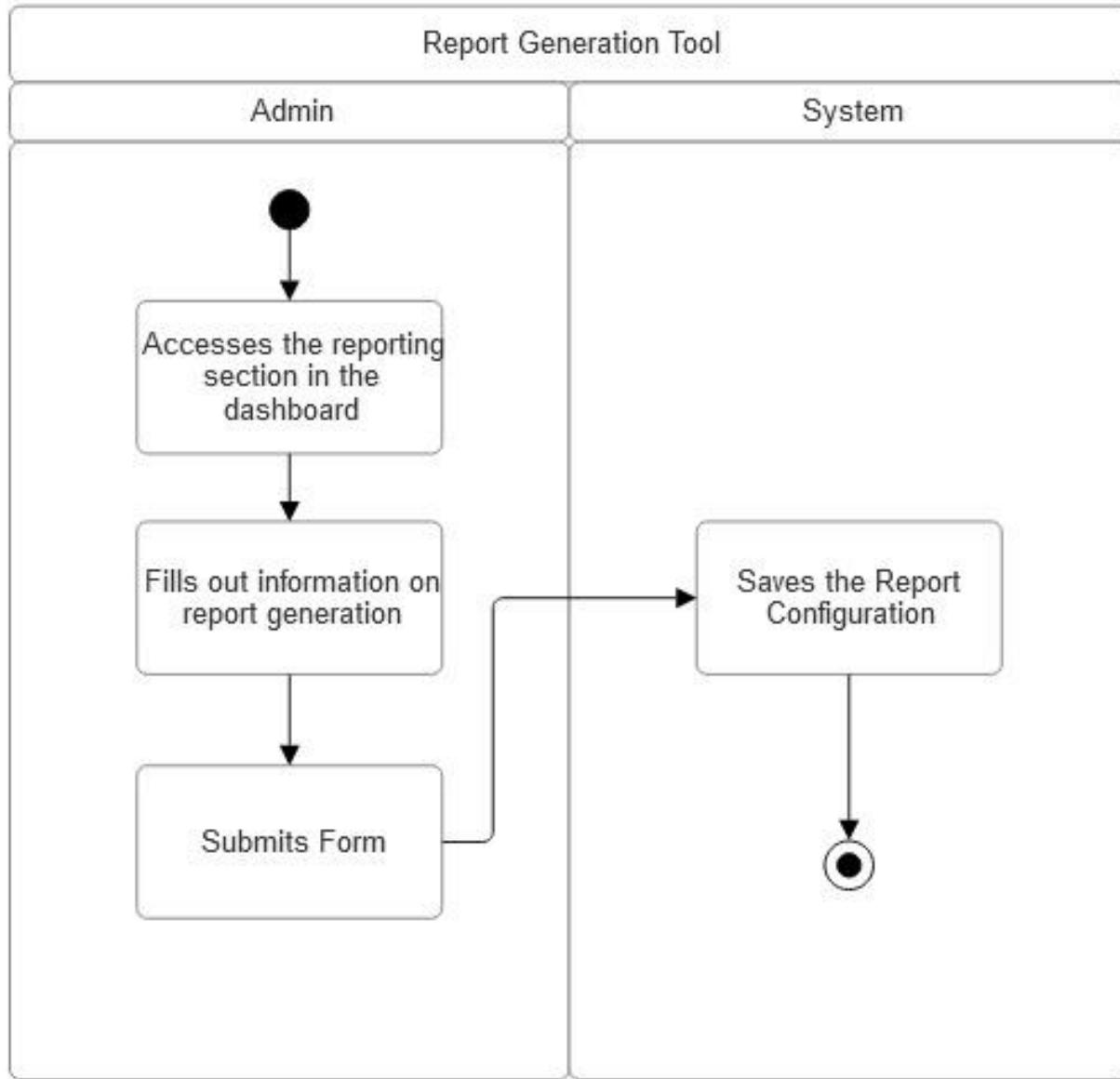
## Property Rentals Management System Requirements Specification



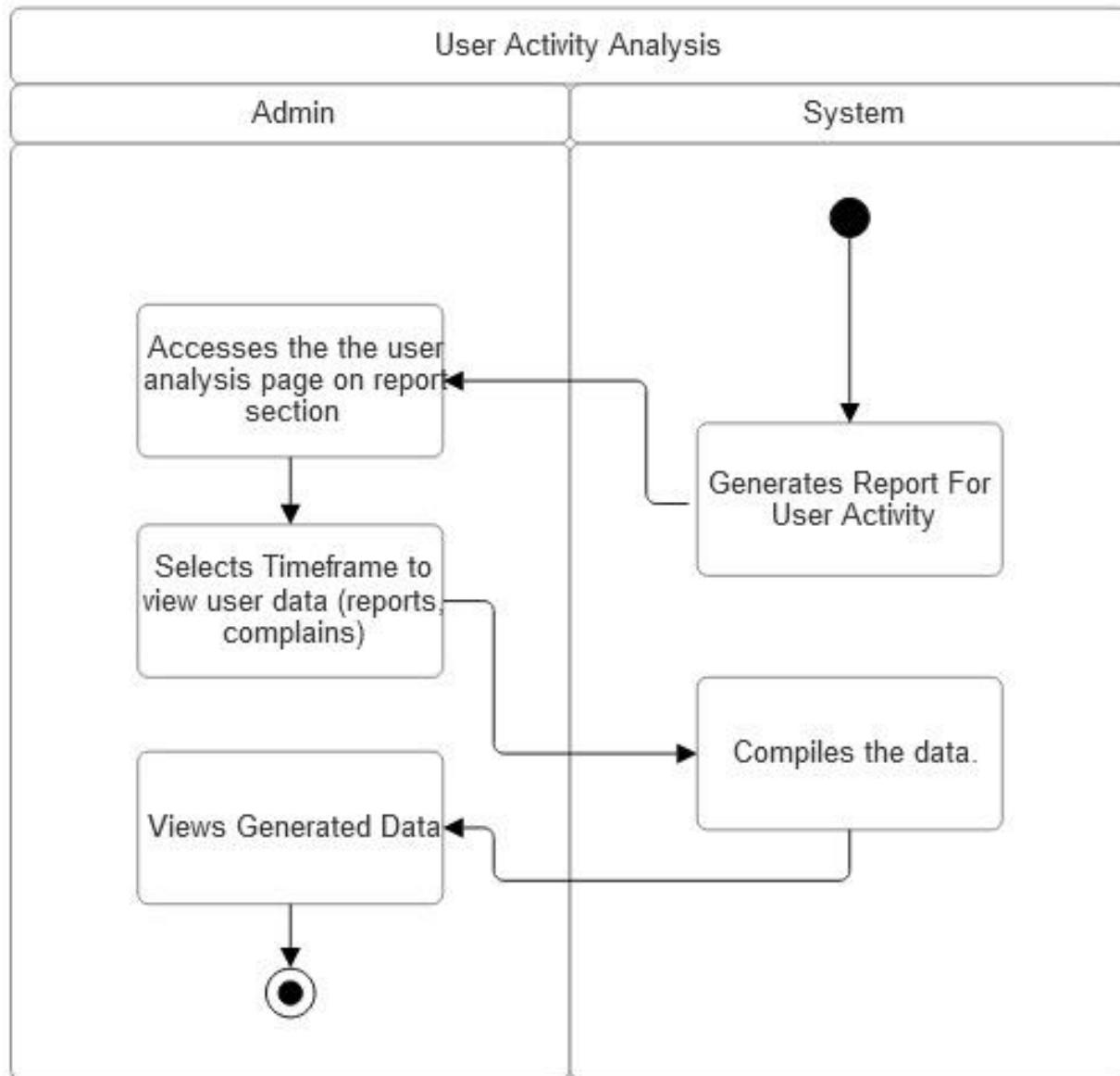
## Property Rentals Management System Requirements Specification



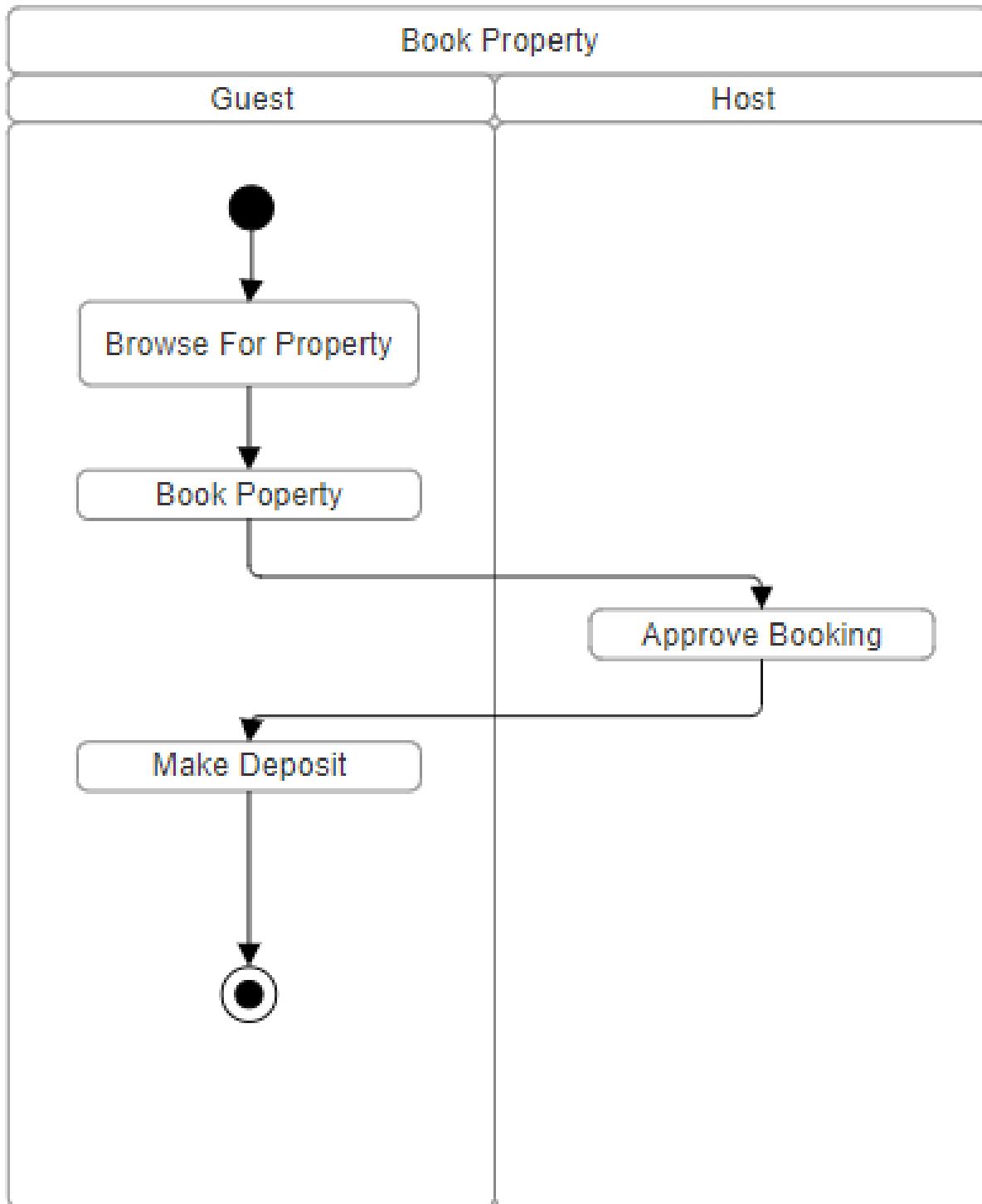
## Property Rentals Management System Requirements Specification



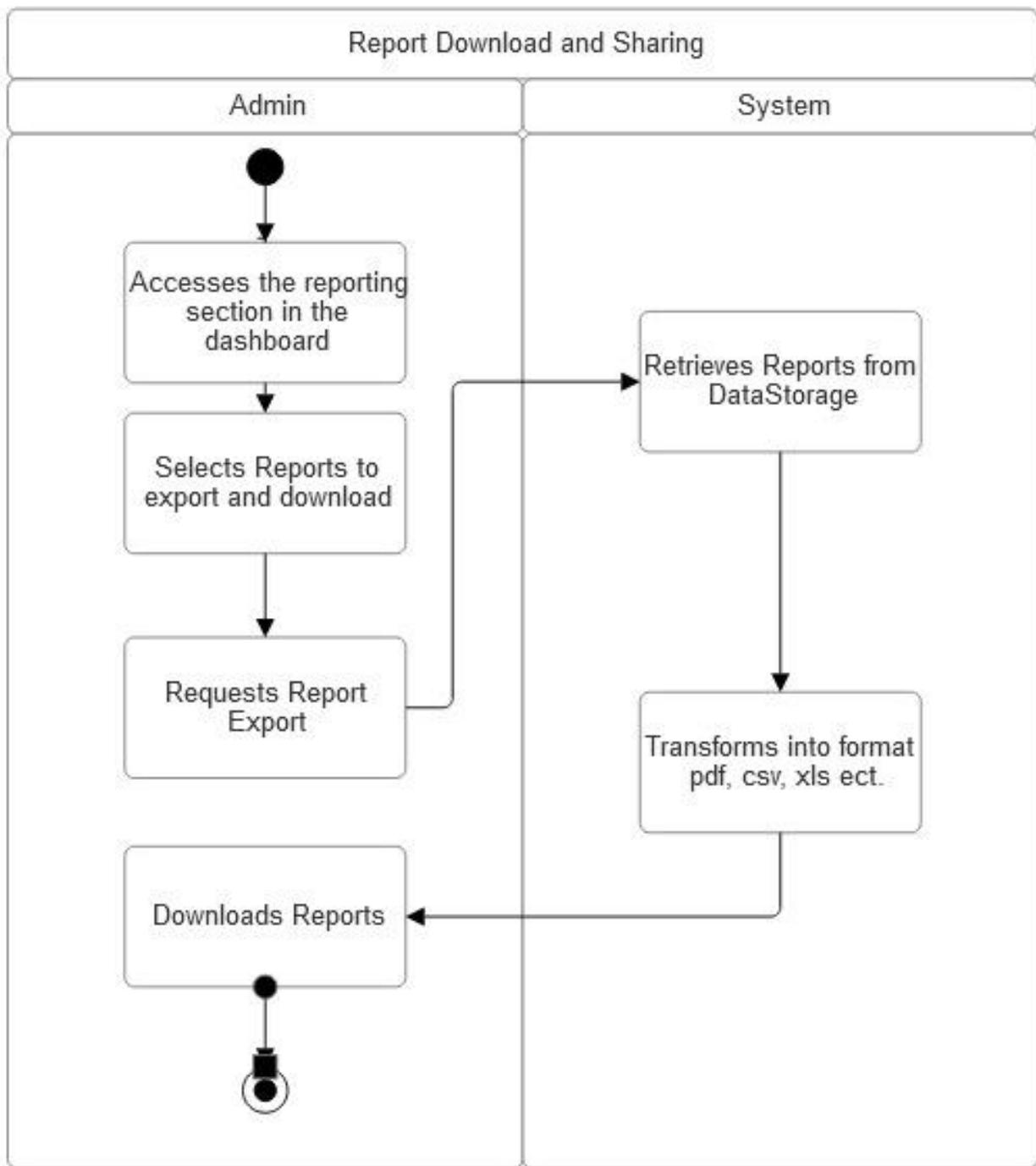
## Property Rentals Management System Requirements Specification



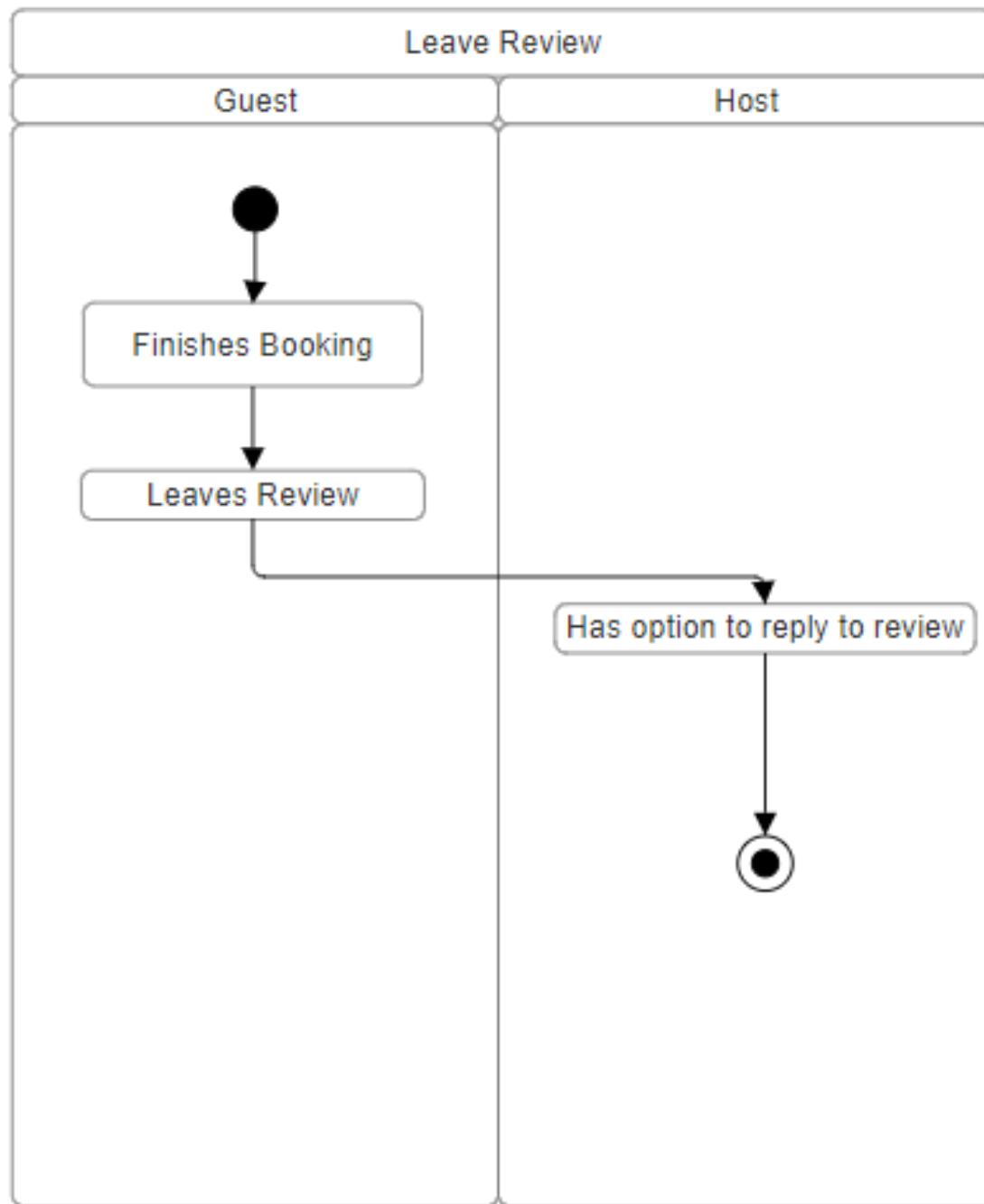
## Property Rentals Management System Requirements Specification



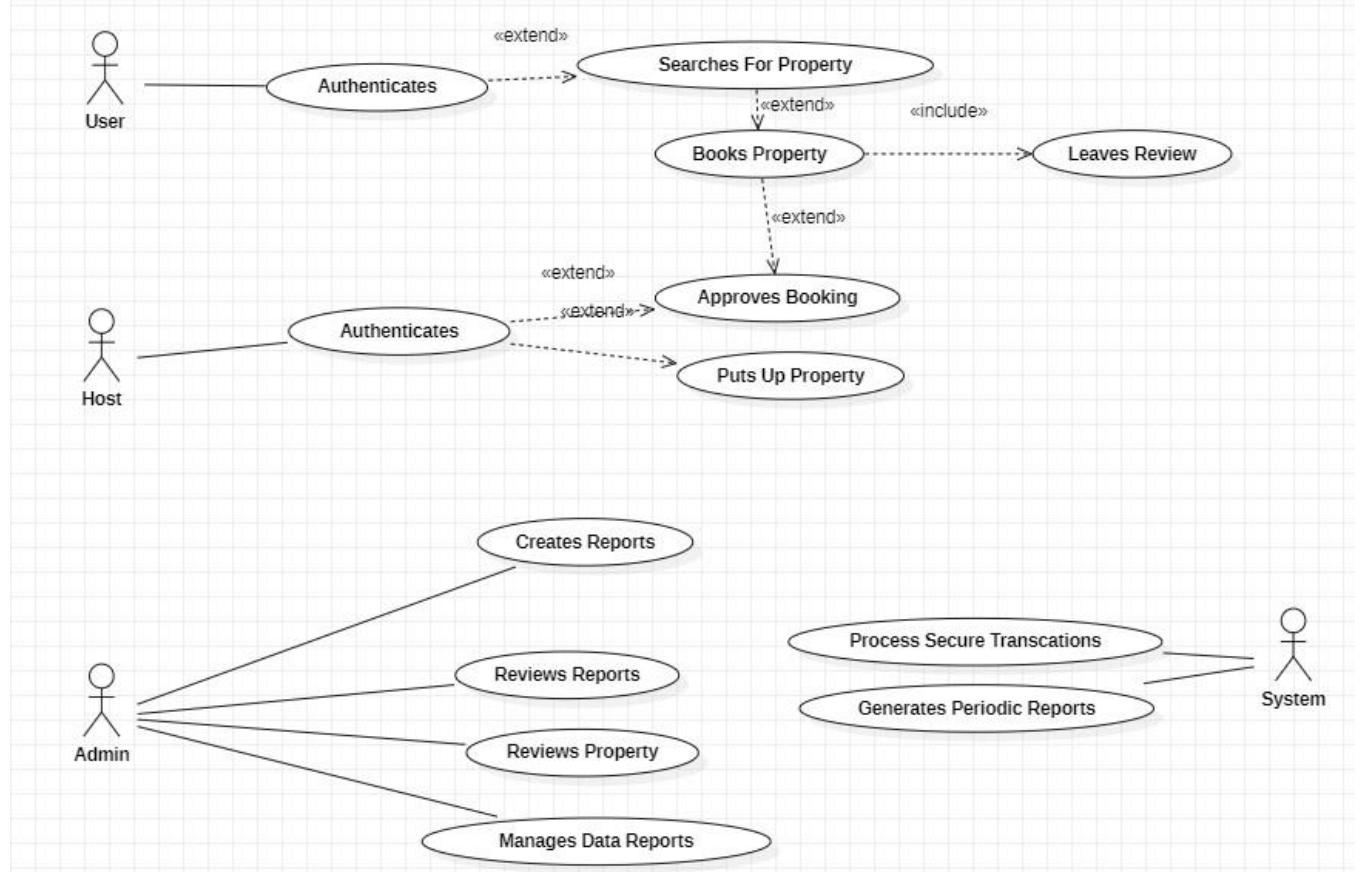
## Property Rentals Management System Requirements Specification



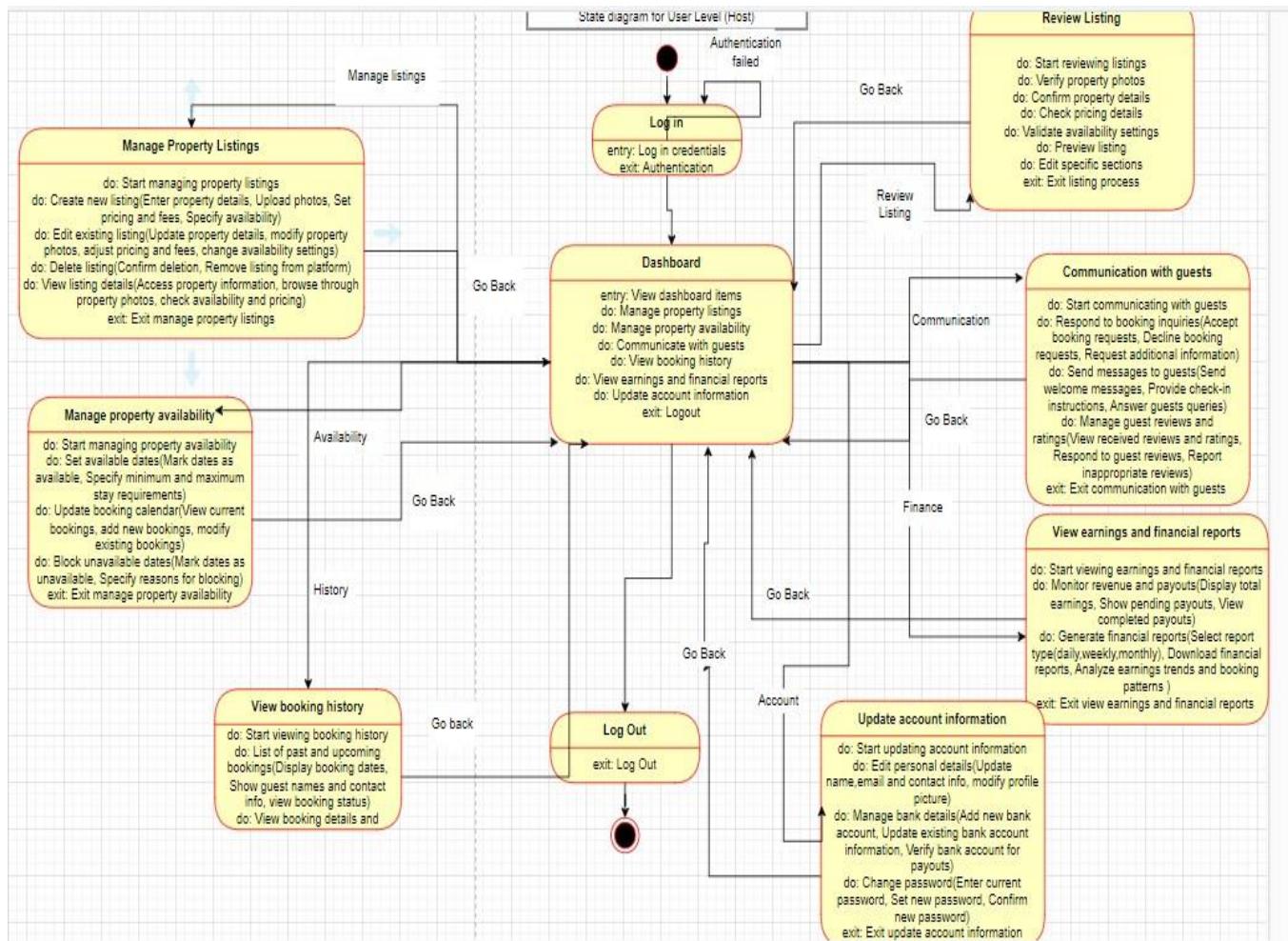
## Property Rentals Management System Requirements Specification



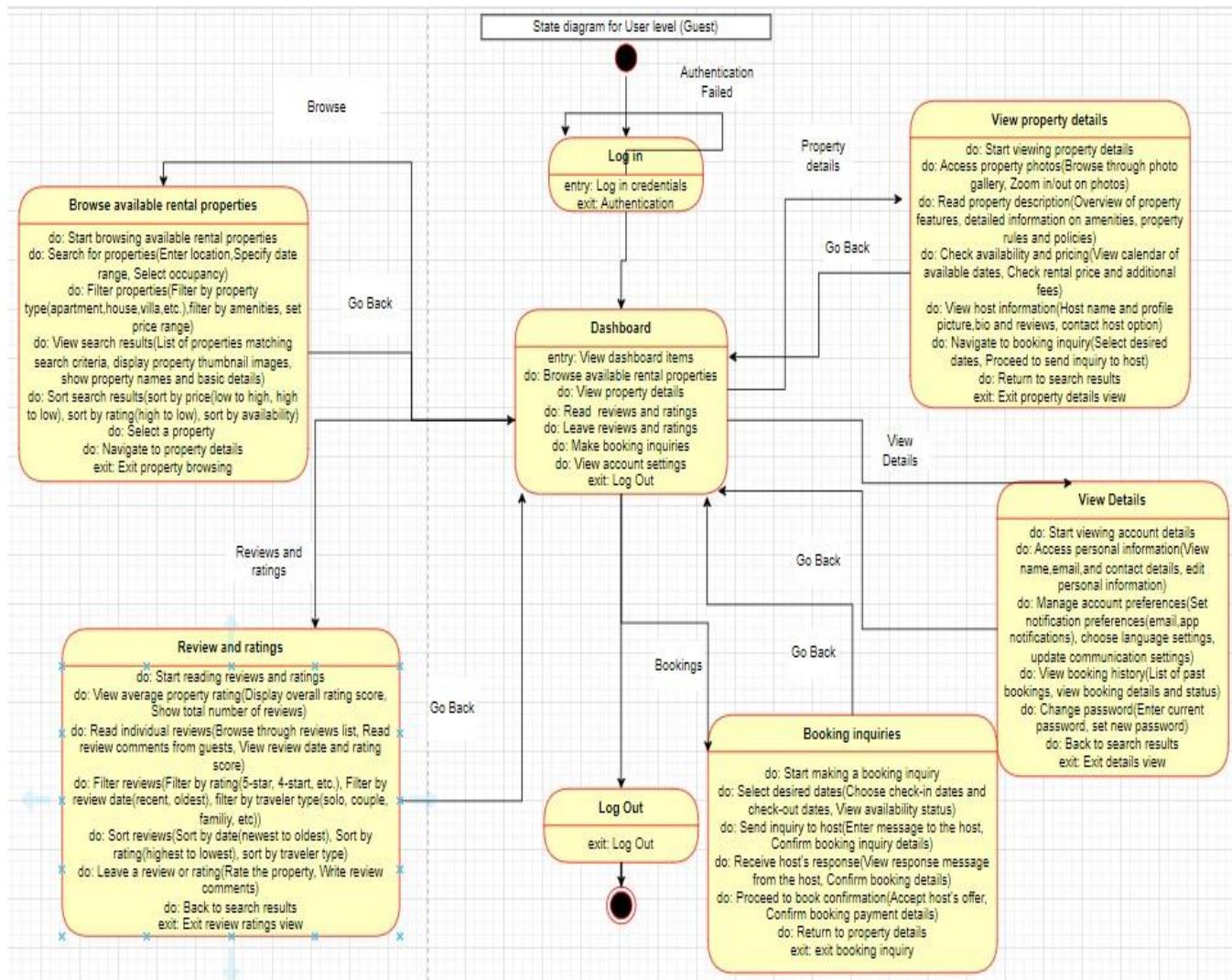
# USE CASE DIAGRAM



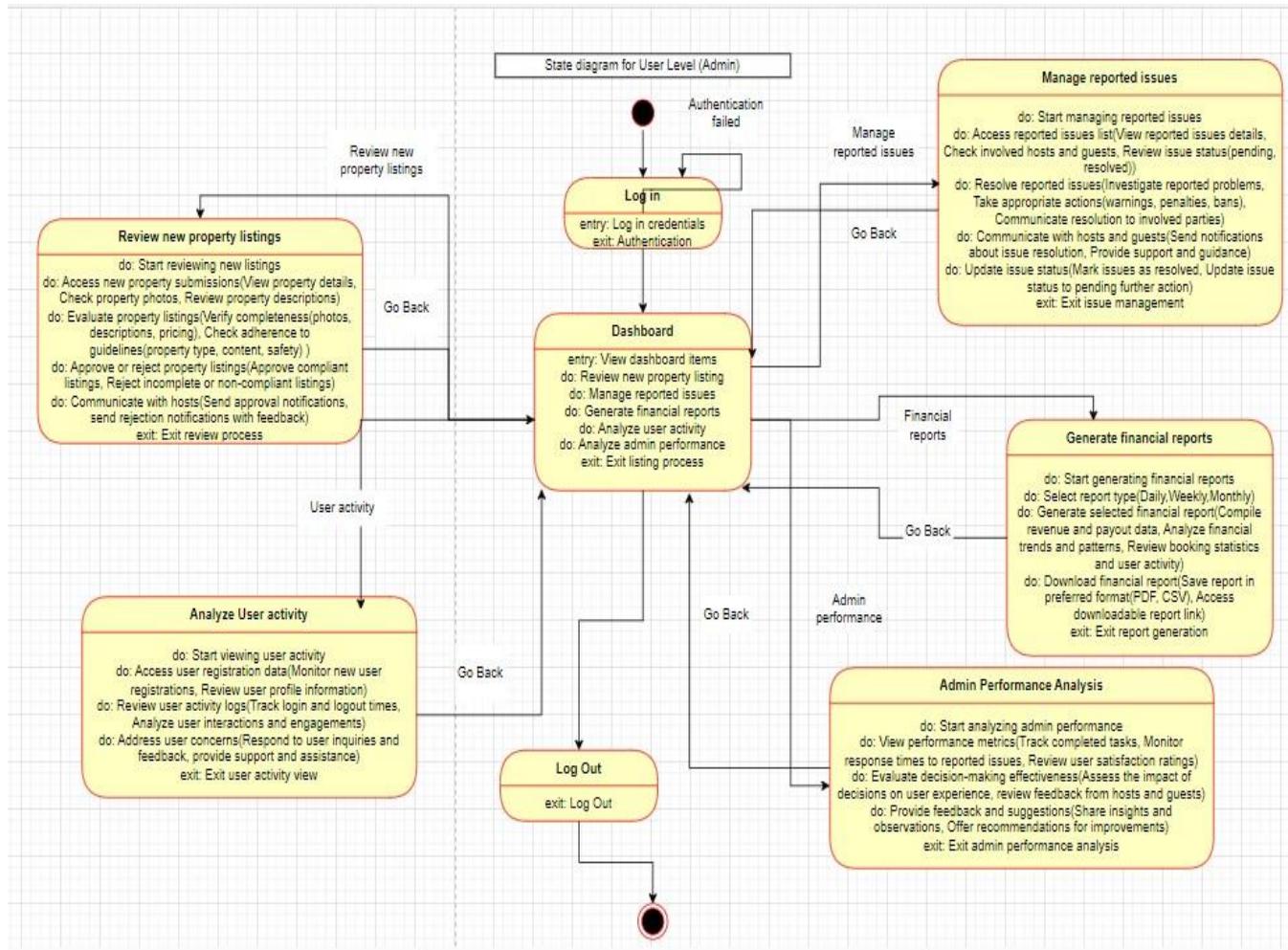
# STATE DIAGRAMS



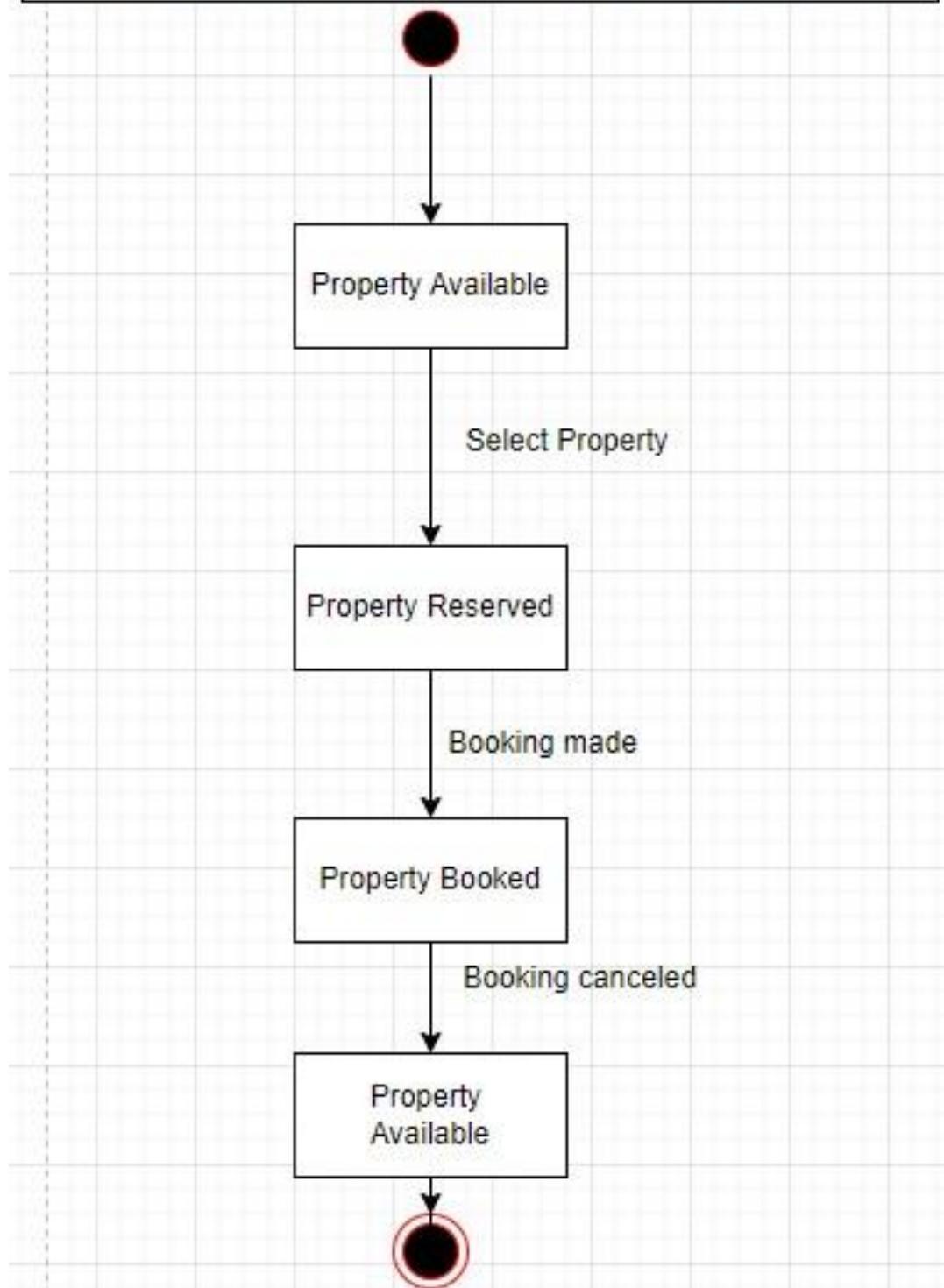
# Property Rentals Management System Requirements Specification



# Property Rentals Management System Requirements Specification

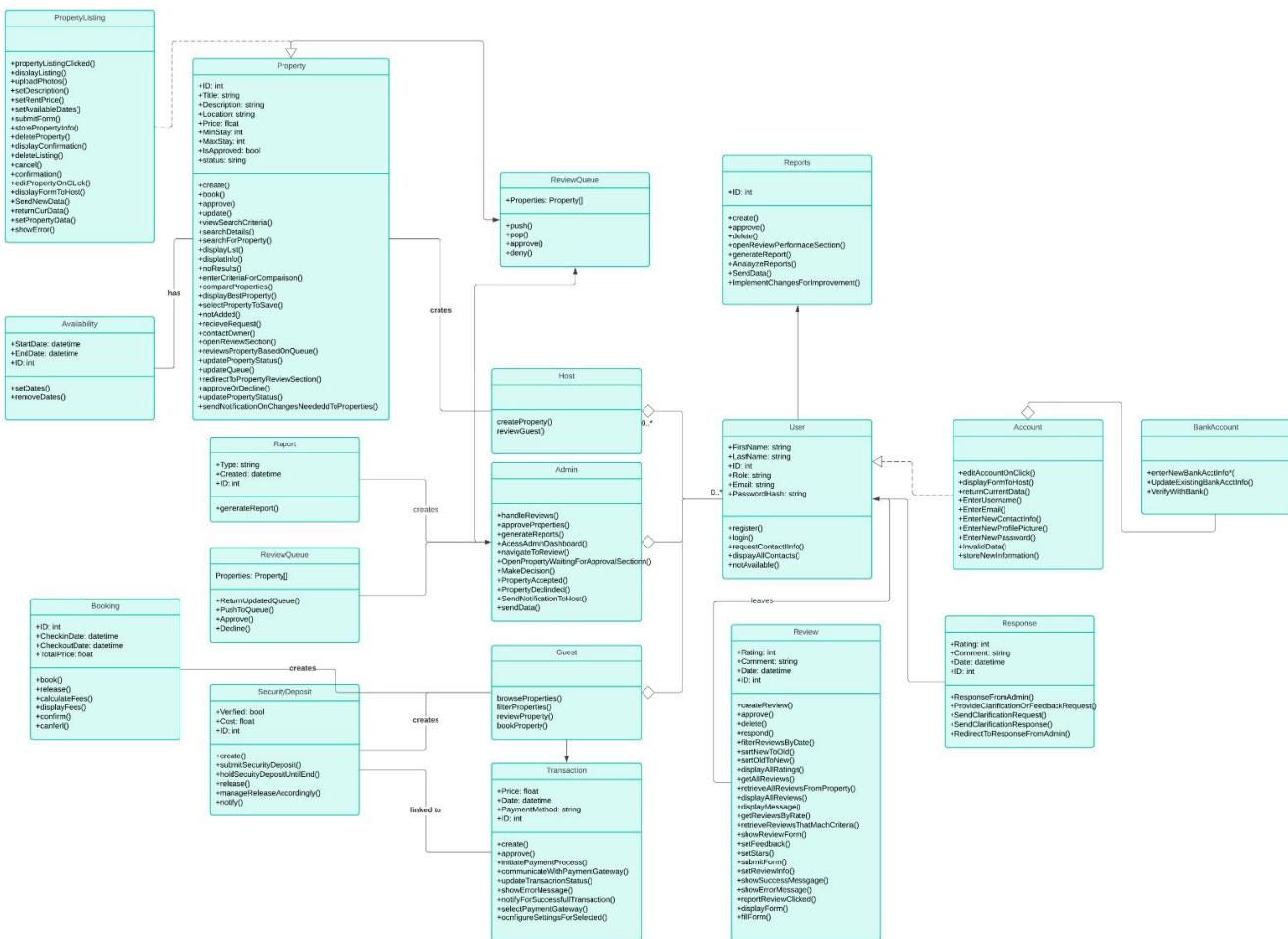


STATE DIAGRAM FOR THE PROPERTY STATUS INTERFACE



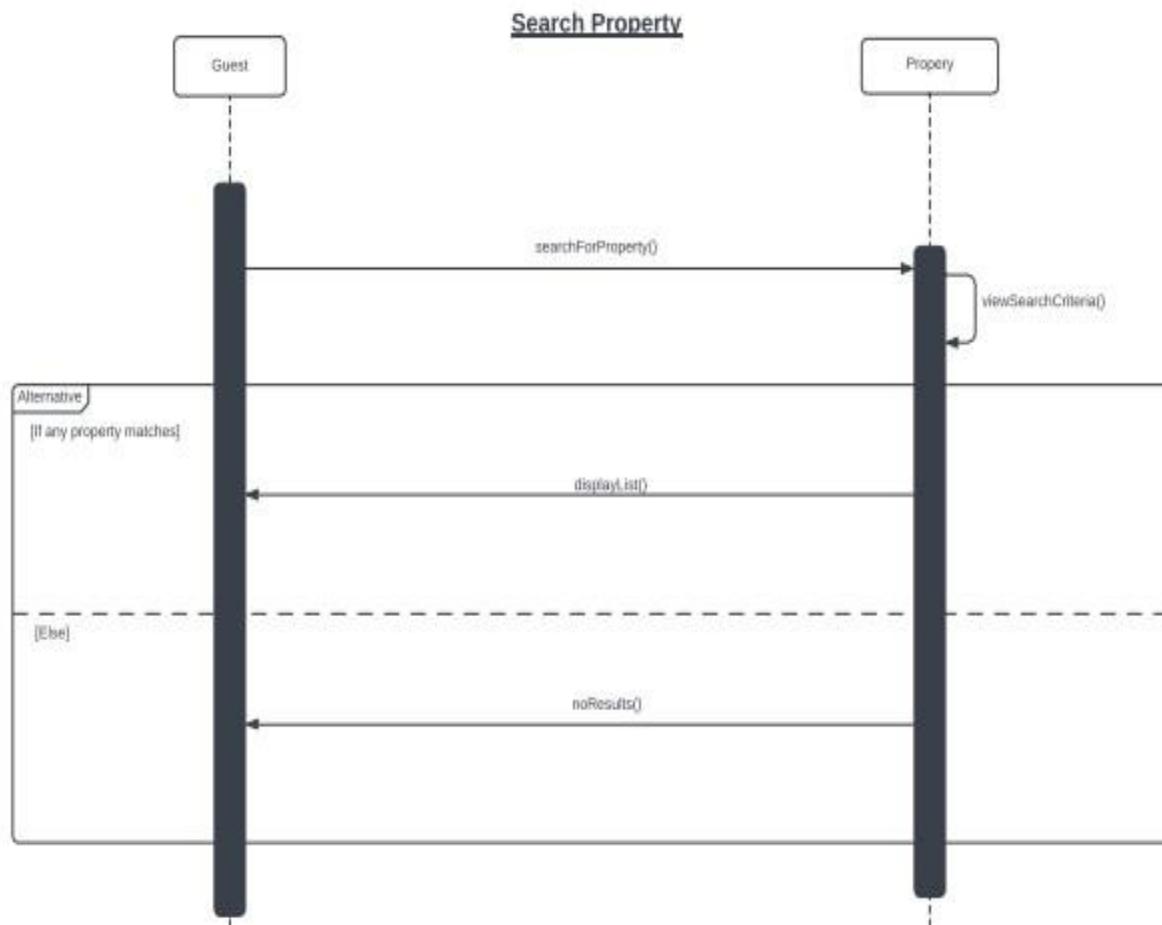
# Property Rentals Management System Requirements Specification

# CLASS DIAGRAM

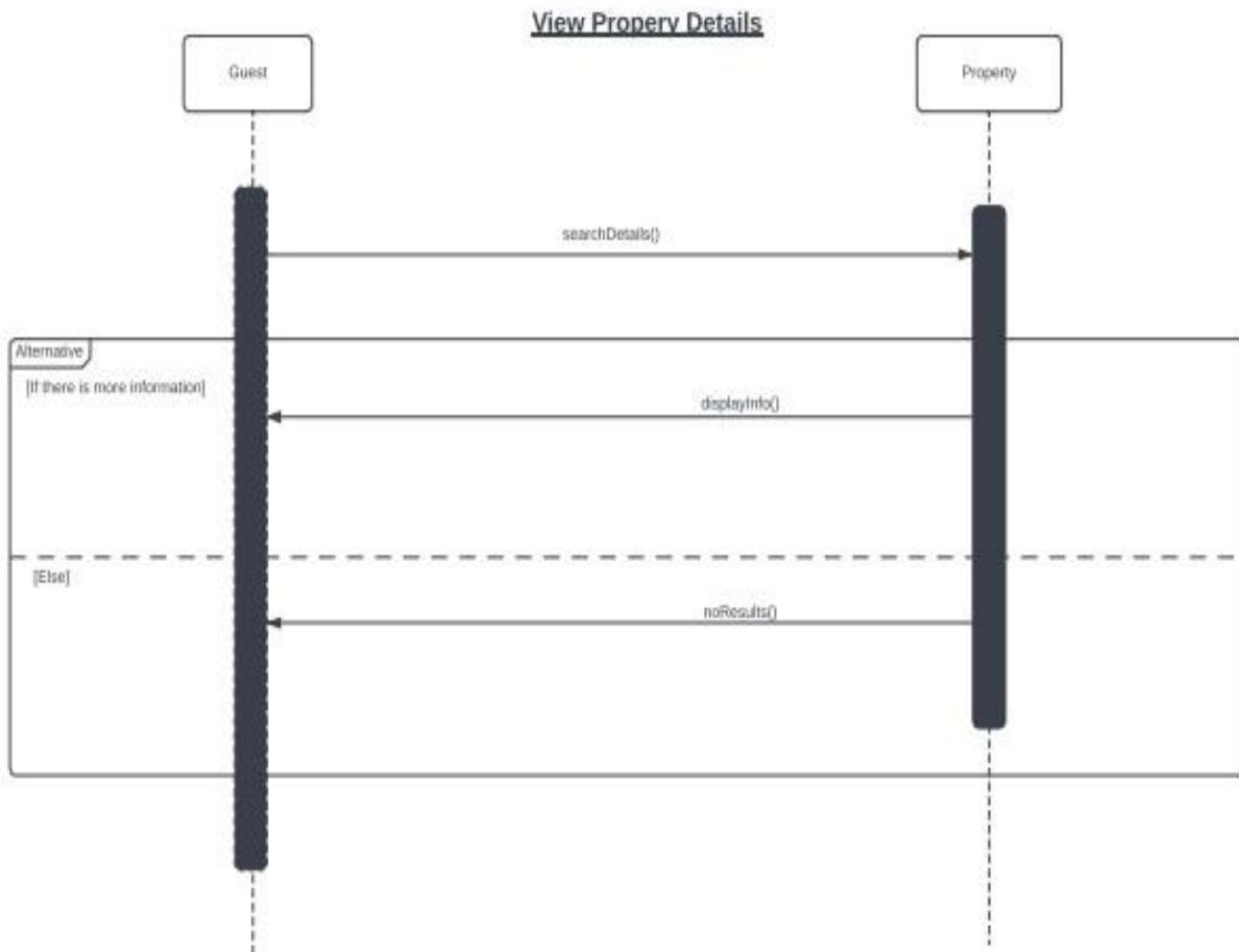


## SEQUENCE DIAGRAMS

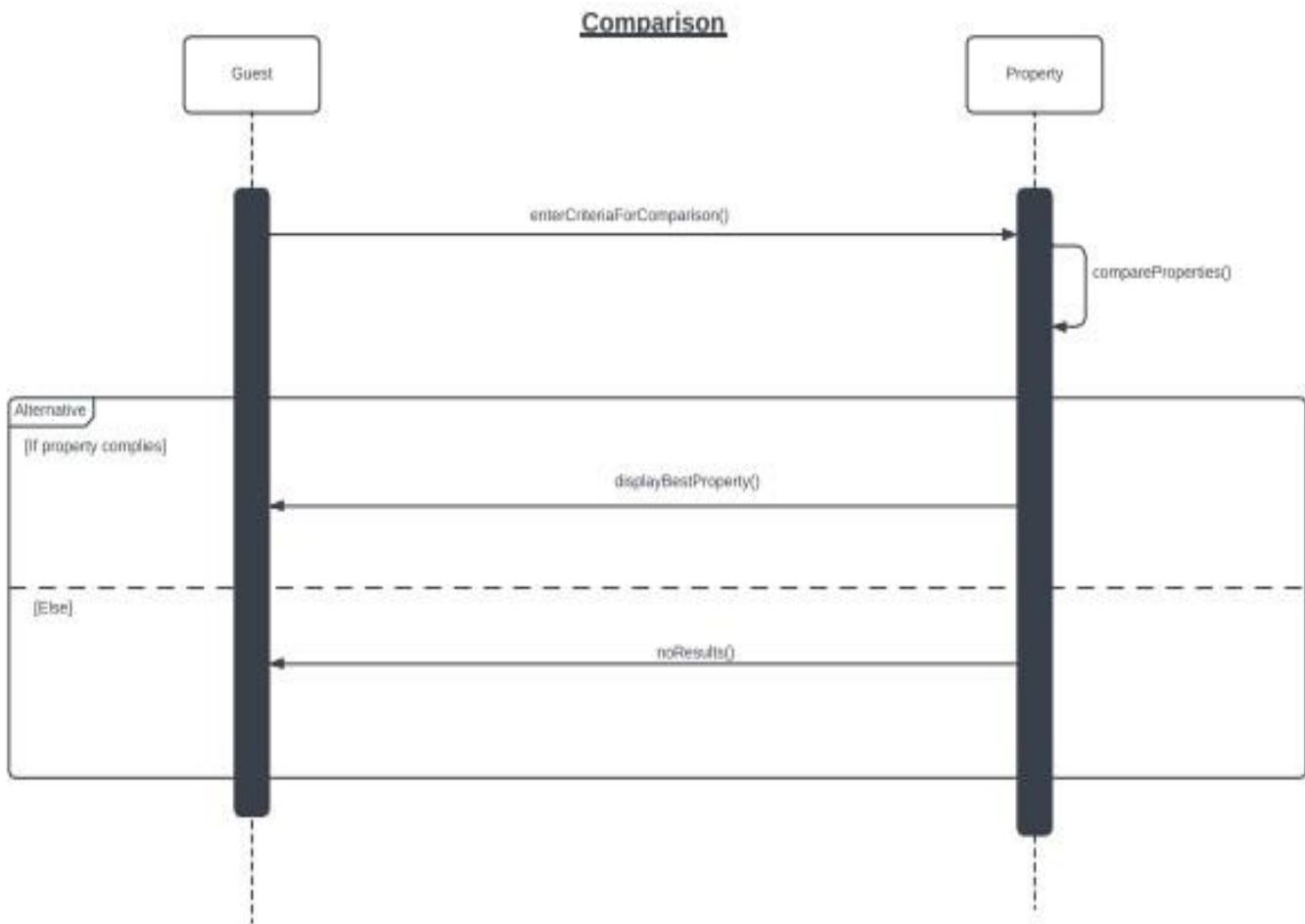
1.



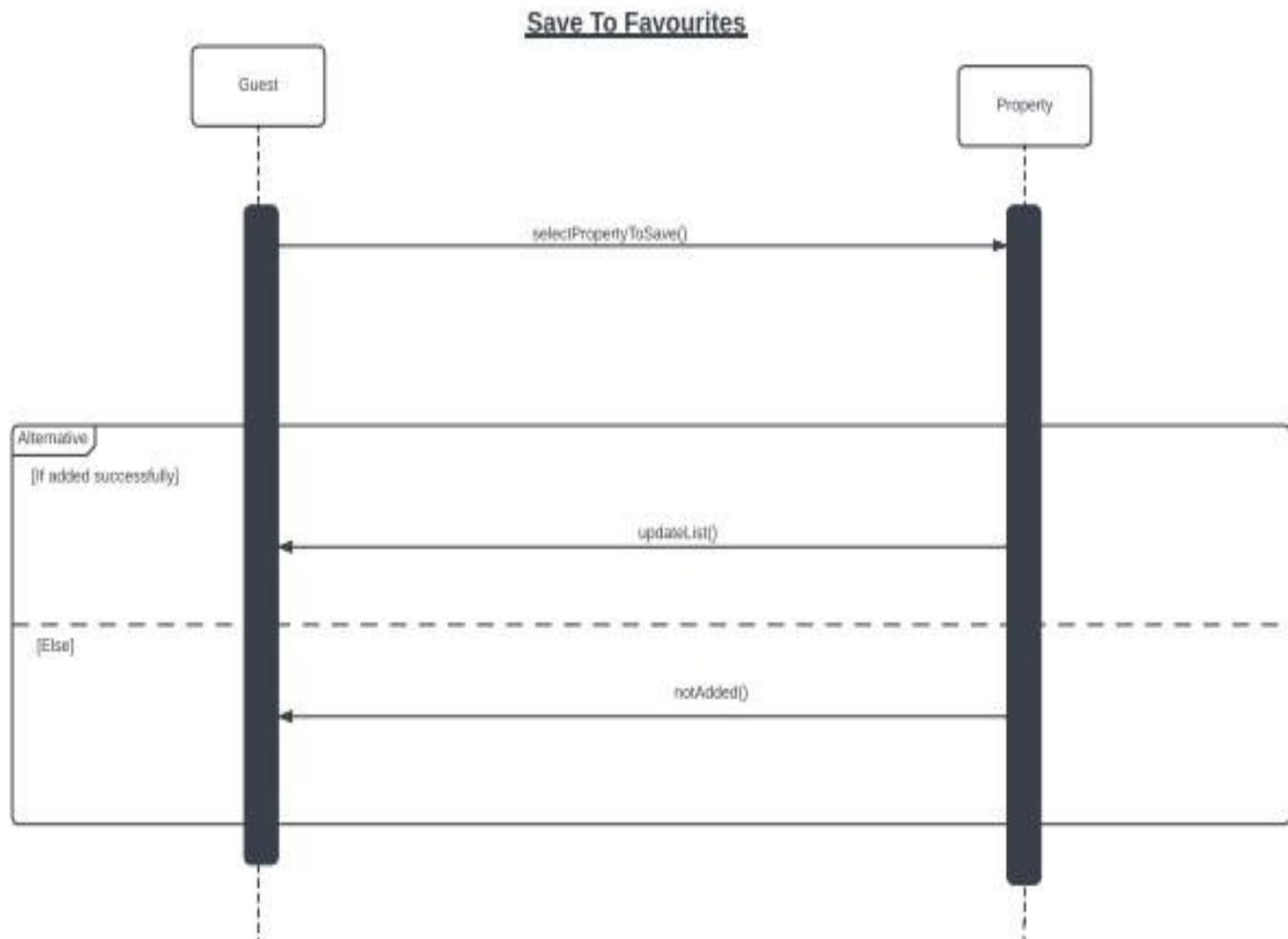
# Property Rentals Management System Requirements Specification



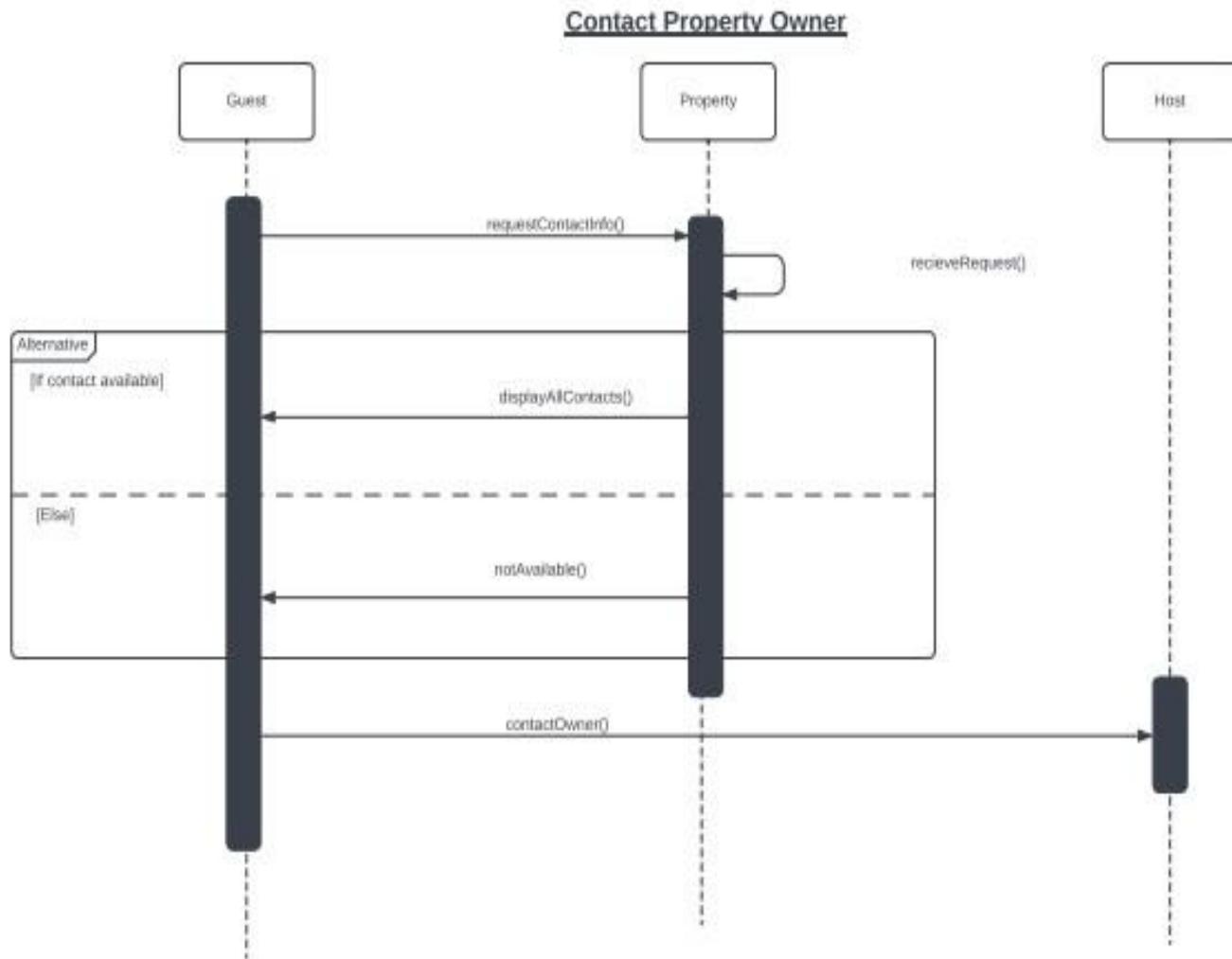
## Property Rentals Management System Requirements Specification



## Property Rentals Management System Requirements Specification



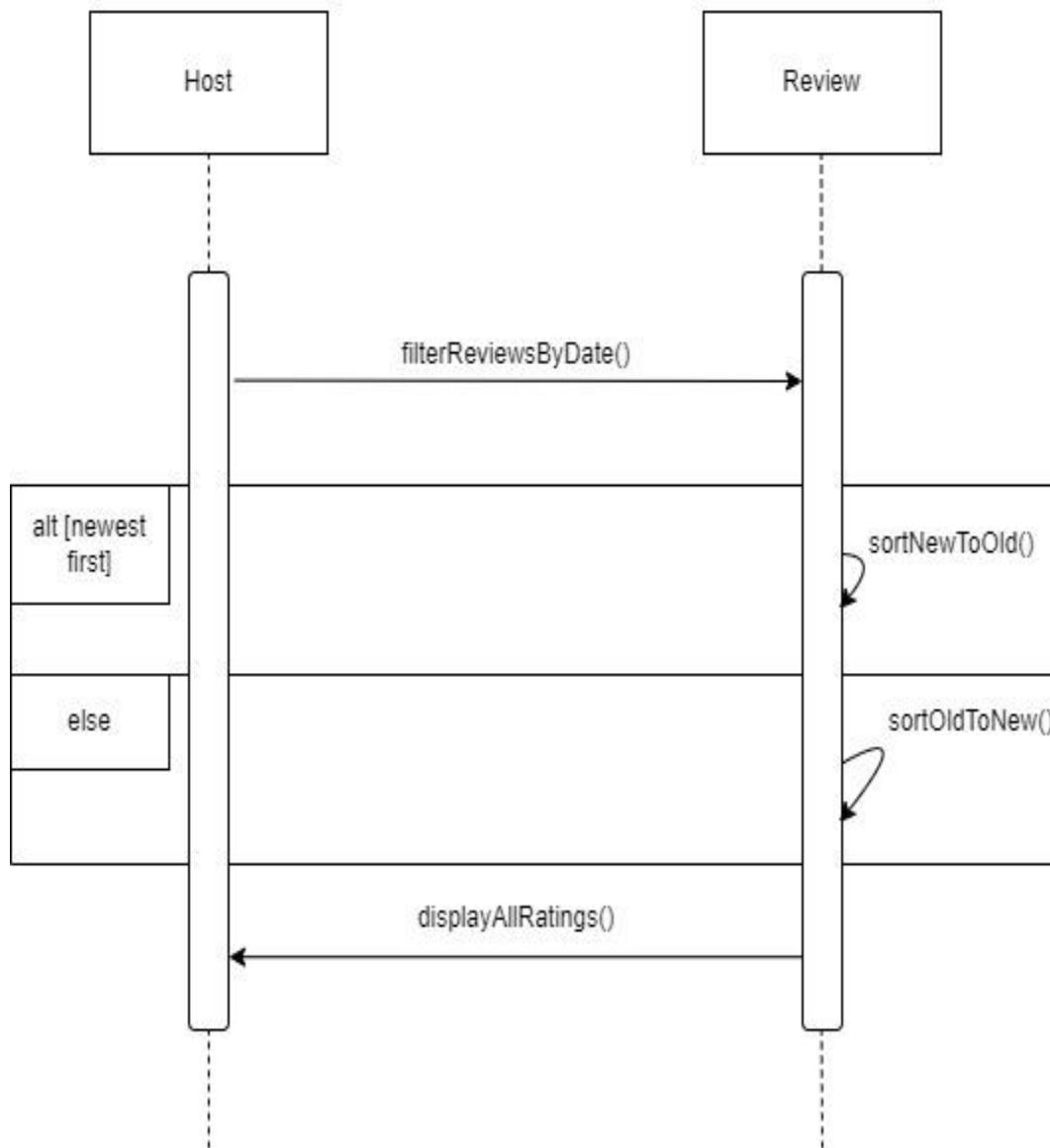
# Property Rentals Management System Requirements Specification



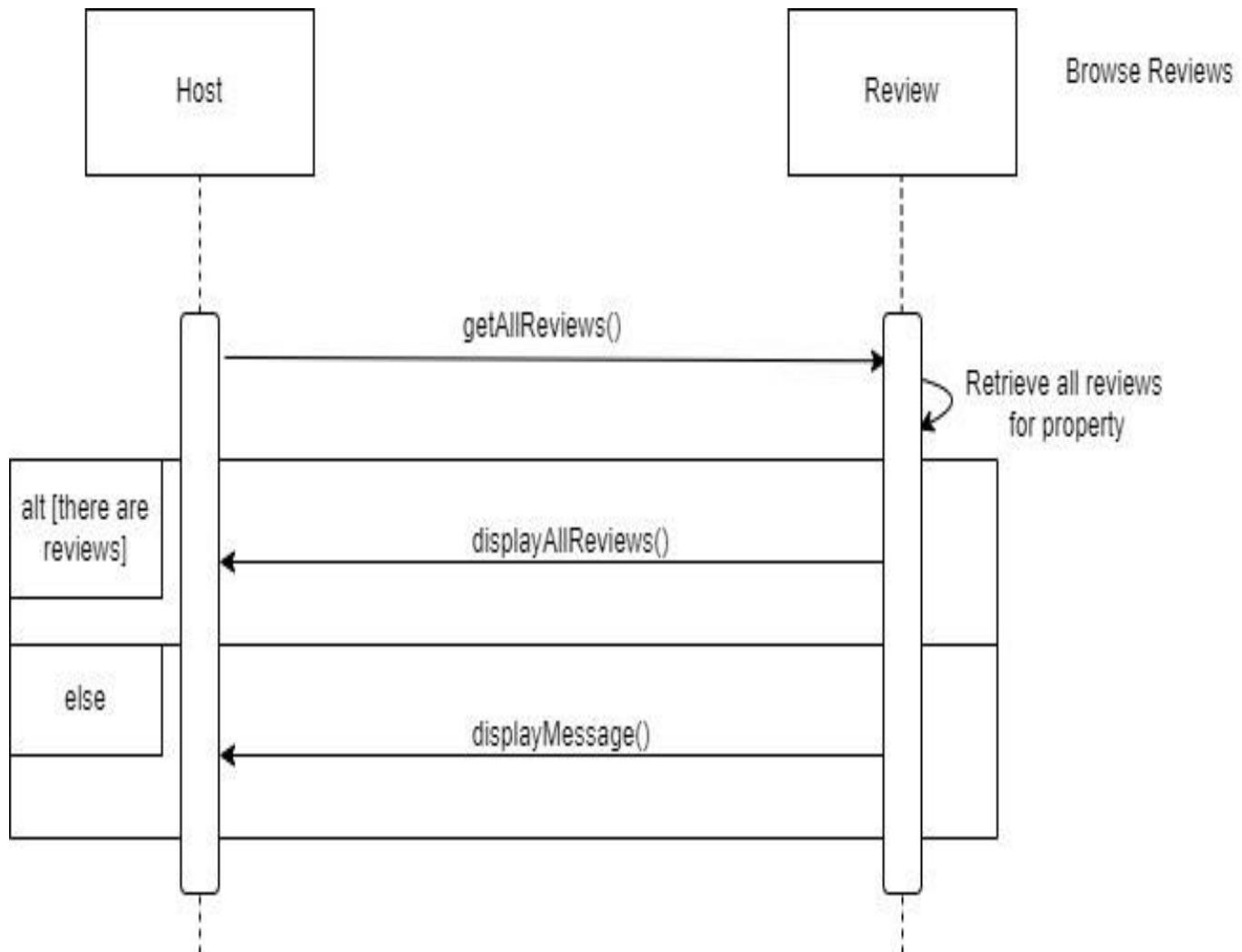
## Property Rentals Management System Requirements Specification

2.

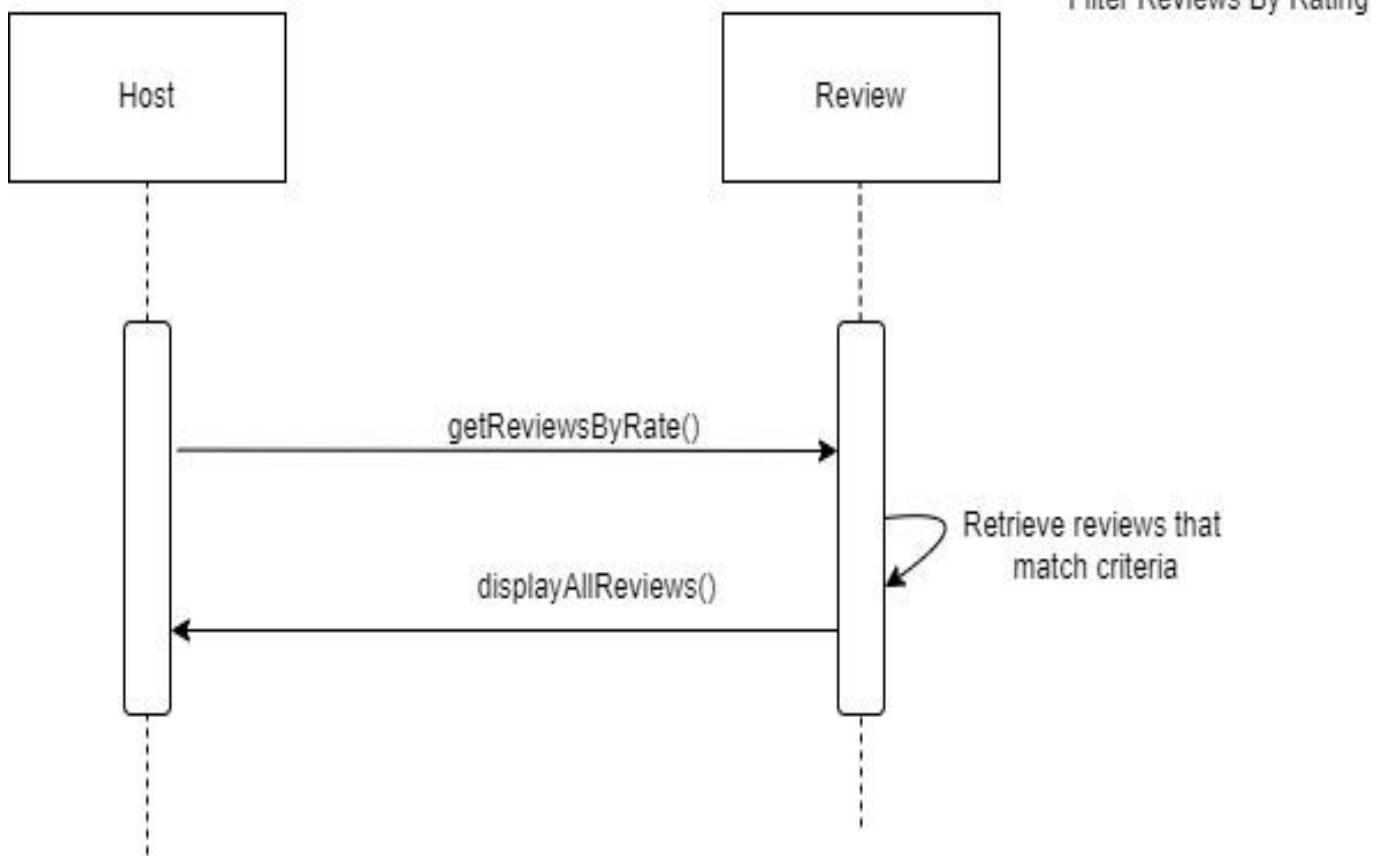
Filter Reviews By Date



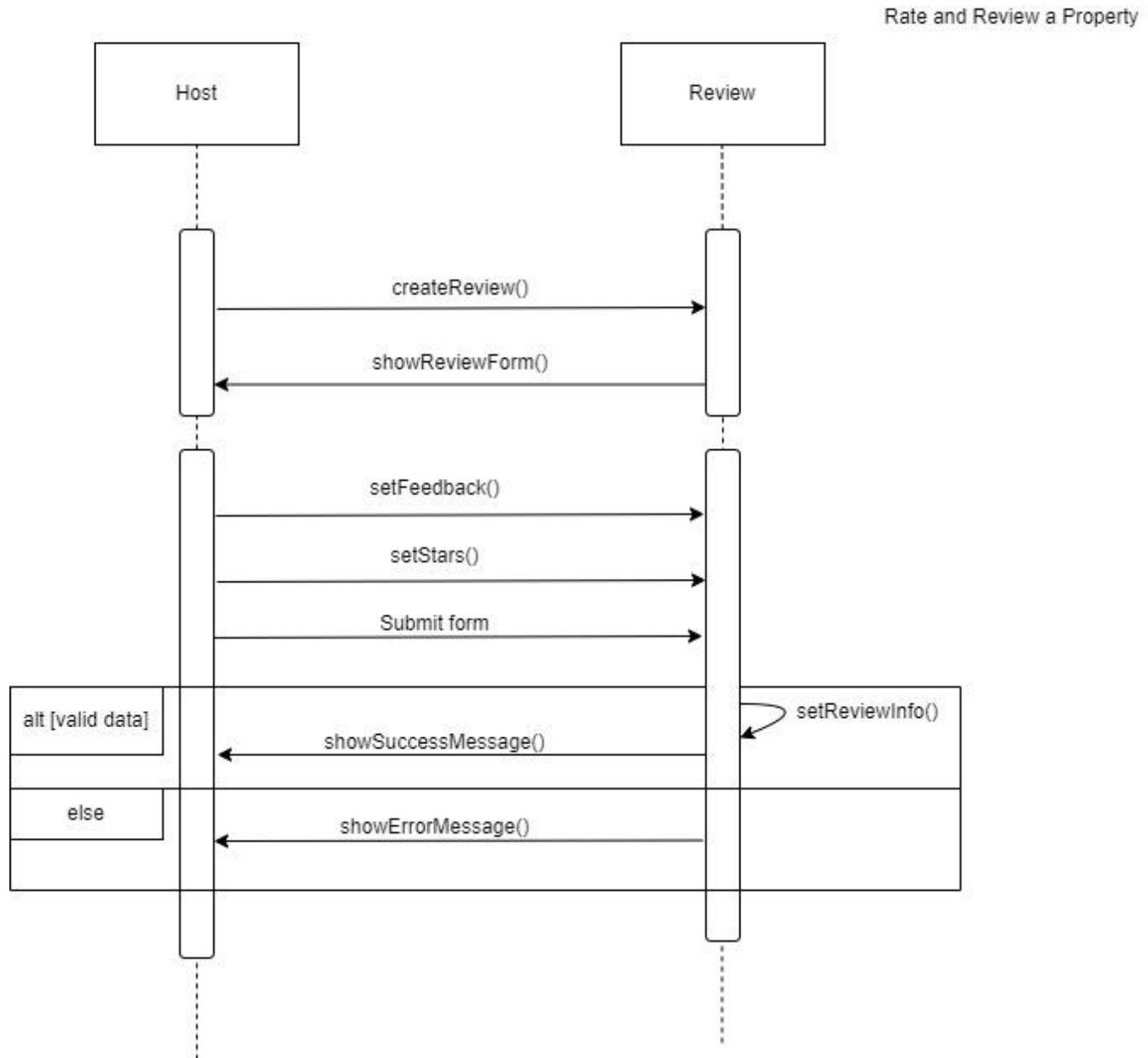
## Property Rentals Management System Requirements Specification



## Property Rentals Management System Requirements Specification

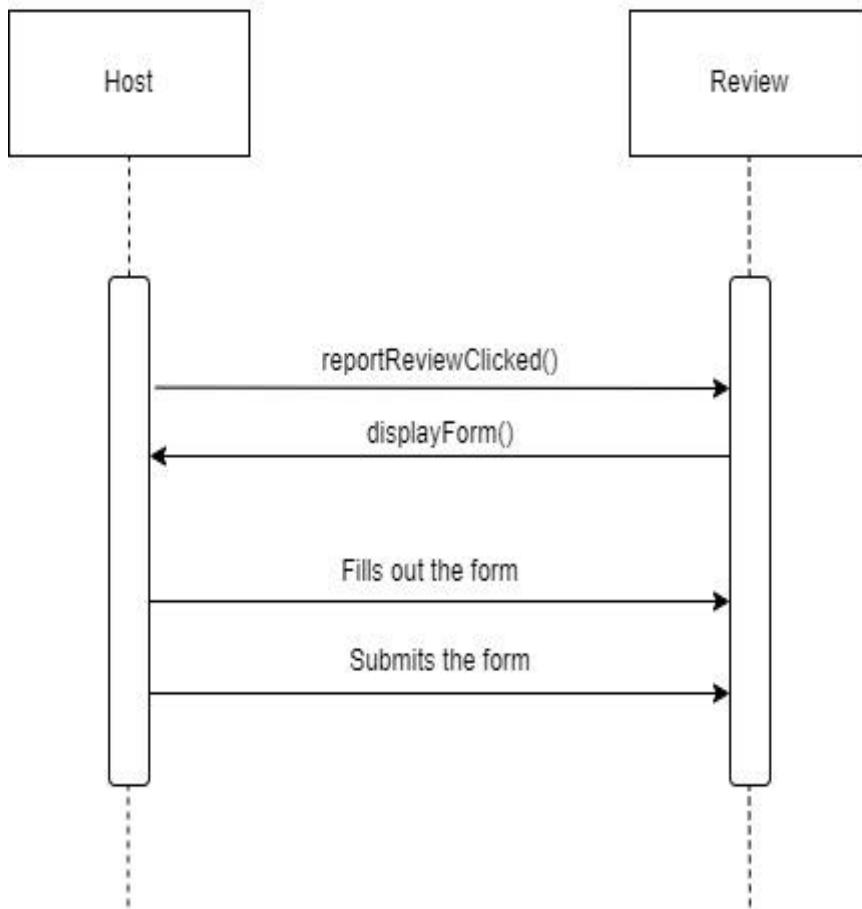


## Property Rentals Management System Requirements Specification



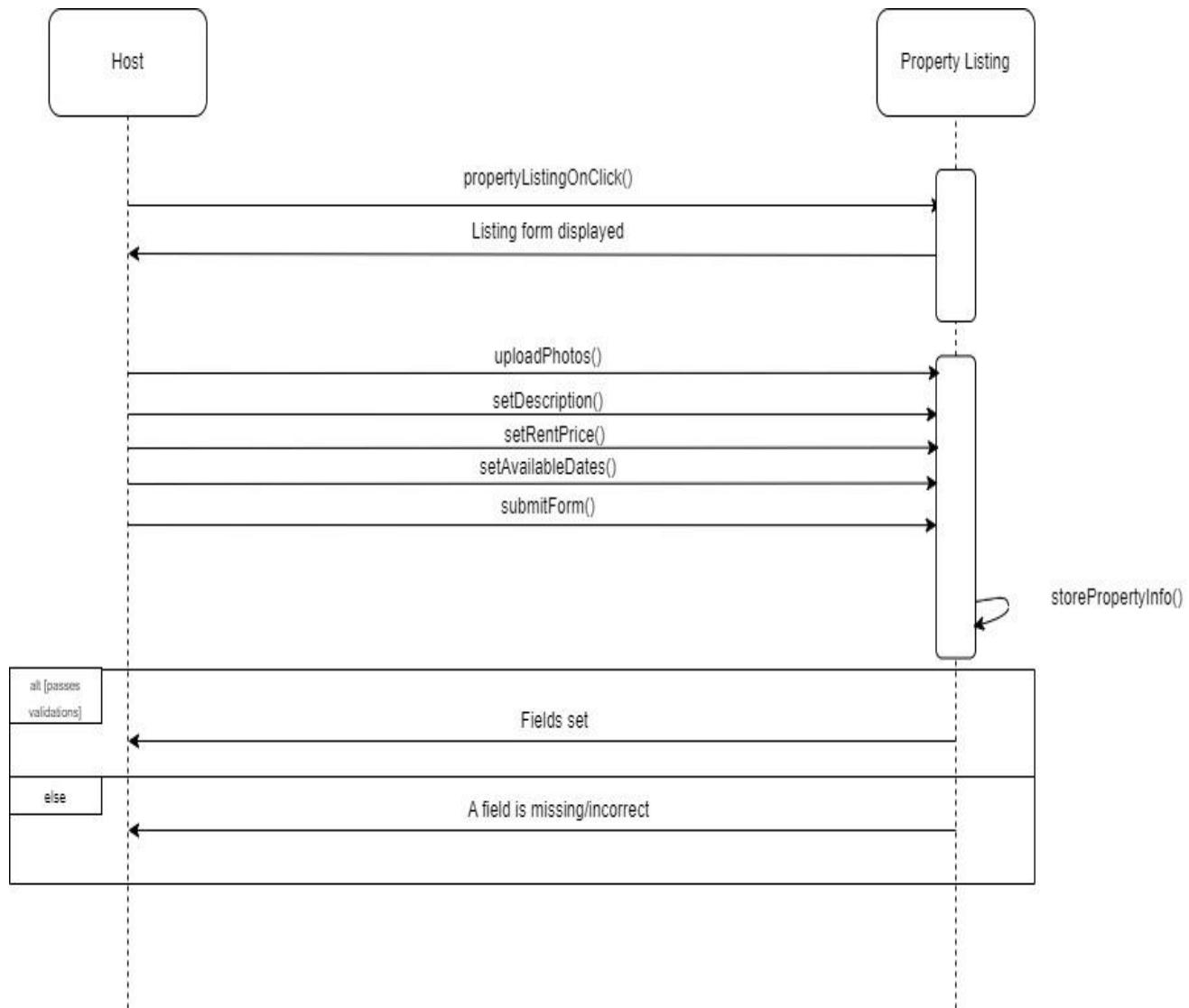
## Property Rentals Management System Requirements Specification

Report Inappropriate Reviews



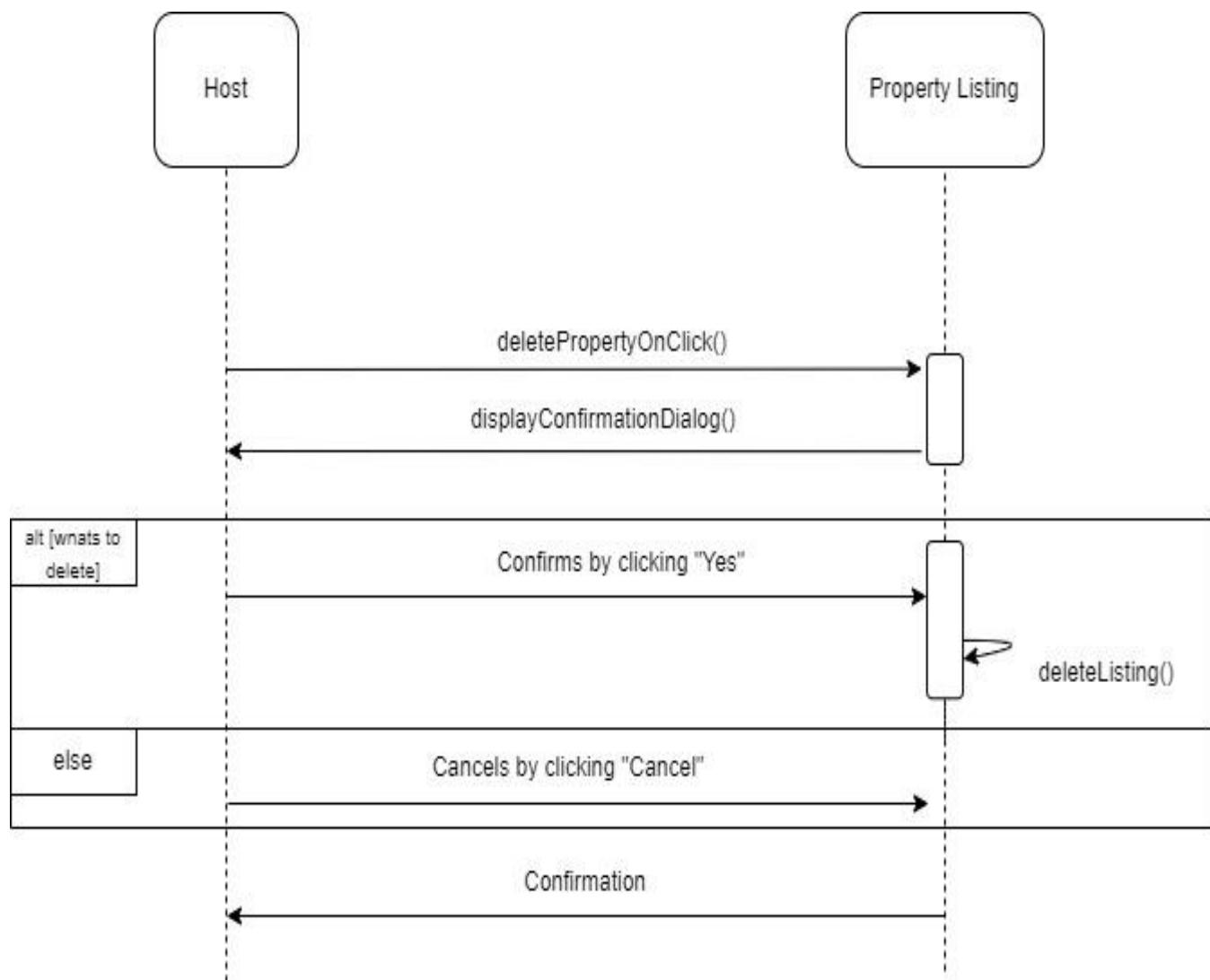
3.

## Create property Listing



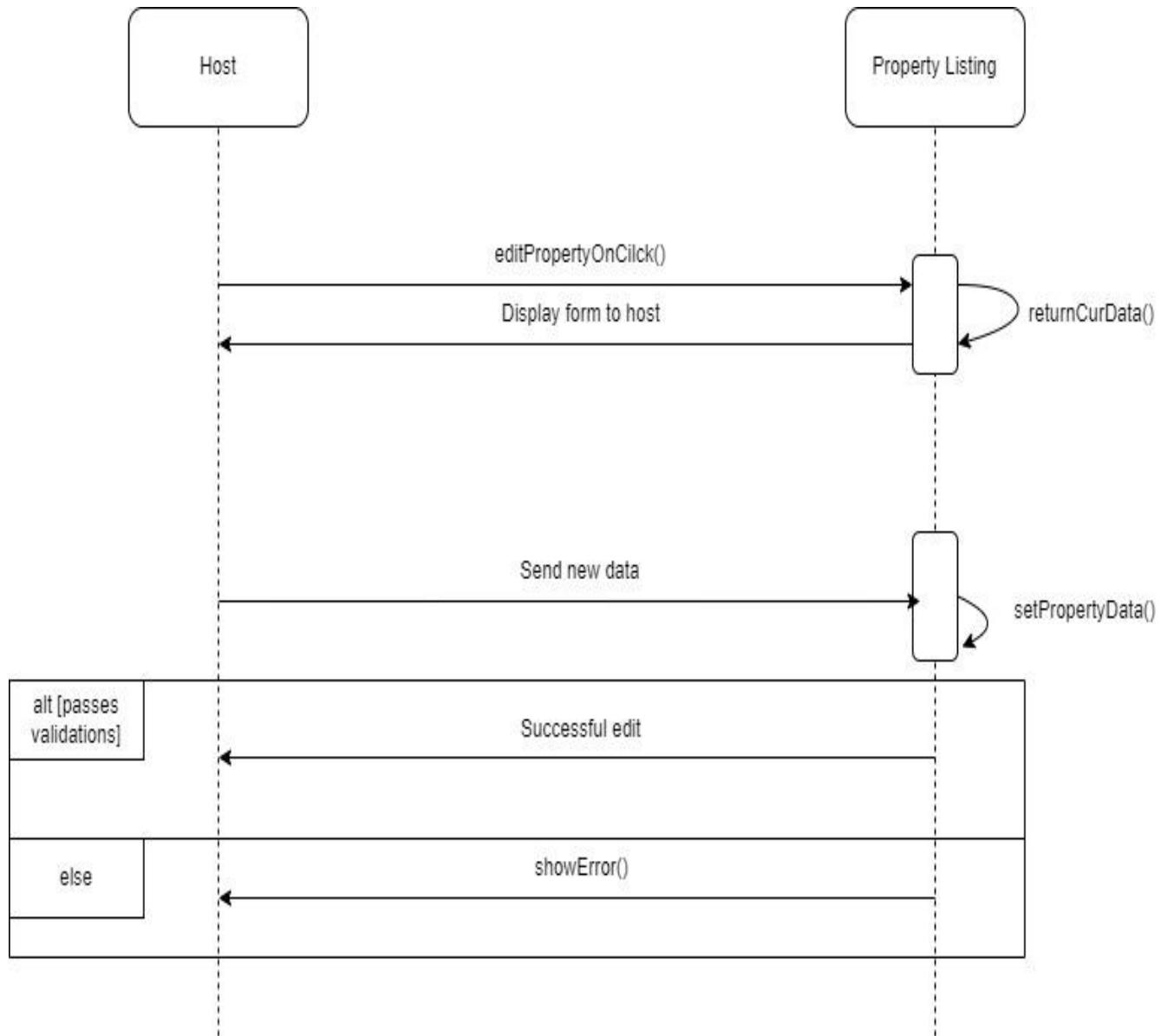
# Property Rentals Management System Requirements Specification

## Delete Property Listing



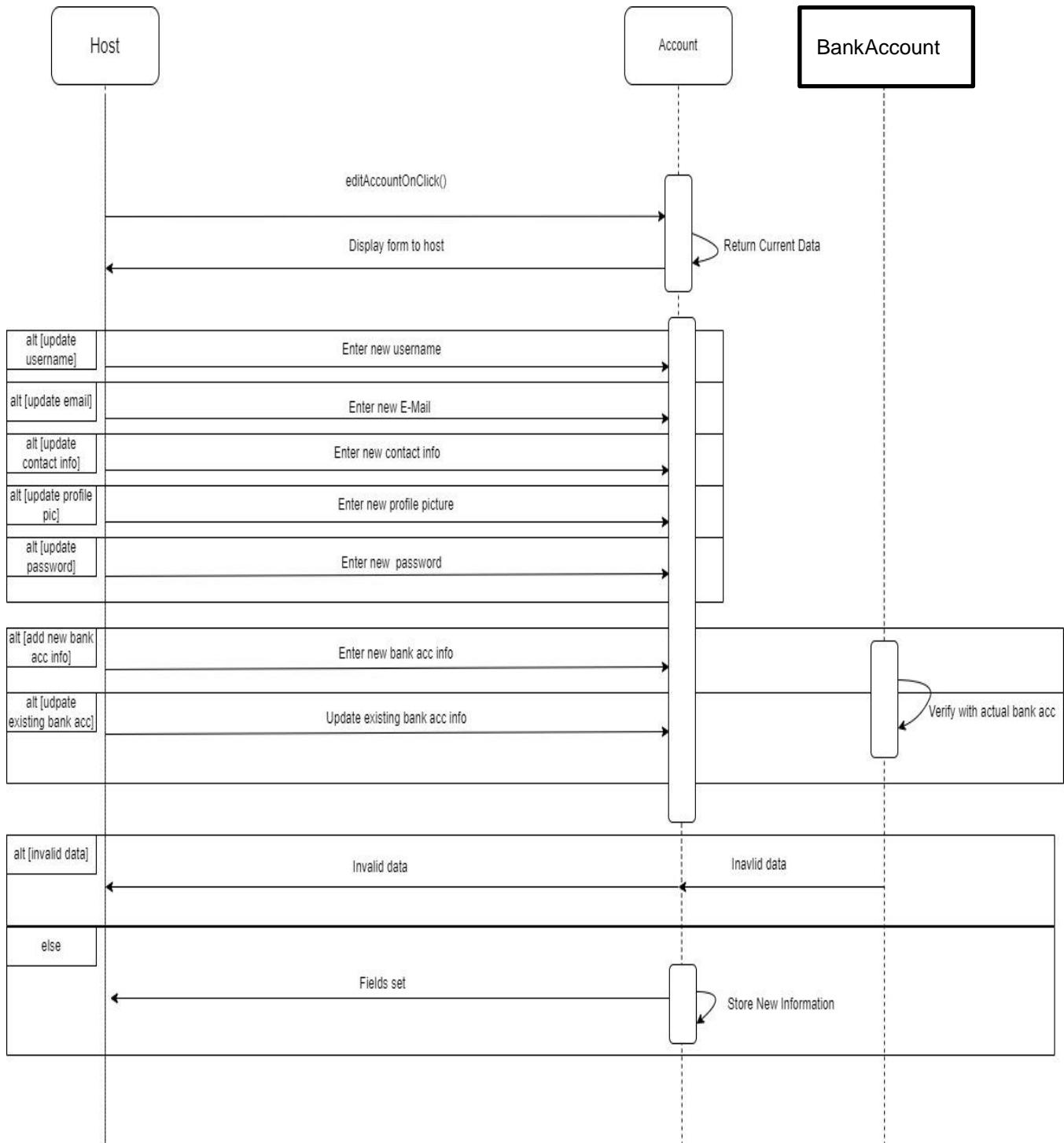
## Property Rentals Management System Requirements Specification

### Edit property

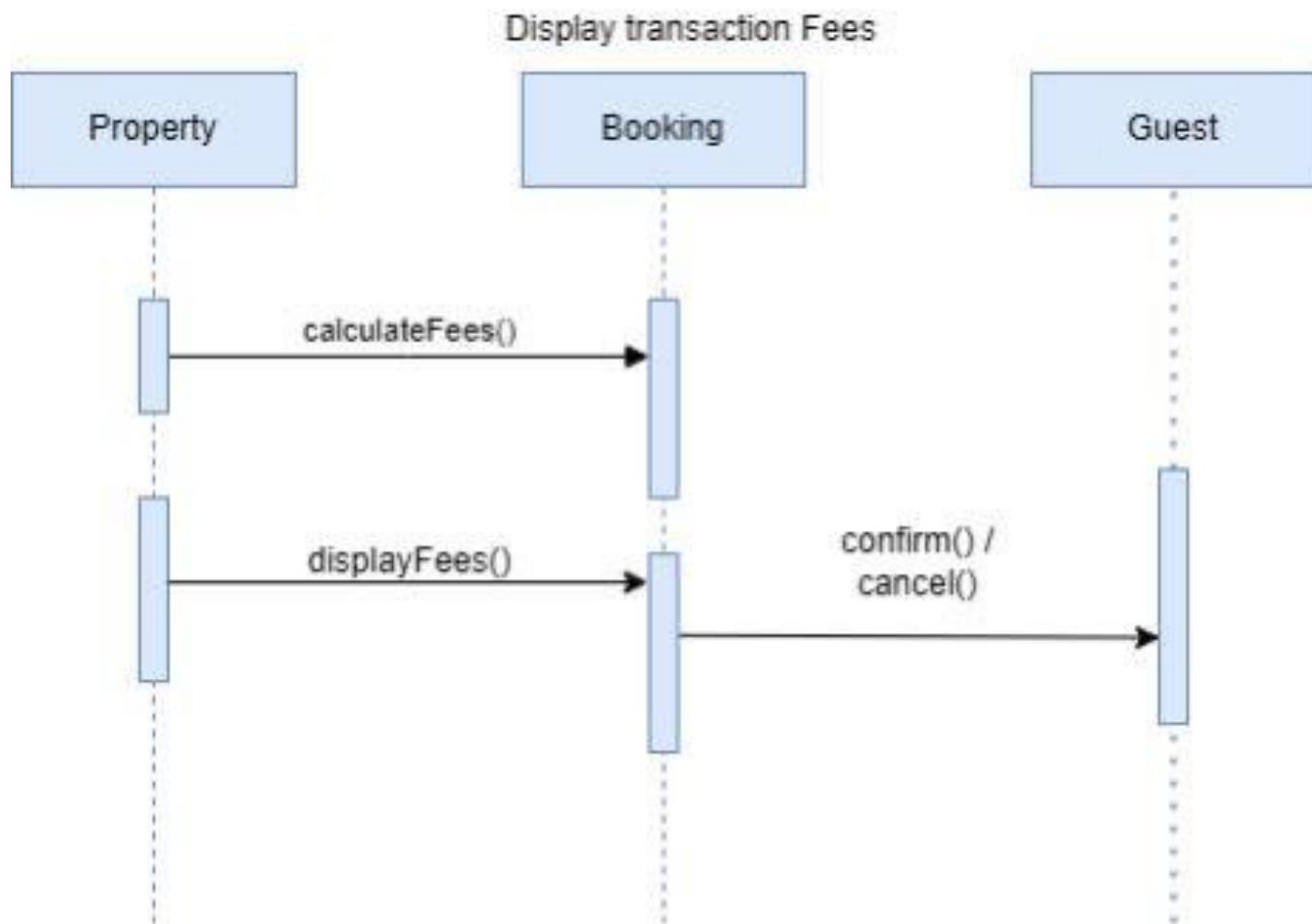


# Property Rentals Management System Requirements Specification

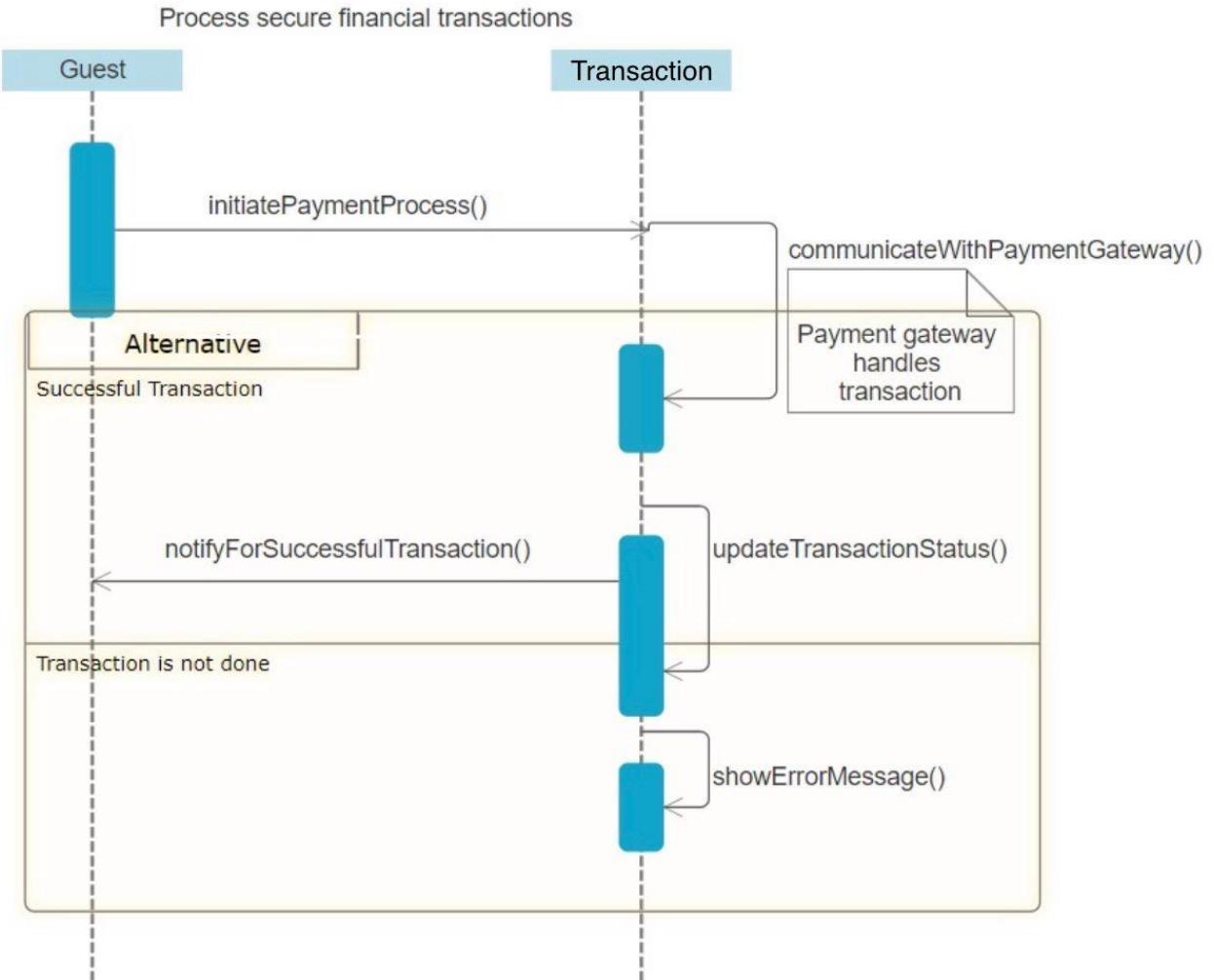
## Update Account Info Host



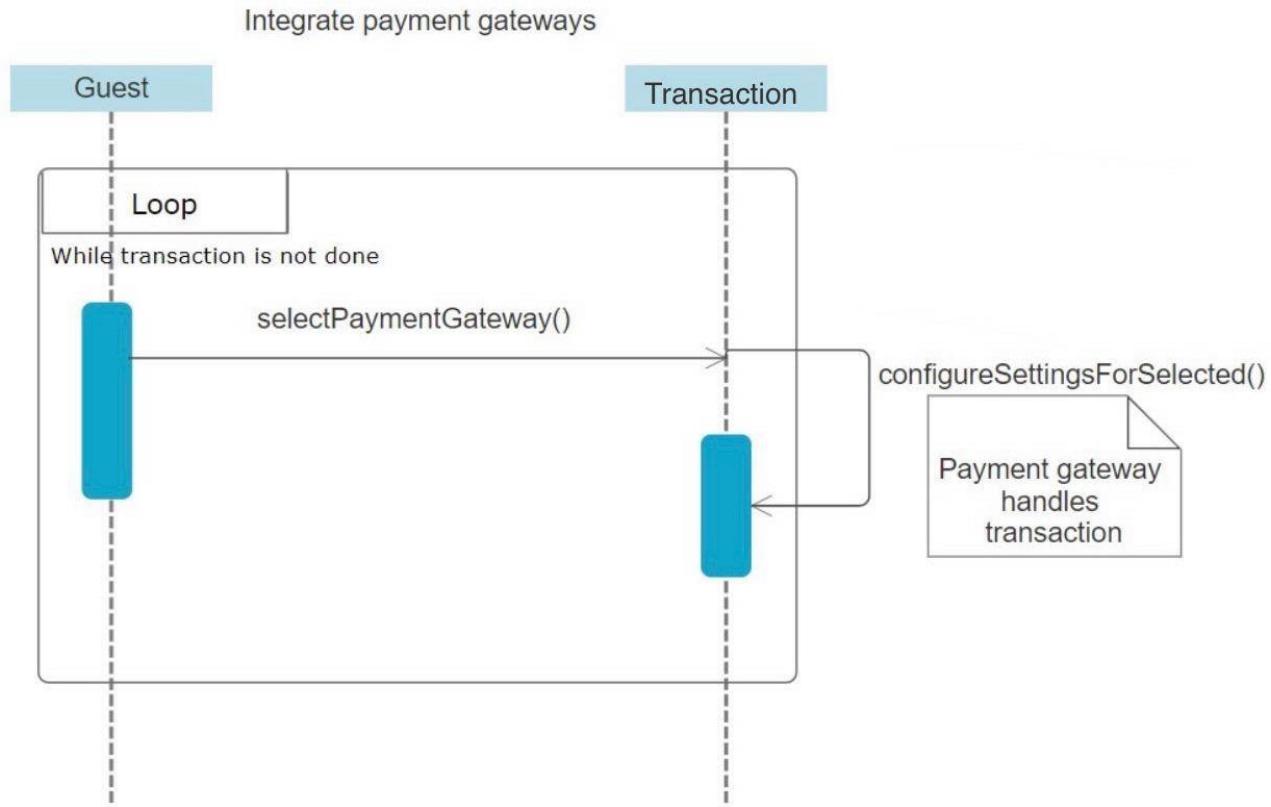
4.



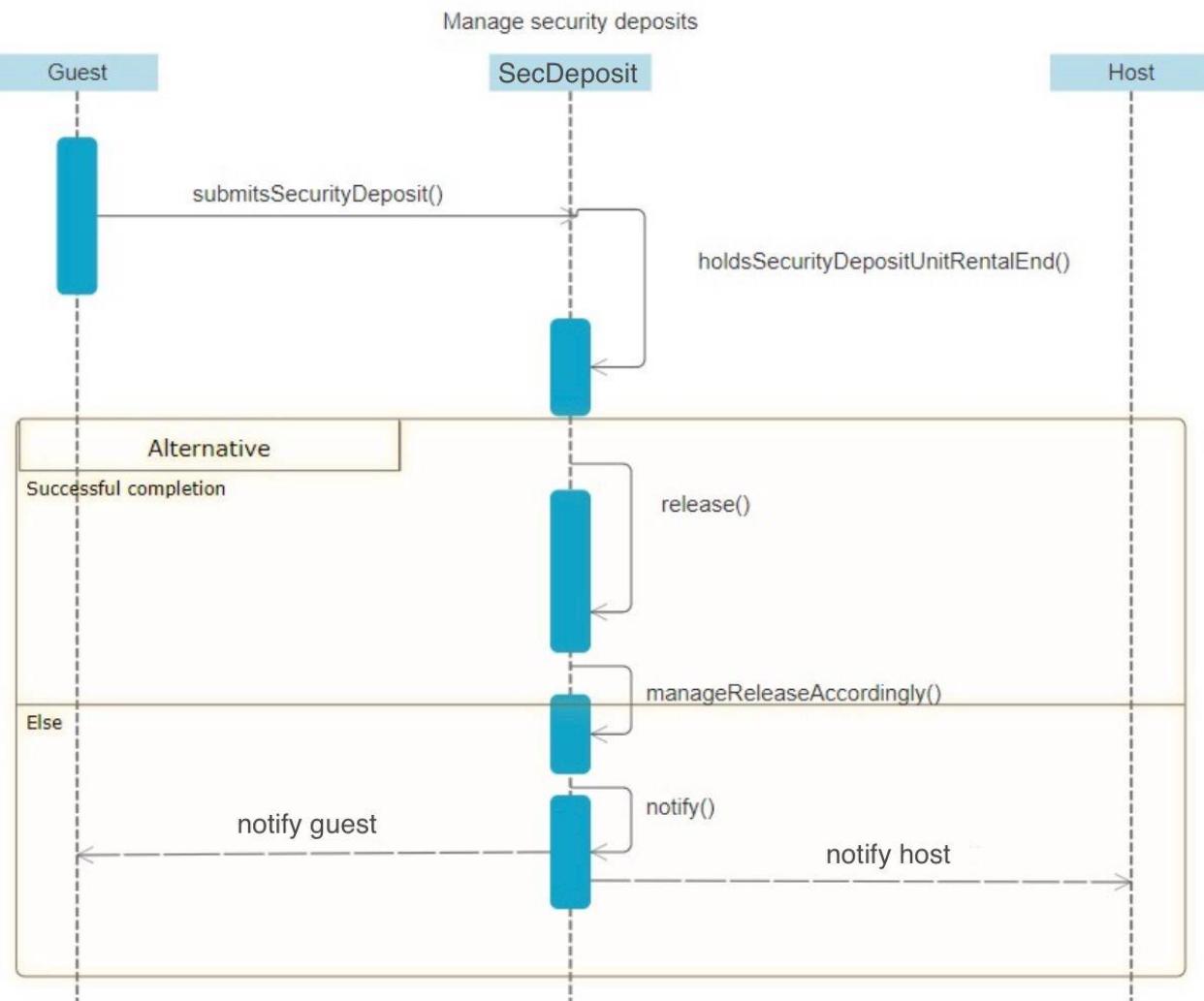
## Property Rentals Management System Requirements Specification



## Property Rentals Management System Requirements Specification

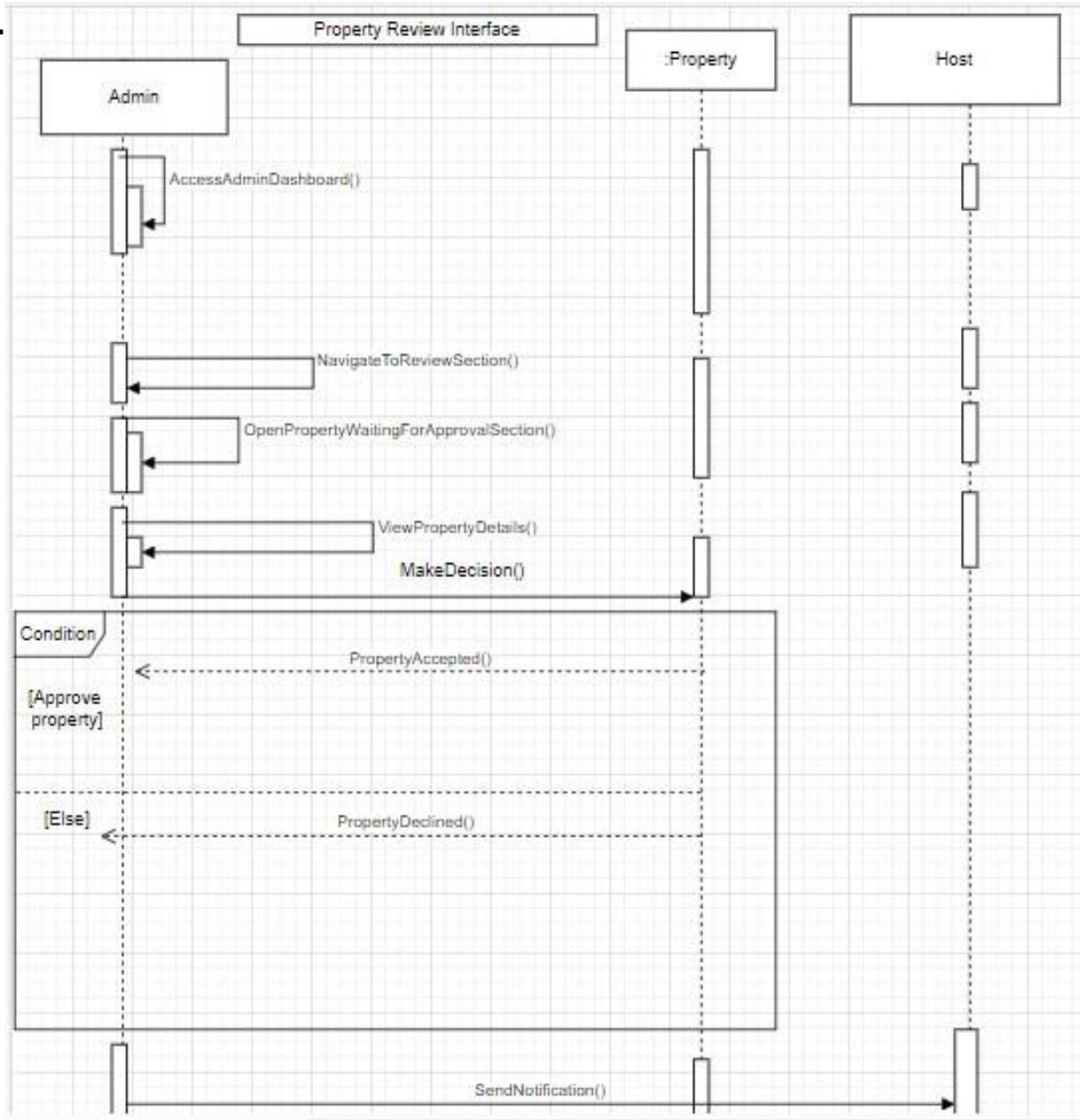


## Property Rentals Management System Requirements Specification

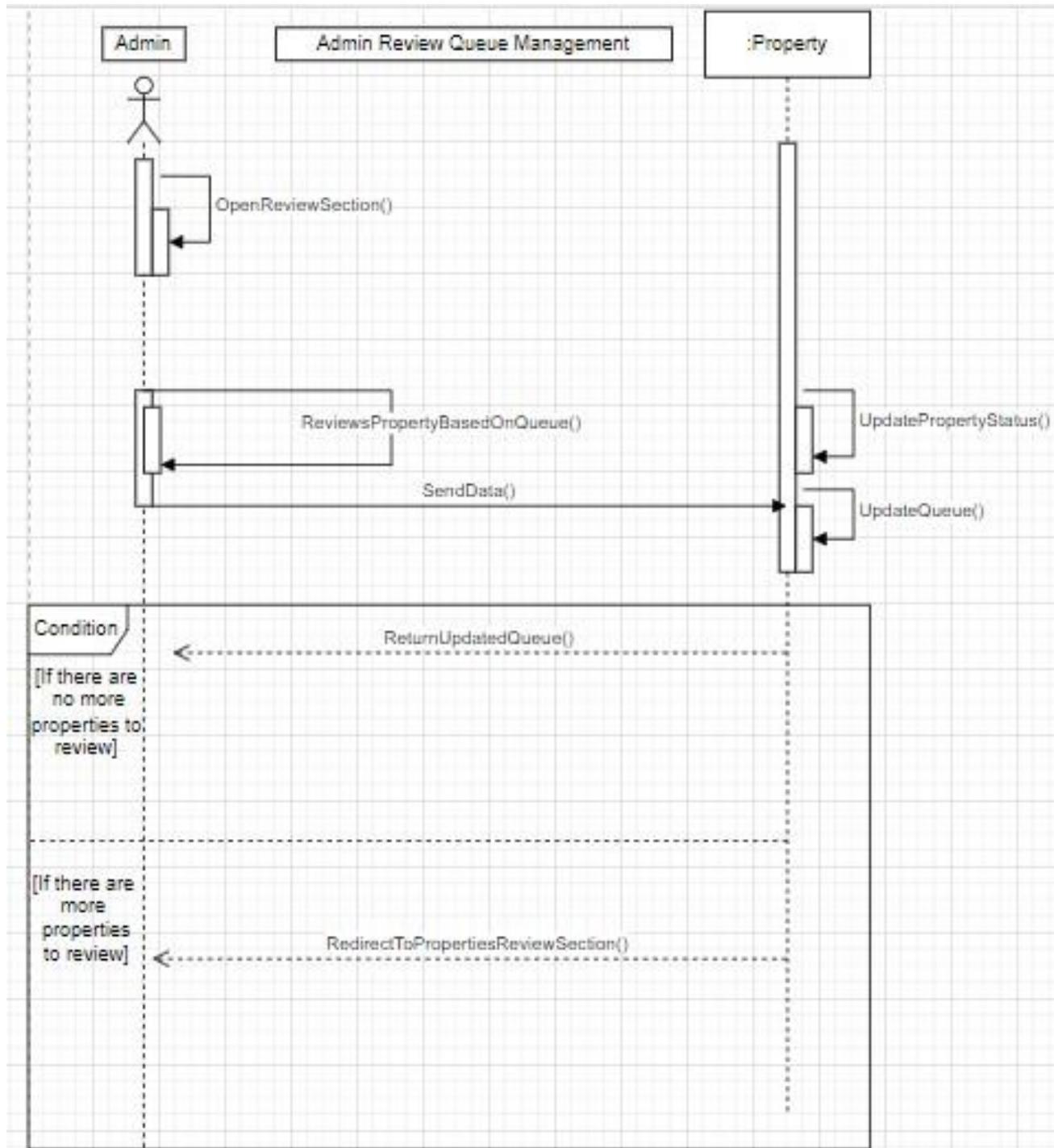


## Property Rentals Management System Requirements Specification

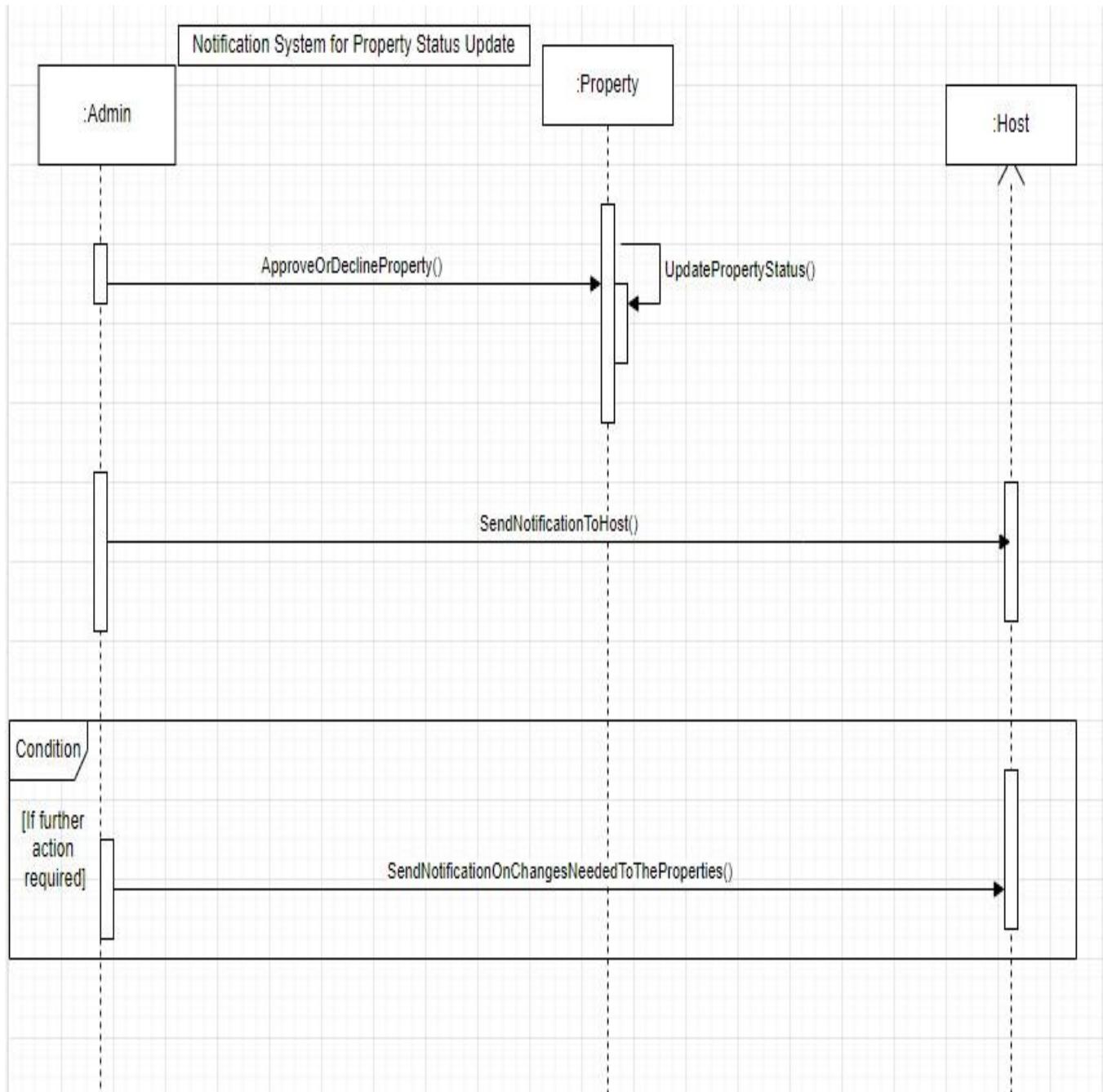
5.



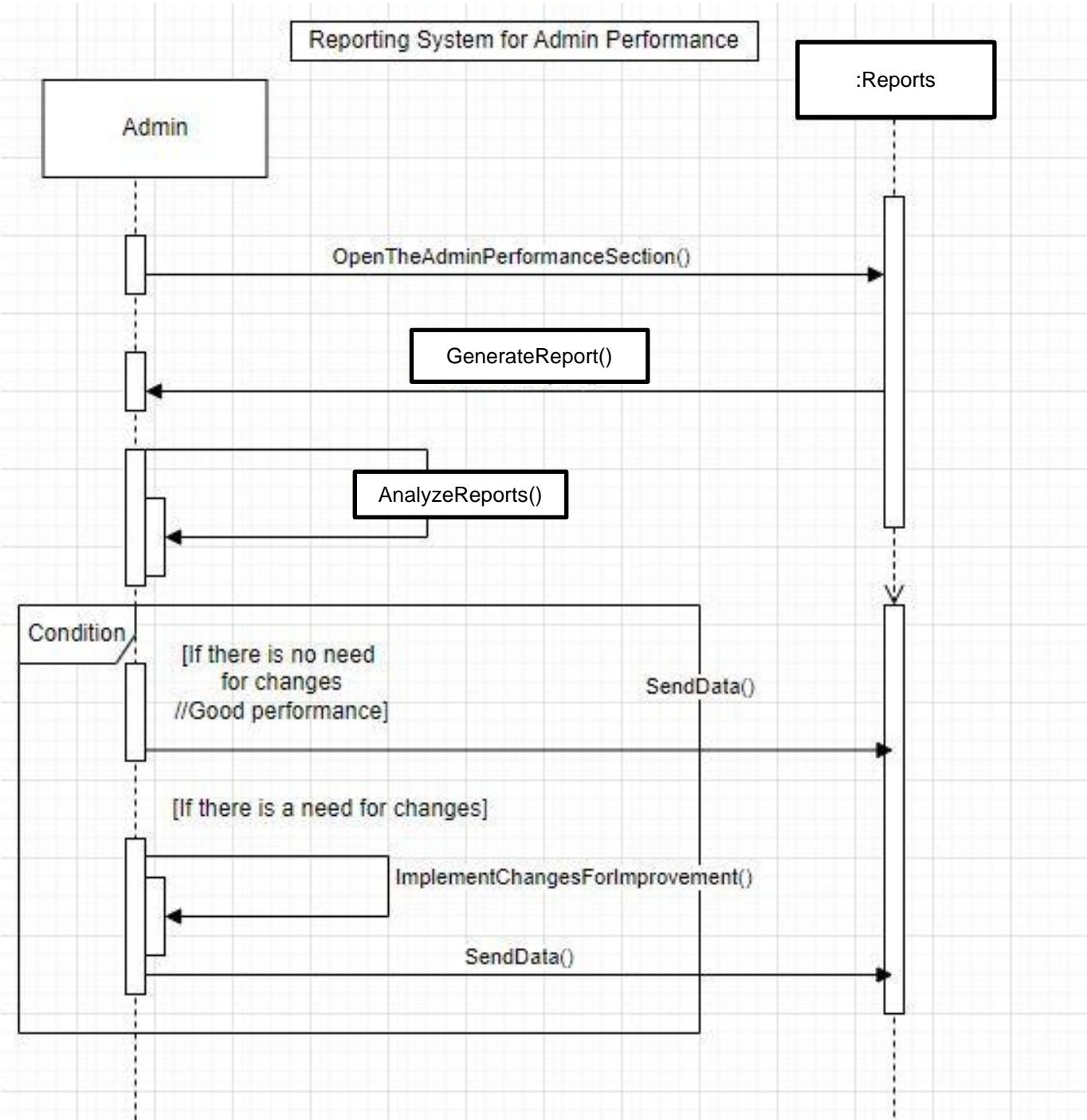
## Property Rentals Management System Requirements Specification



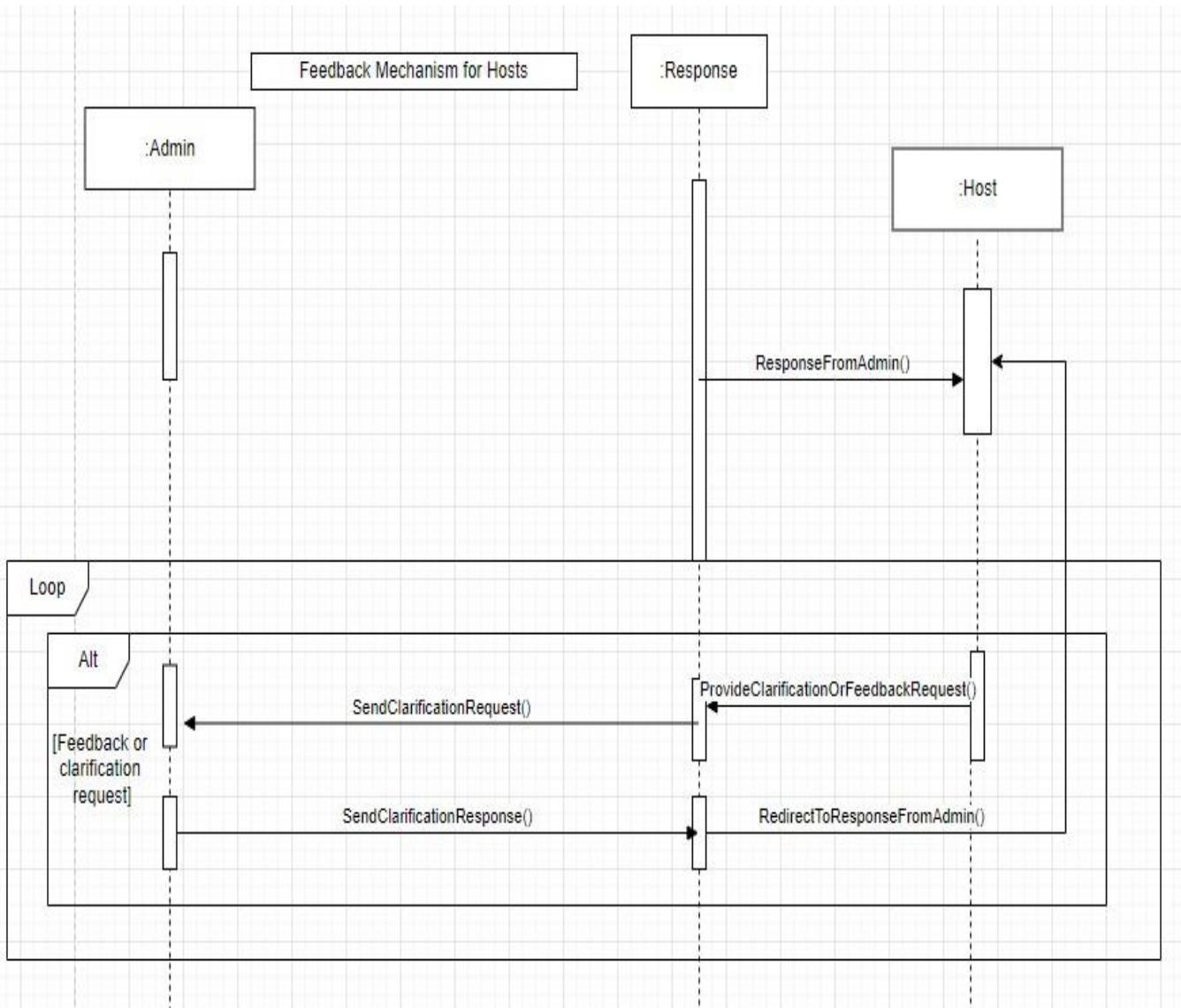
## Property Rentals Management System Requirements Specification



## Property Rentals Management System Requirements Specification

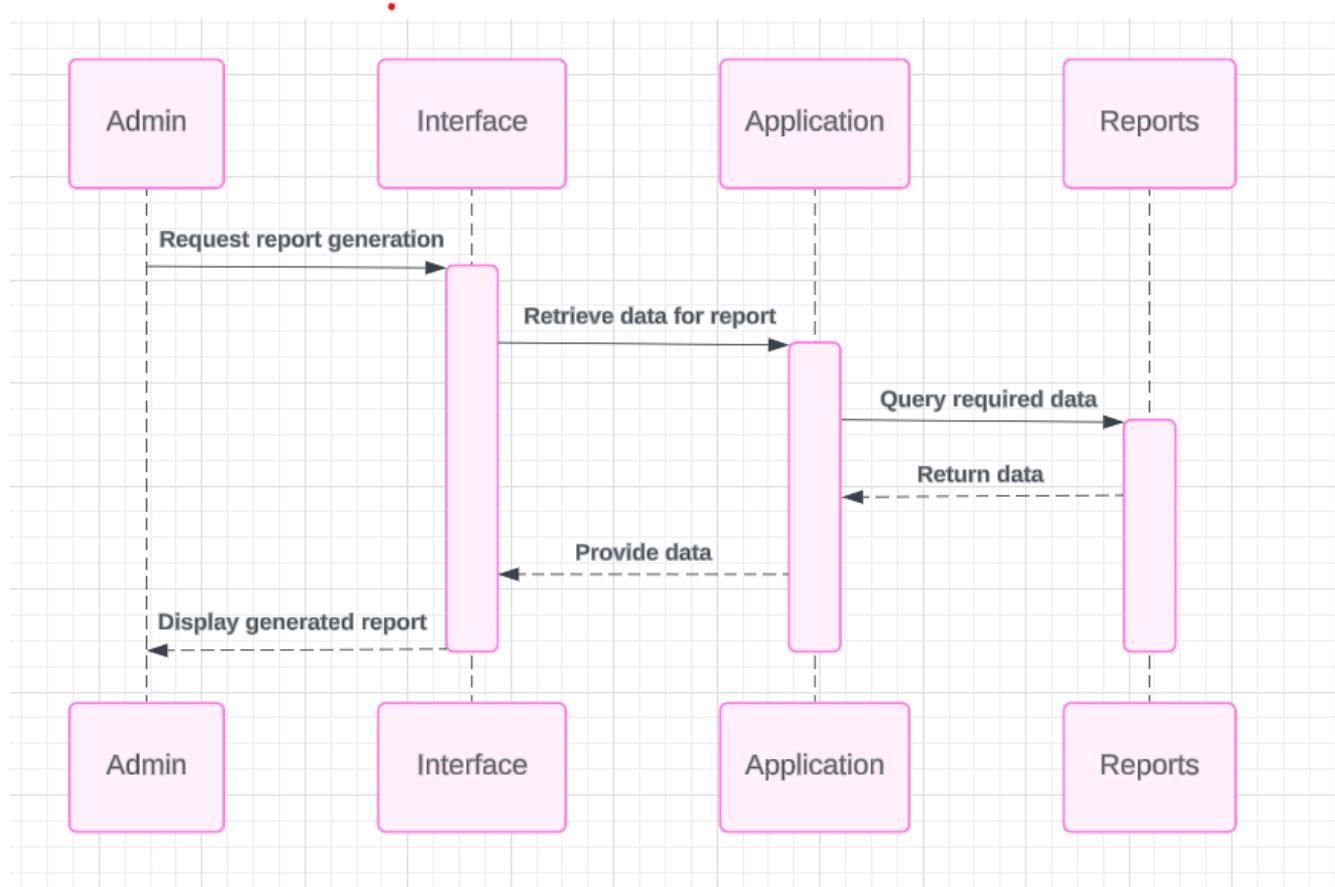


## Property Rentals Management System Requirements Specification

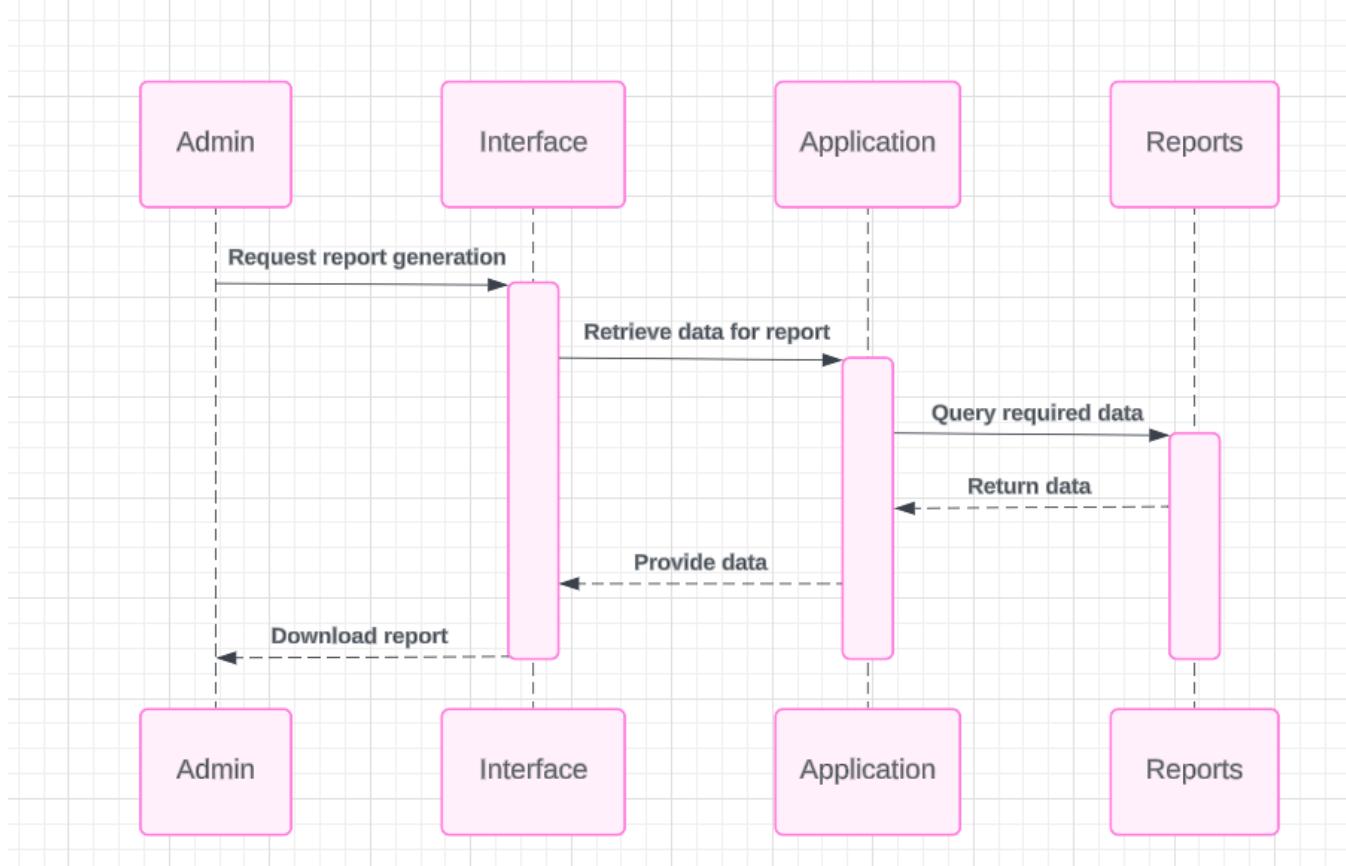


## Property Rentals Management System Requirements Specification

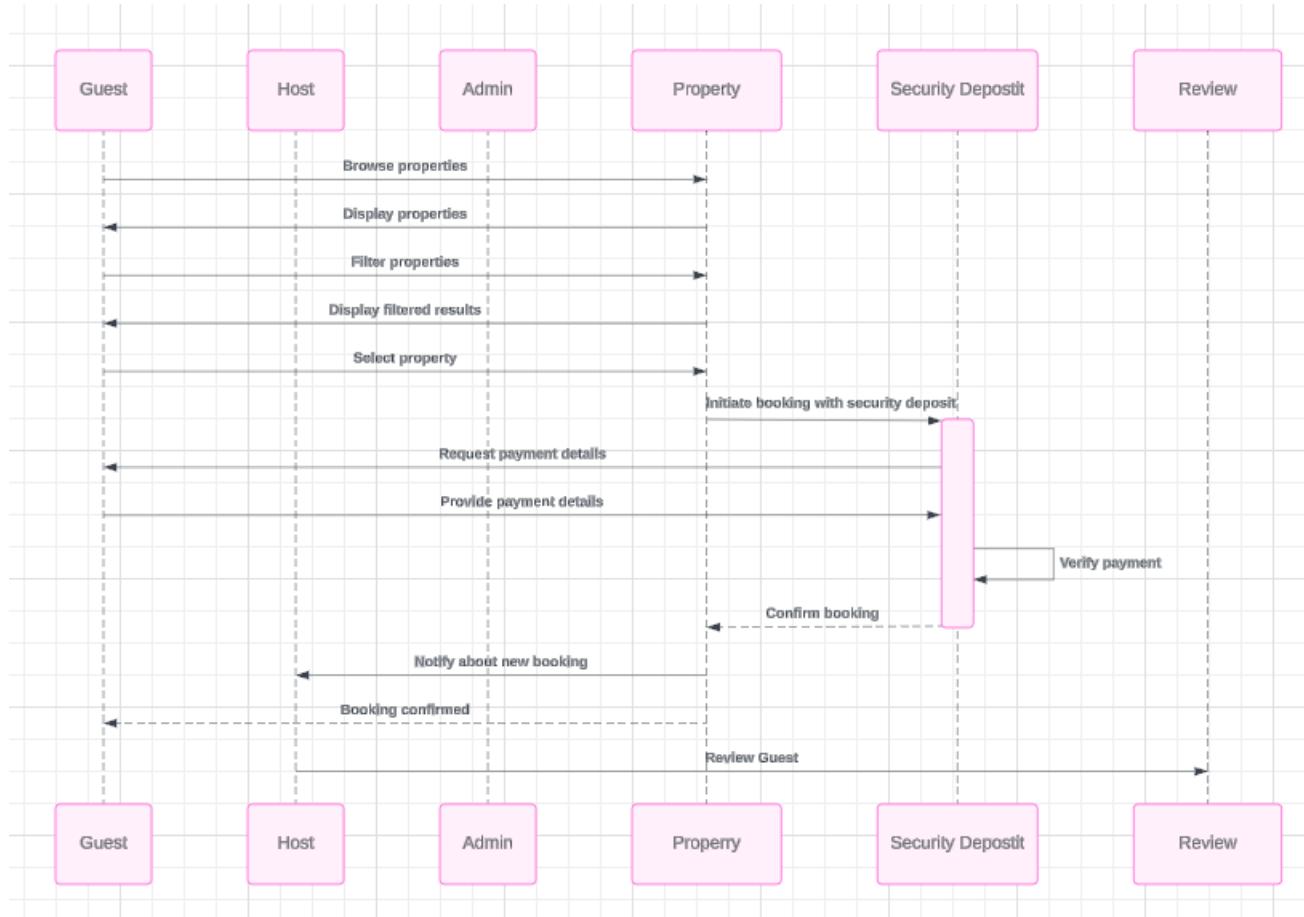
6.



## Property Rentals Management System Requirements Specification



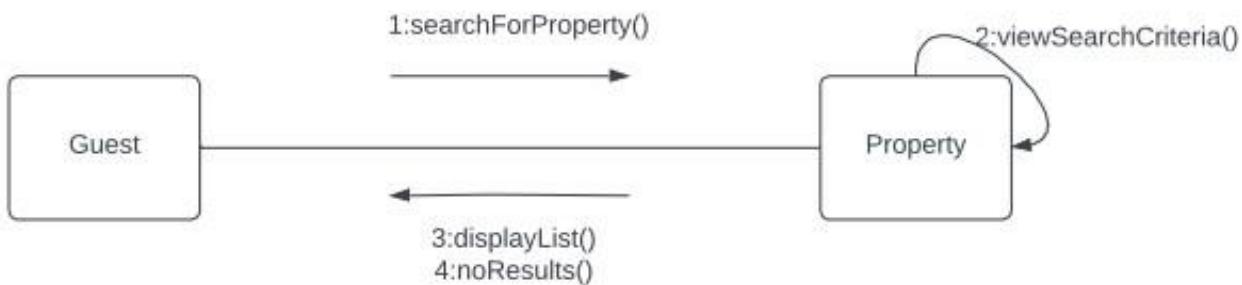
## Property Rentals Management System Requirements Specification



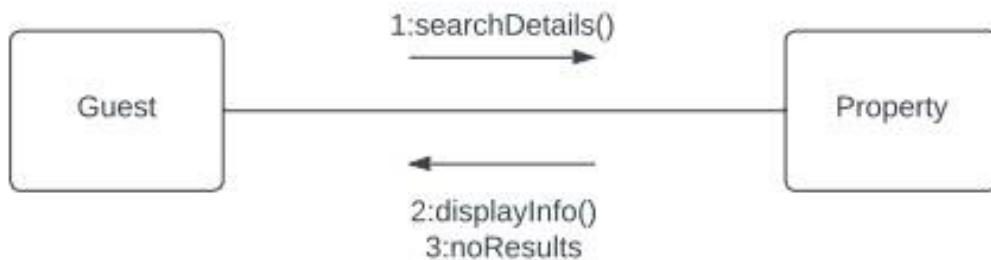
## COLLABORATION DIAGRAMS

1.

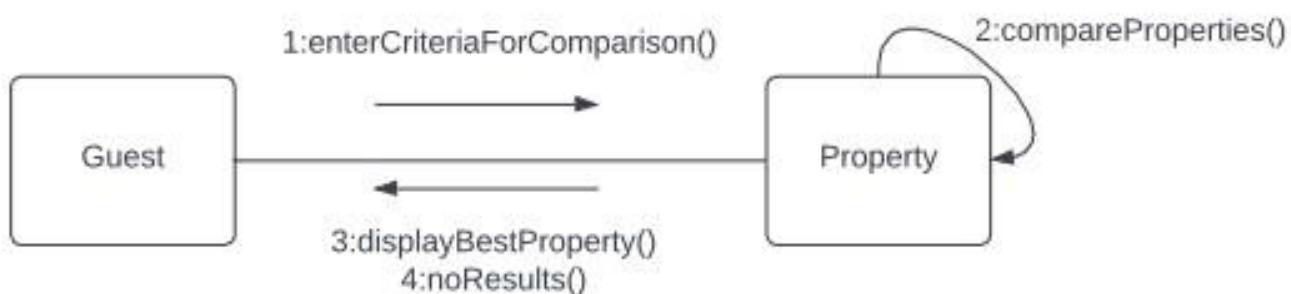
### Search Property



### View Property Details



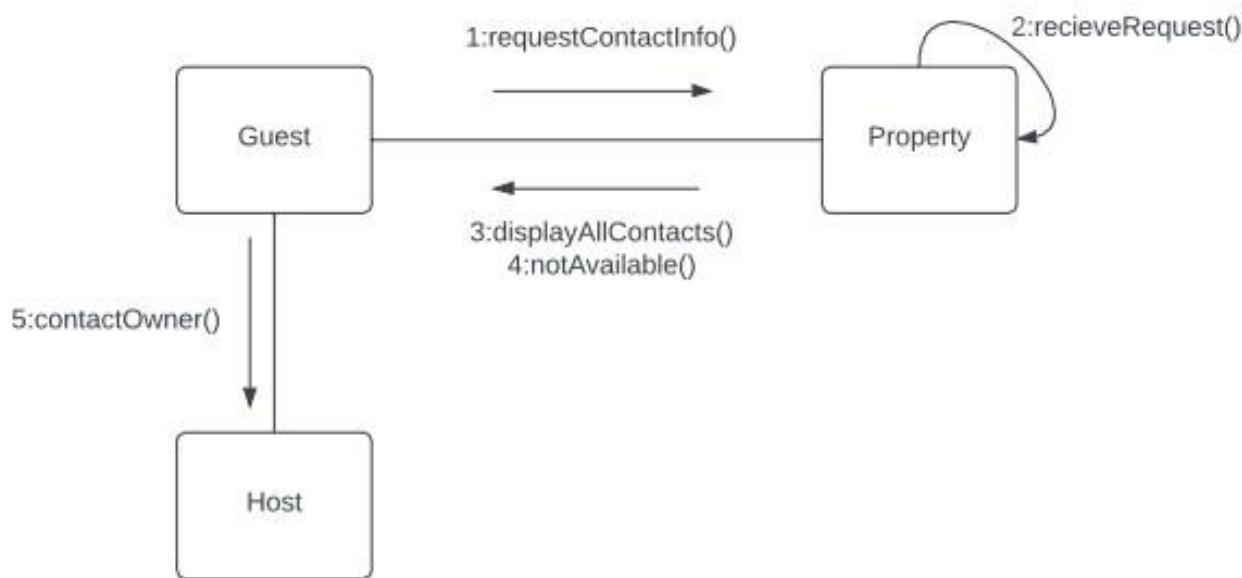
### Comparison



## Save To Favourites



## Contact Property Owner

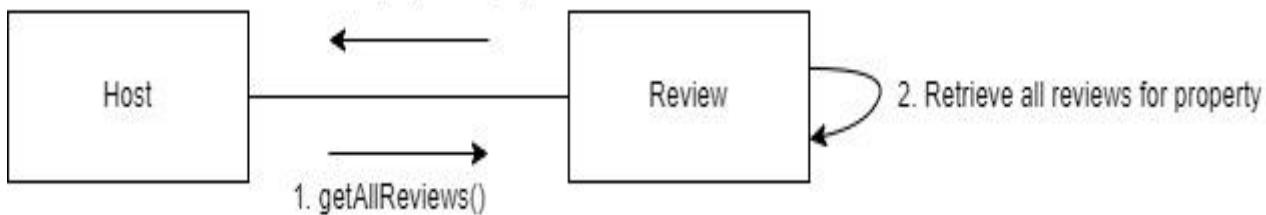


## Property Rentals Management System Requirements Specification

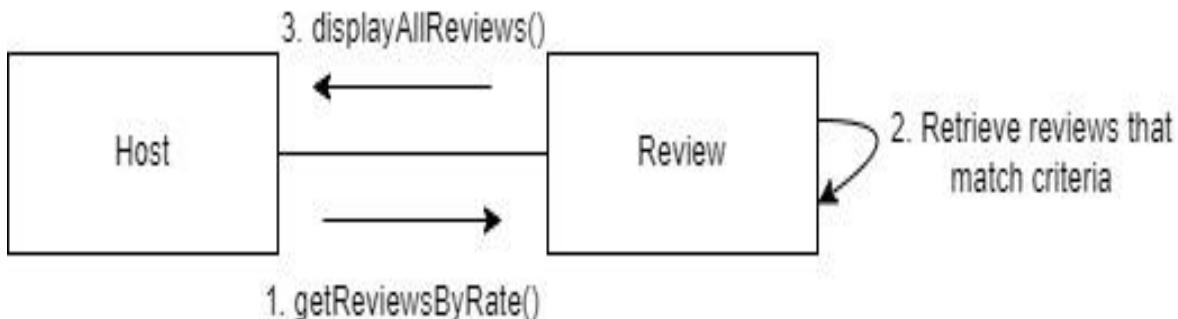
2.

Browse reviews

[there are reviews]  
3.1 displayAllReviews()  
[else]  
3.2 displayMessage()

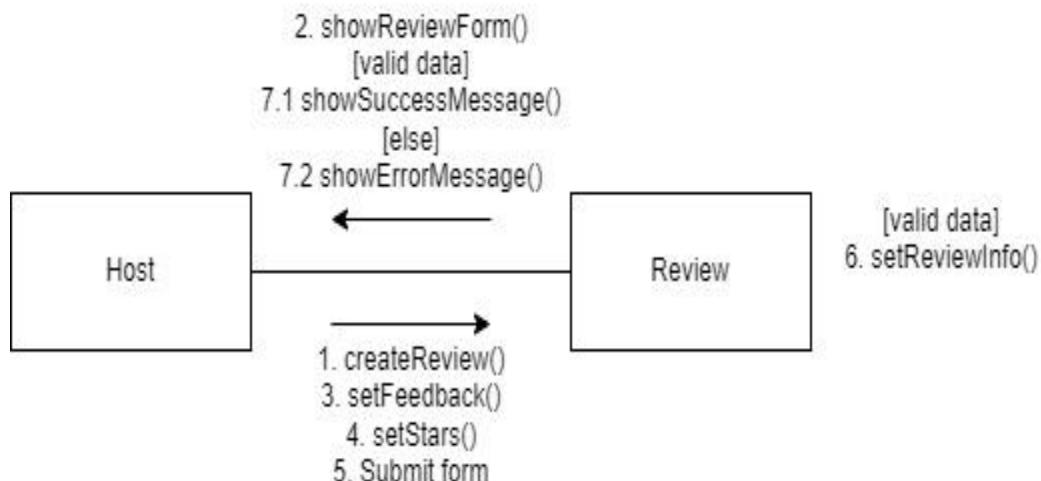


Filter Reviews By Rating



## Property Rentals Management System Requirements Specification

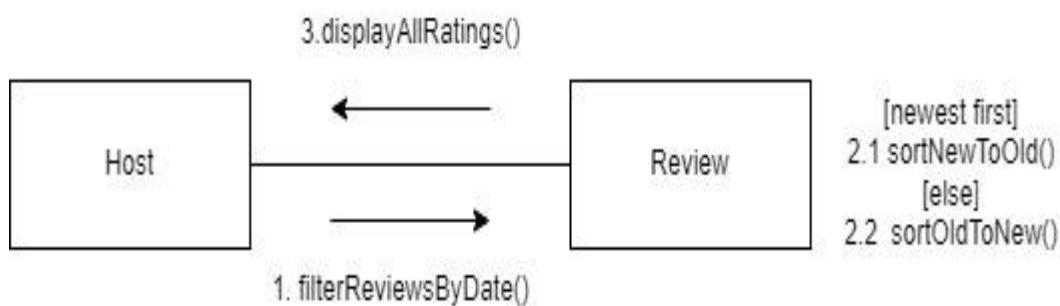
Rate and Review Property



Report Inappropriate Reviews

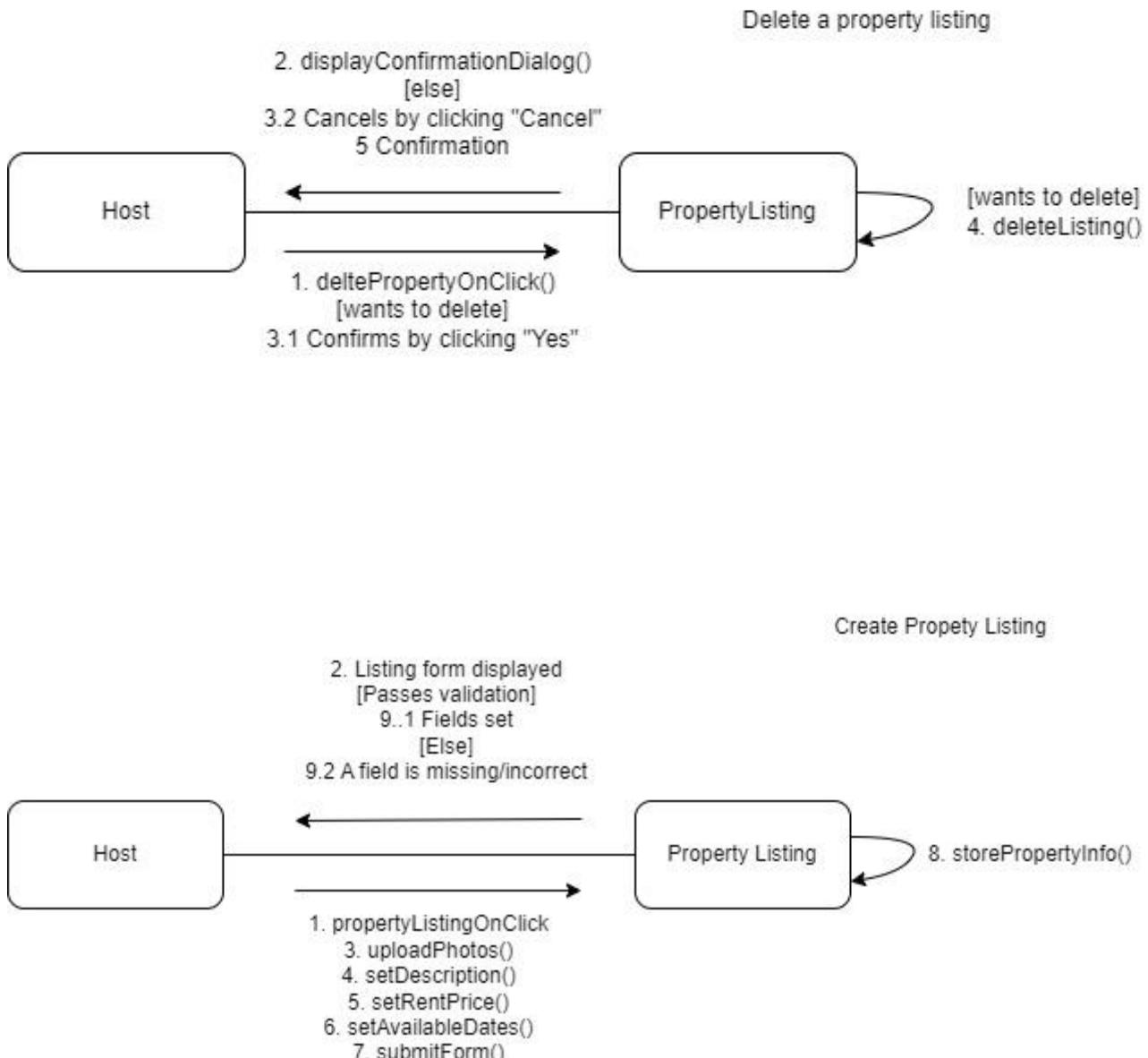


Filter Reviews By Date

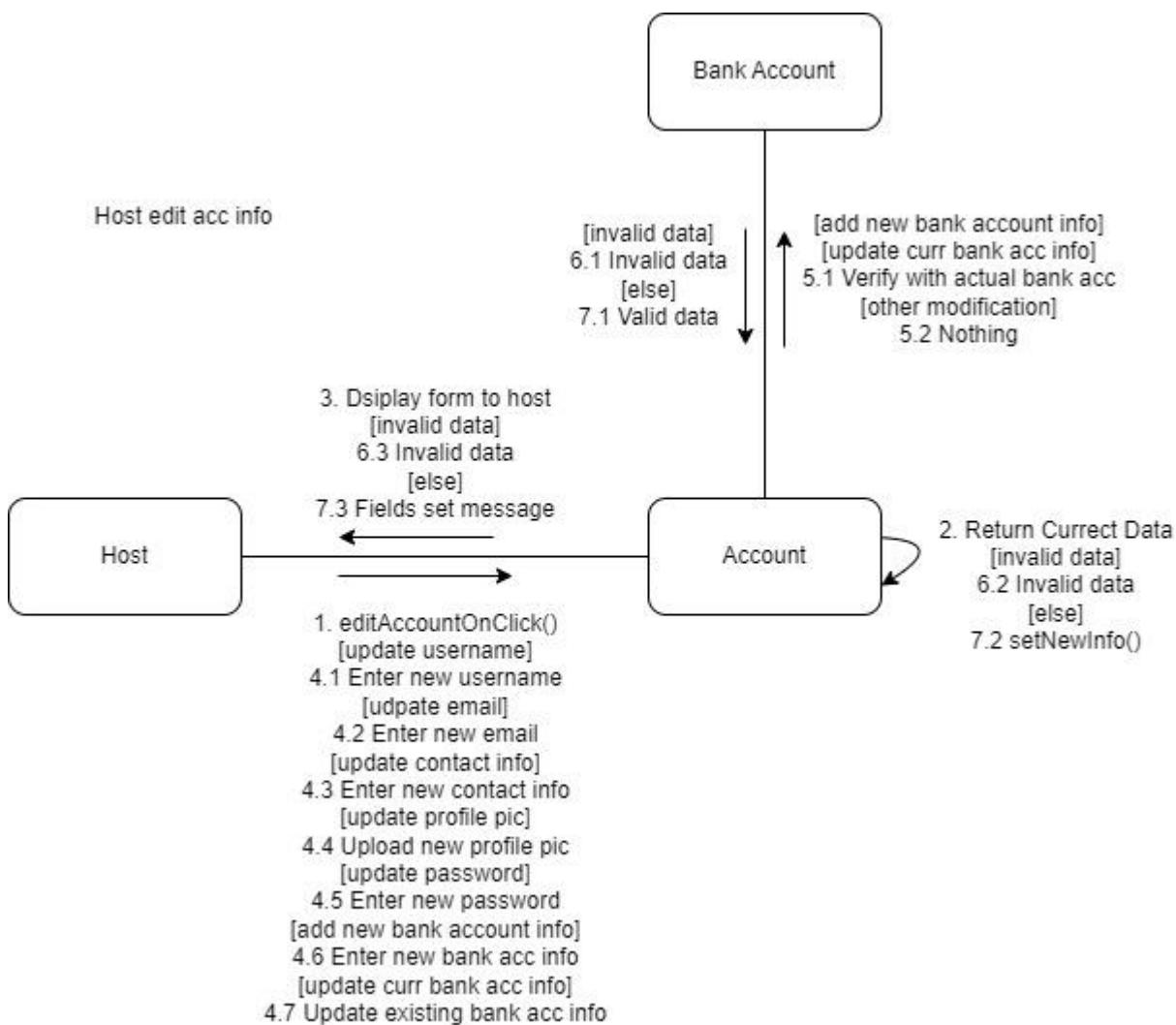
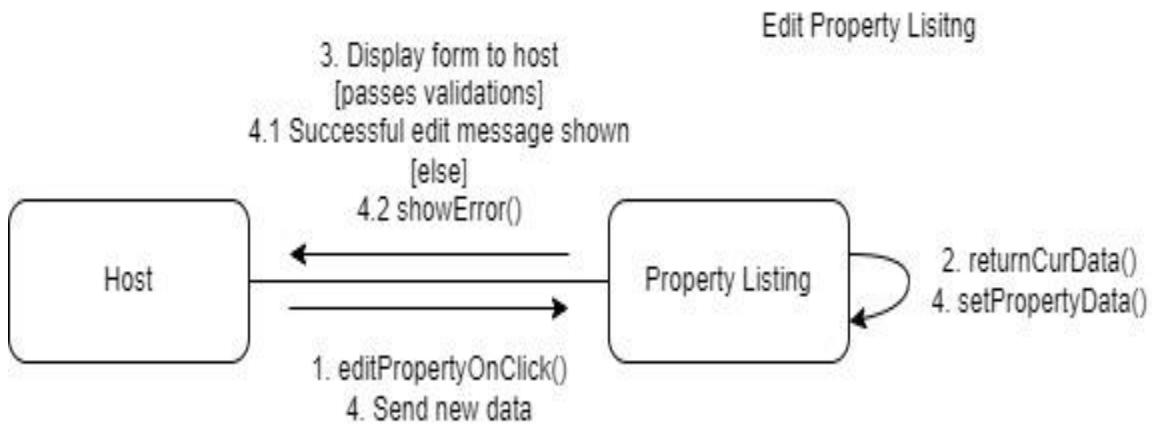


## Property Rentals Management System Requirements Specification

**3.**

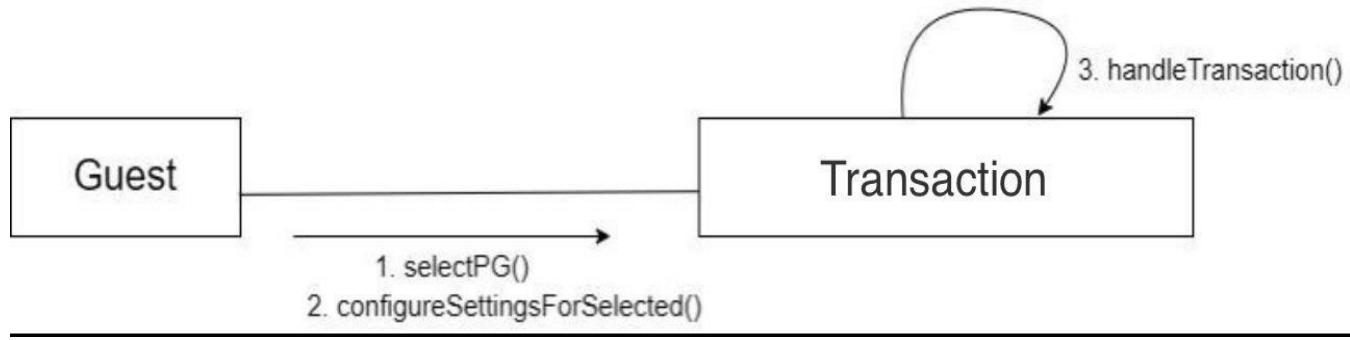


## Property Rentals Management System Requirements Specification



4.

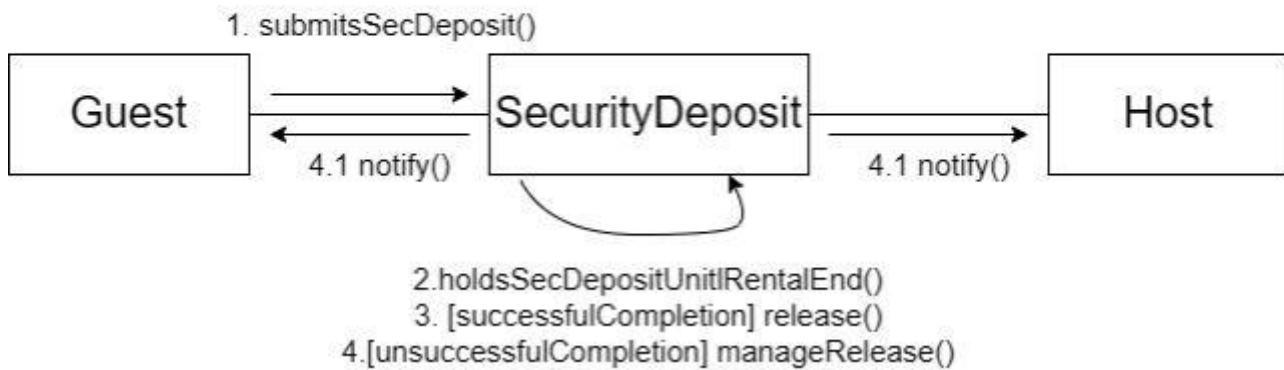
Integrate payment gateways



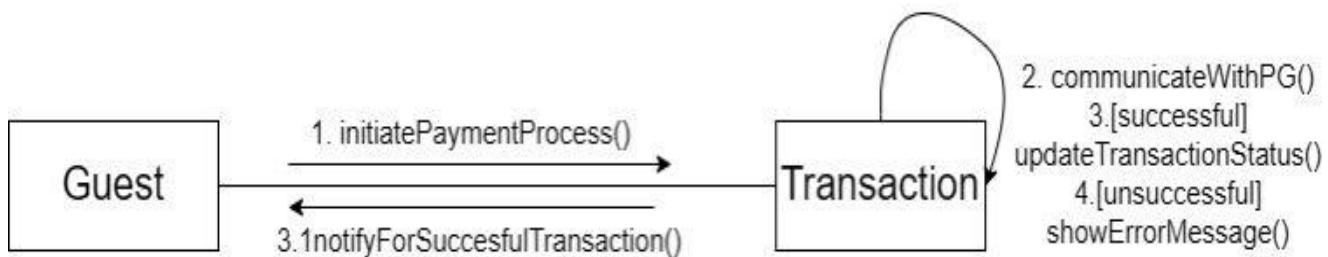
Display transaction Fees



## Manage Security Deposits

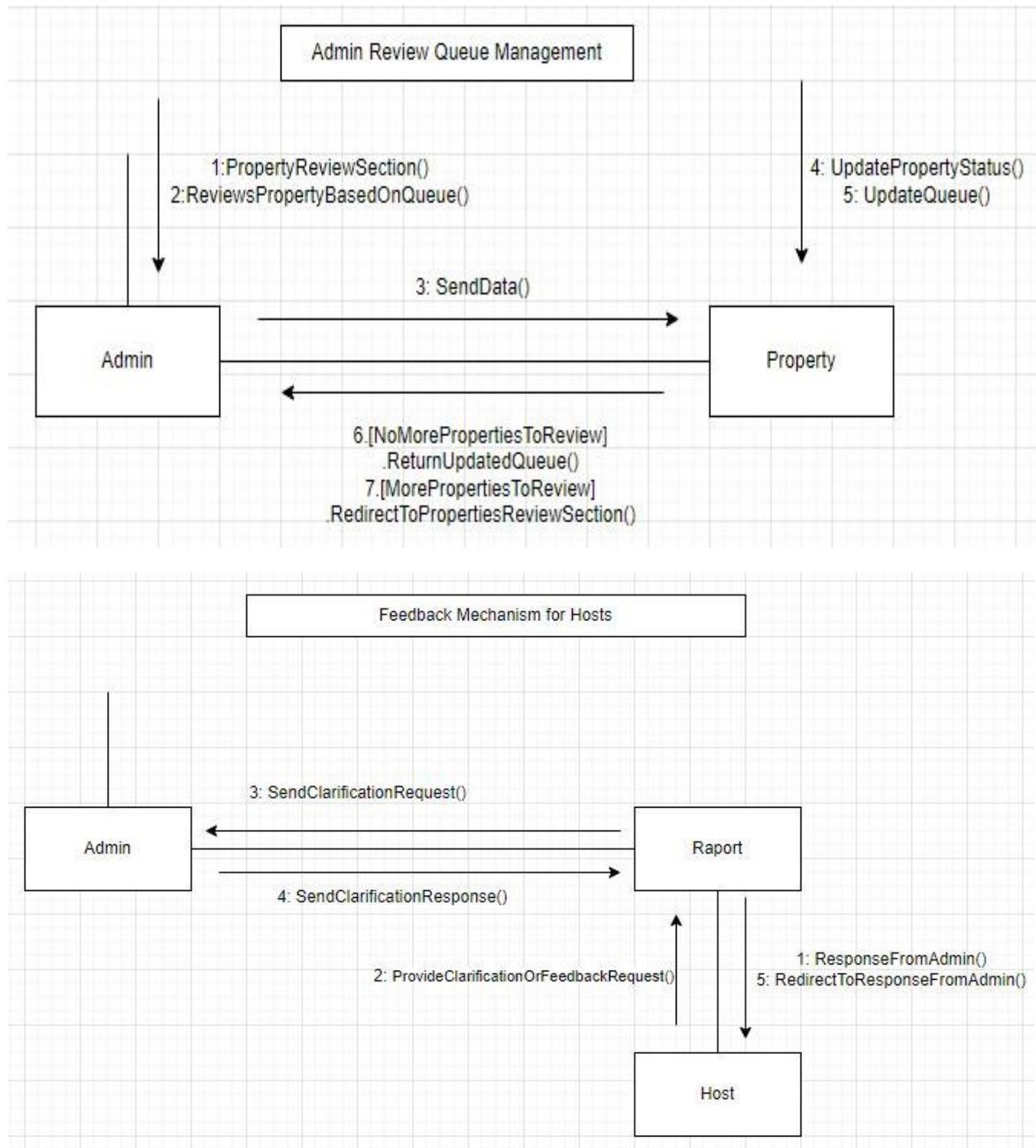


## Process secure financial transactions

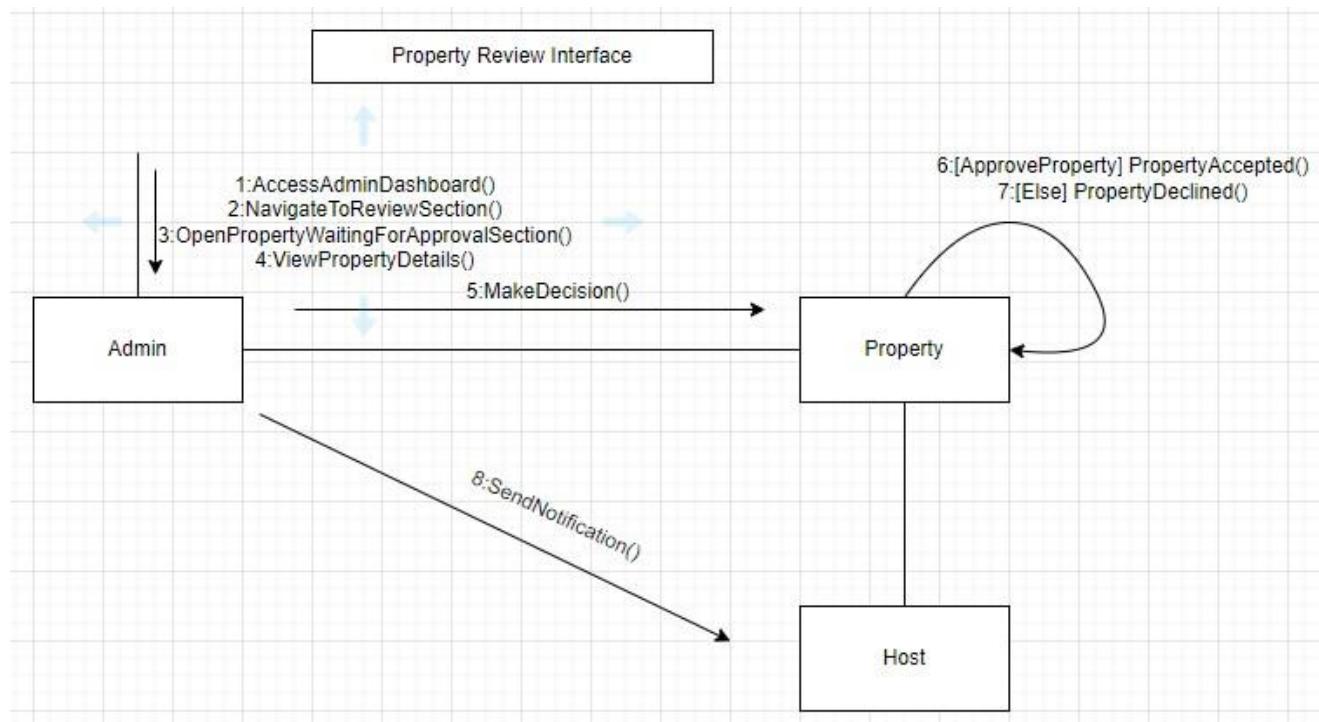
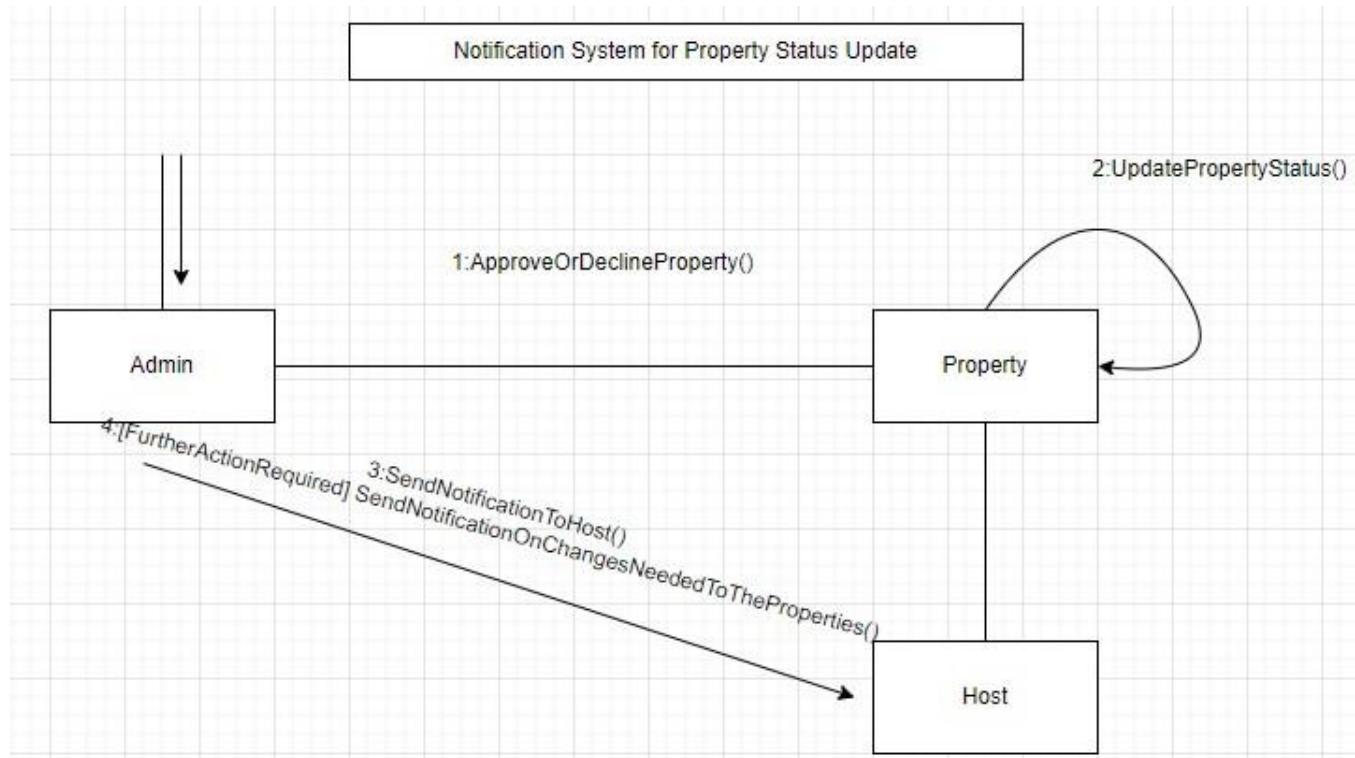


## Property Rentals Management System Requirements Specification

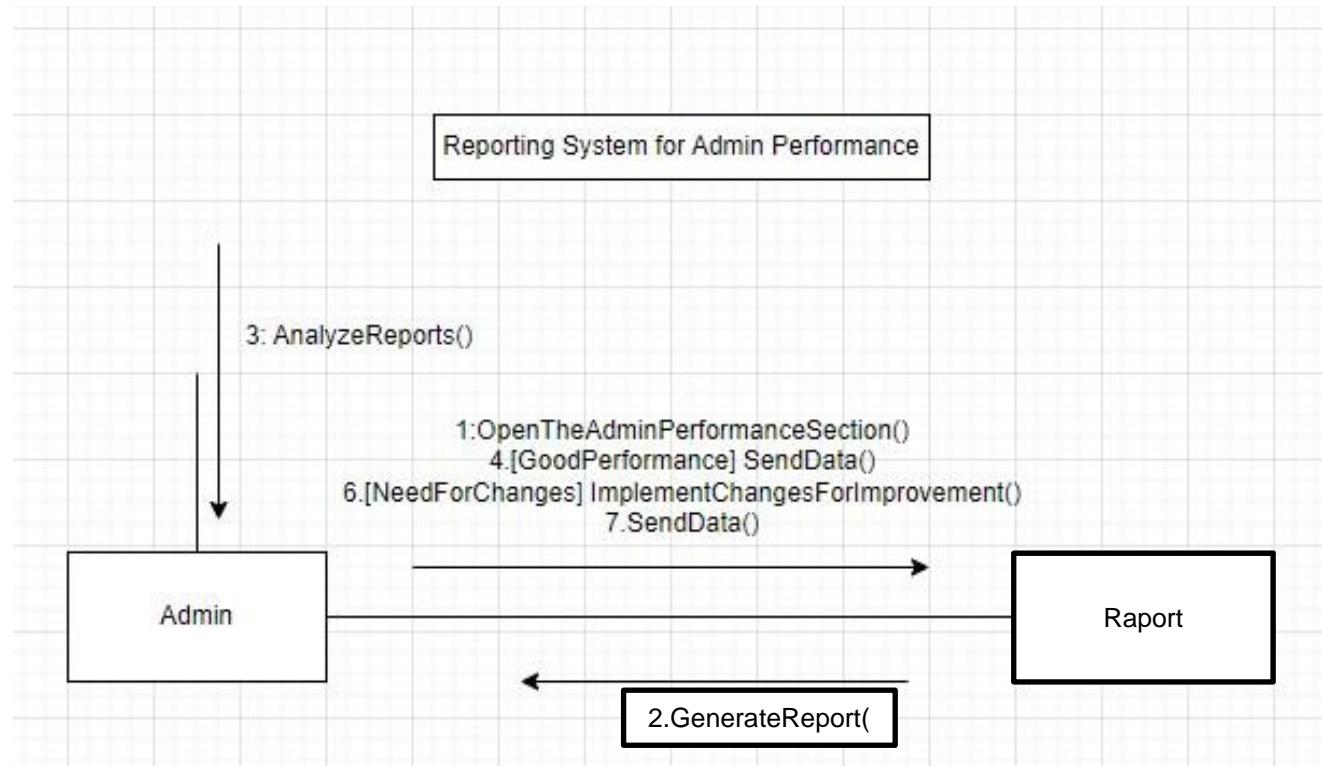
**5.**



## Property Rentals Management System Requirements Specification

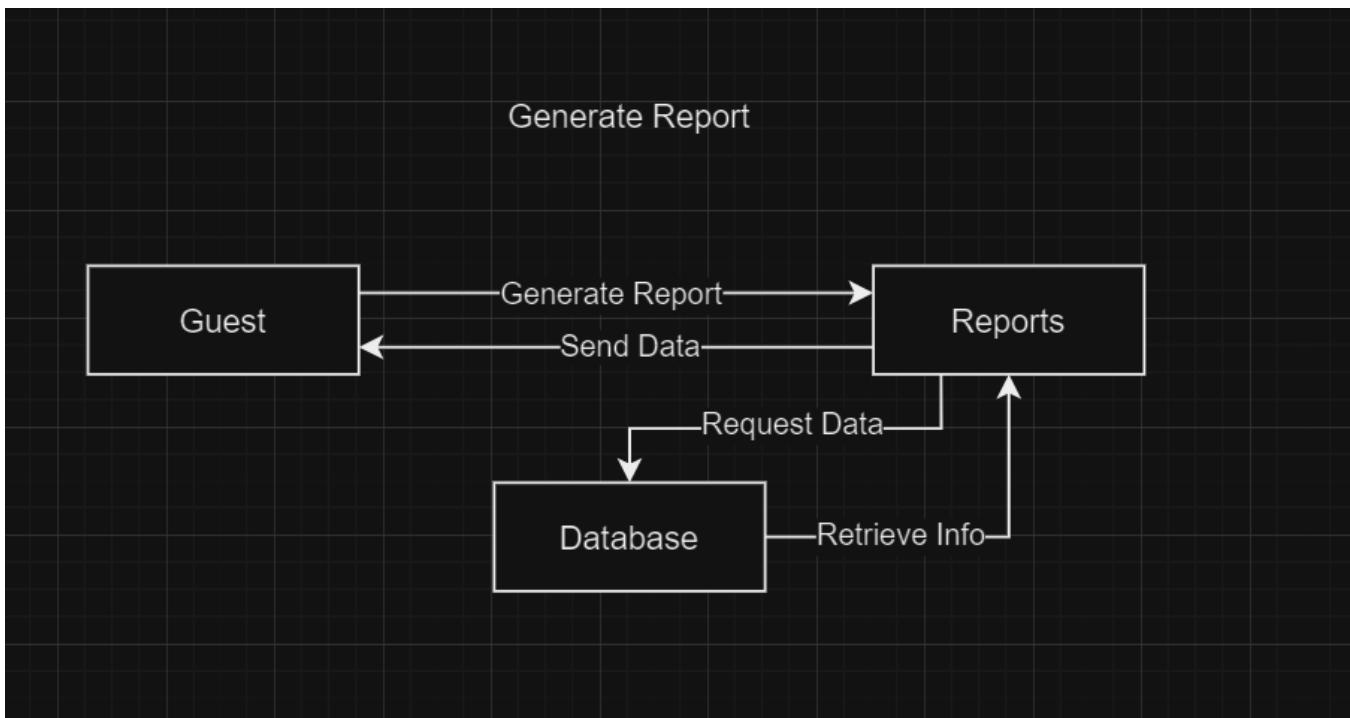
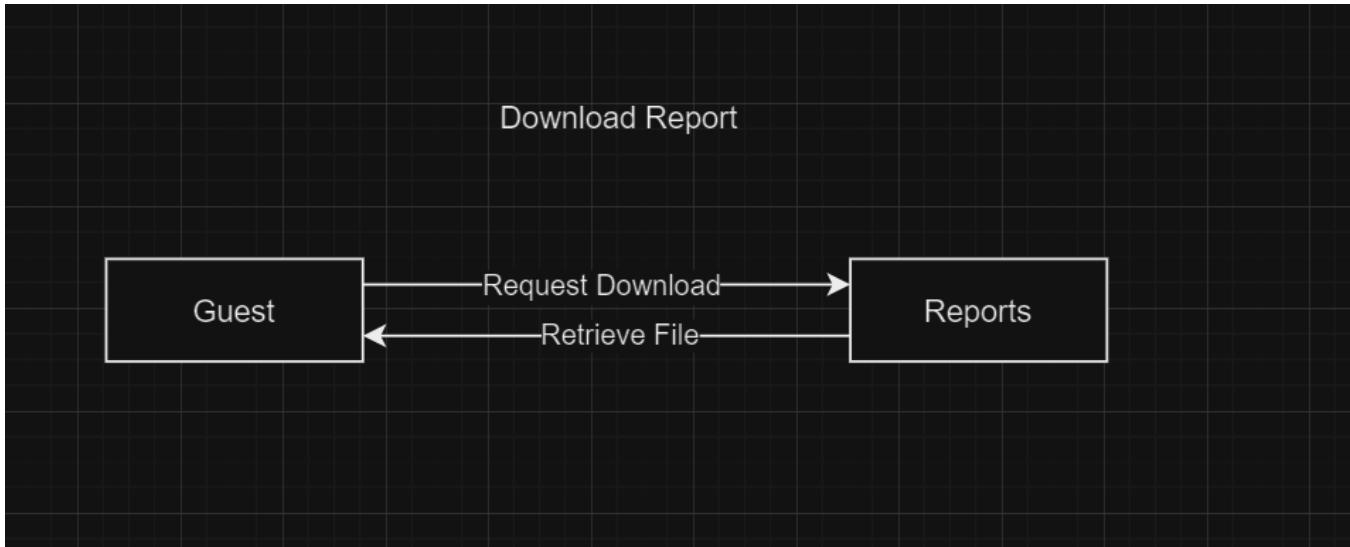


## Property Rentals Management System Requirements Specification

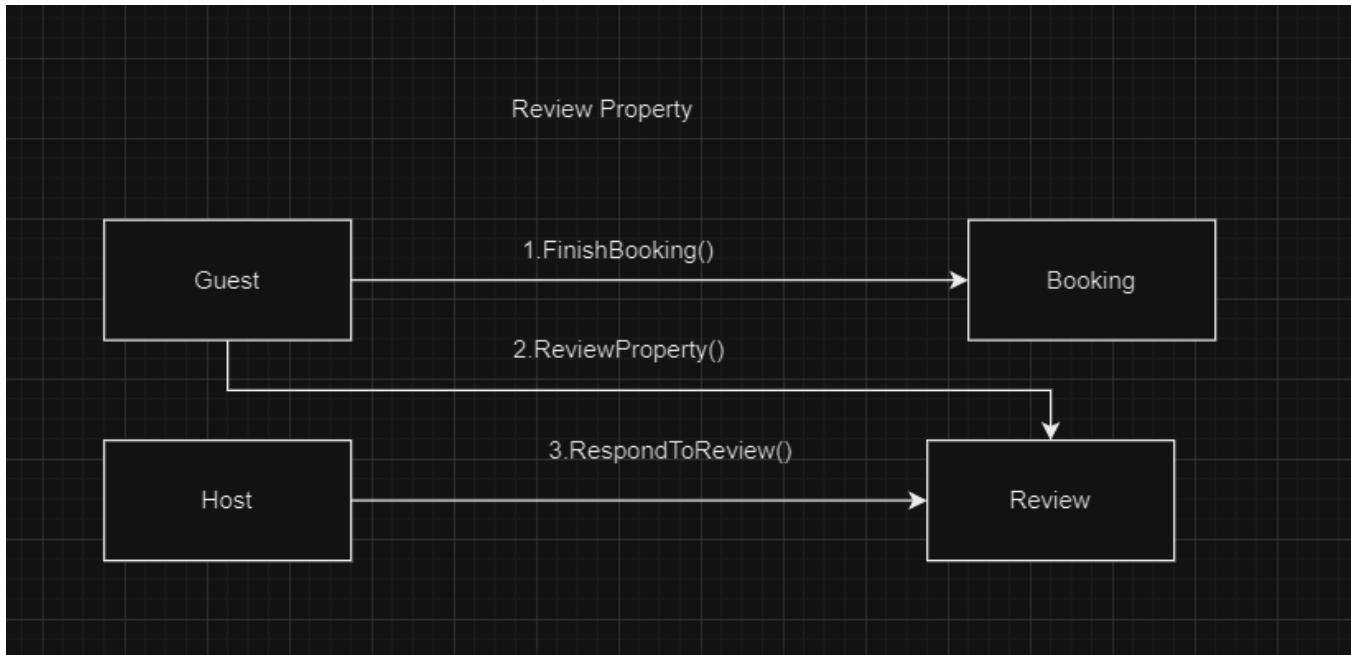
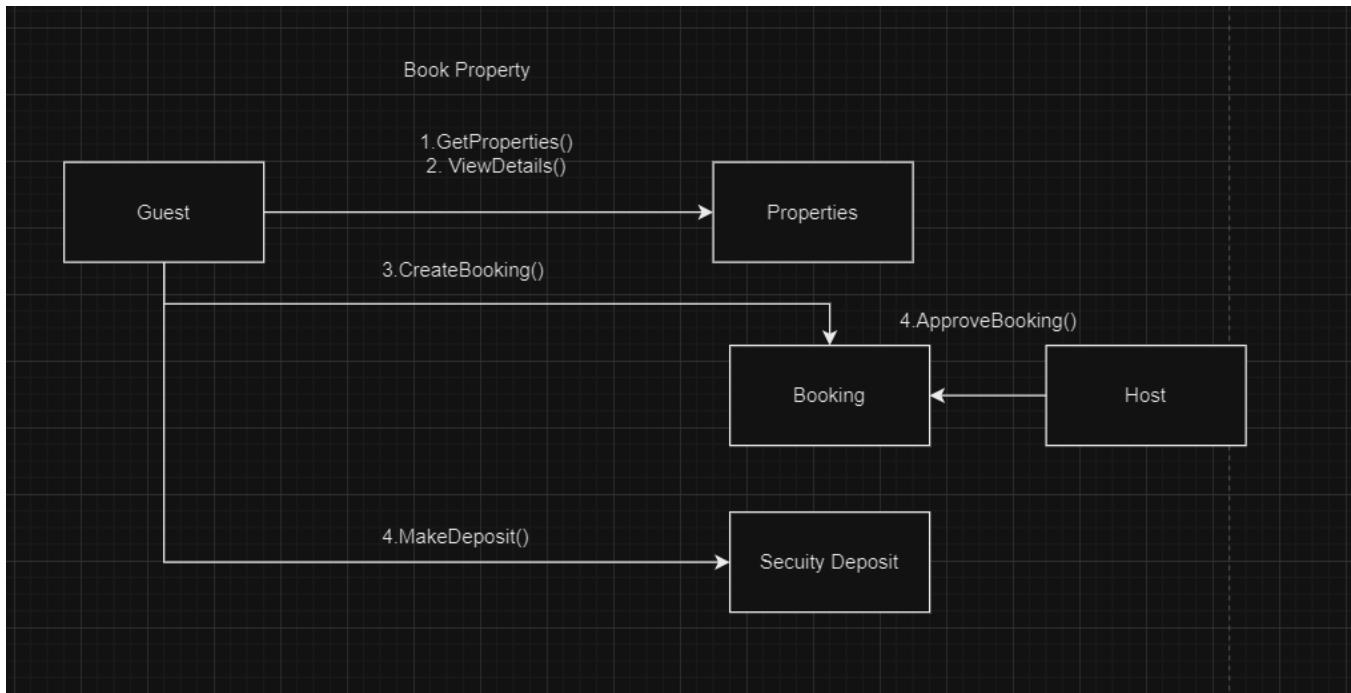


## Property Rentals Management System Requirements Specification

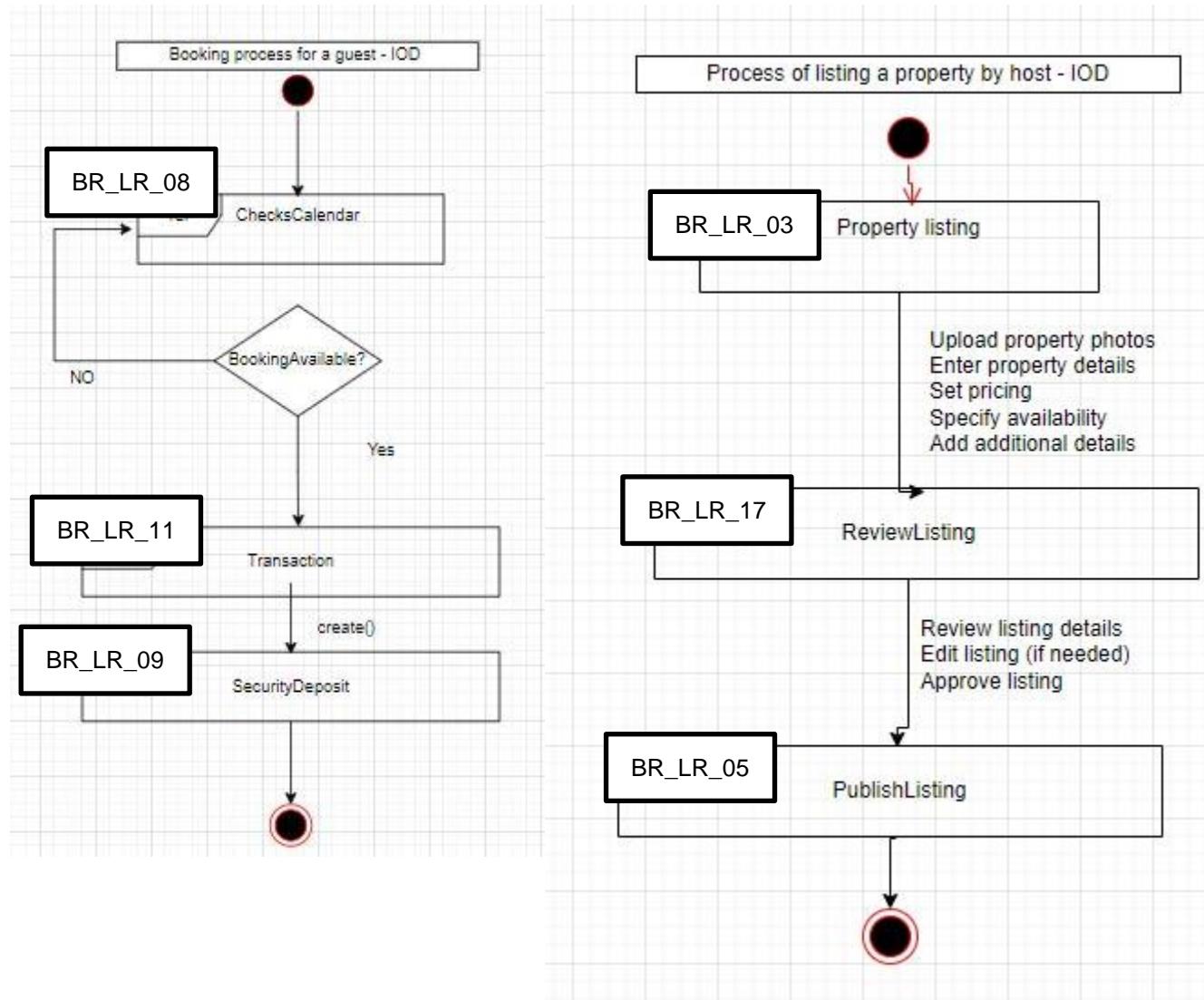
6.



## Property Rentals Management System Requirements Specification

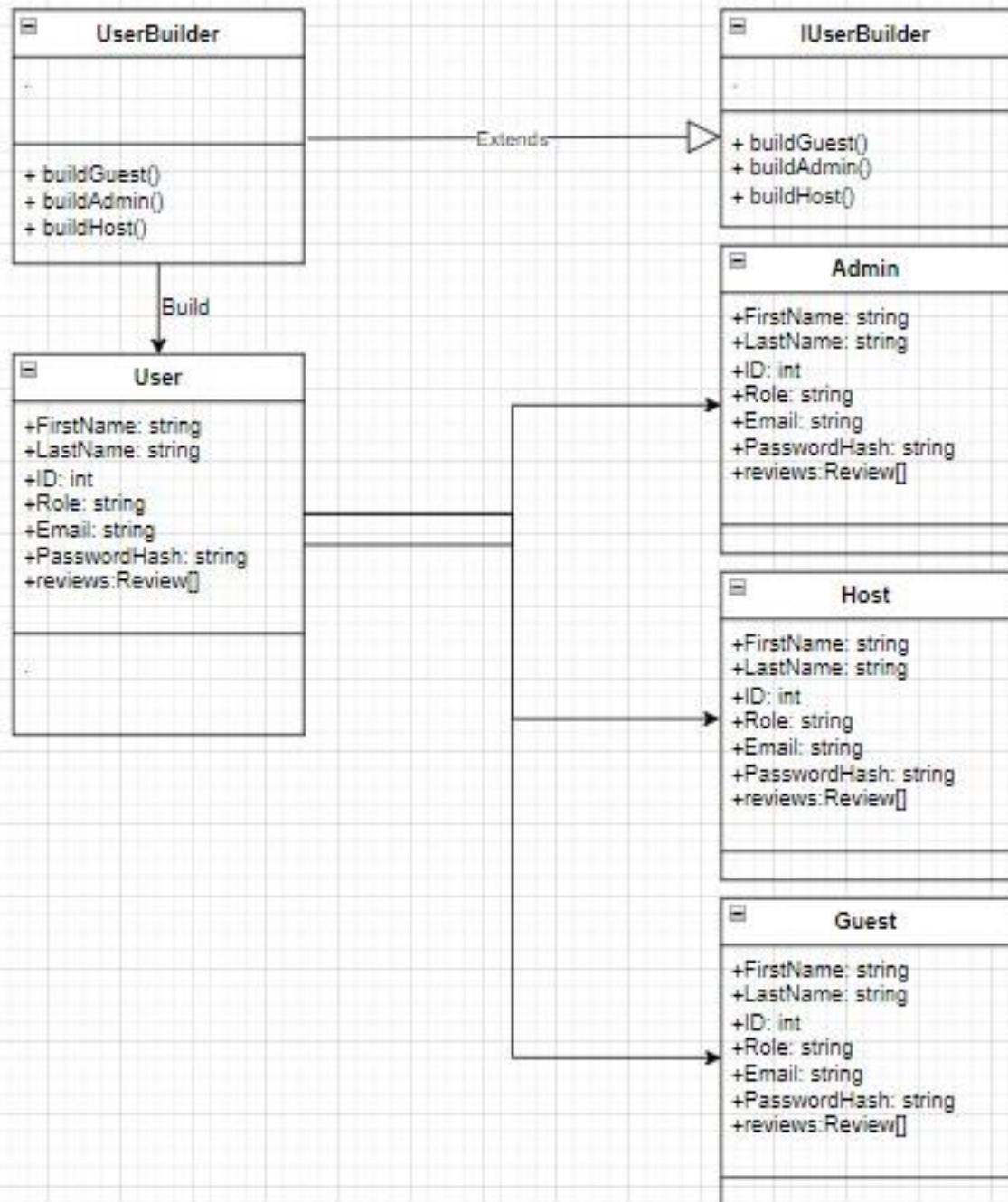


## INTERACTION OVERVIEW DIAGRAM

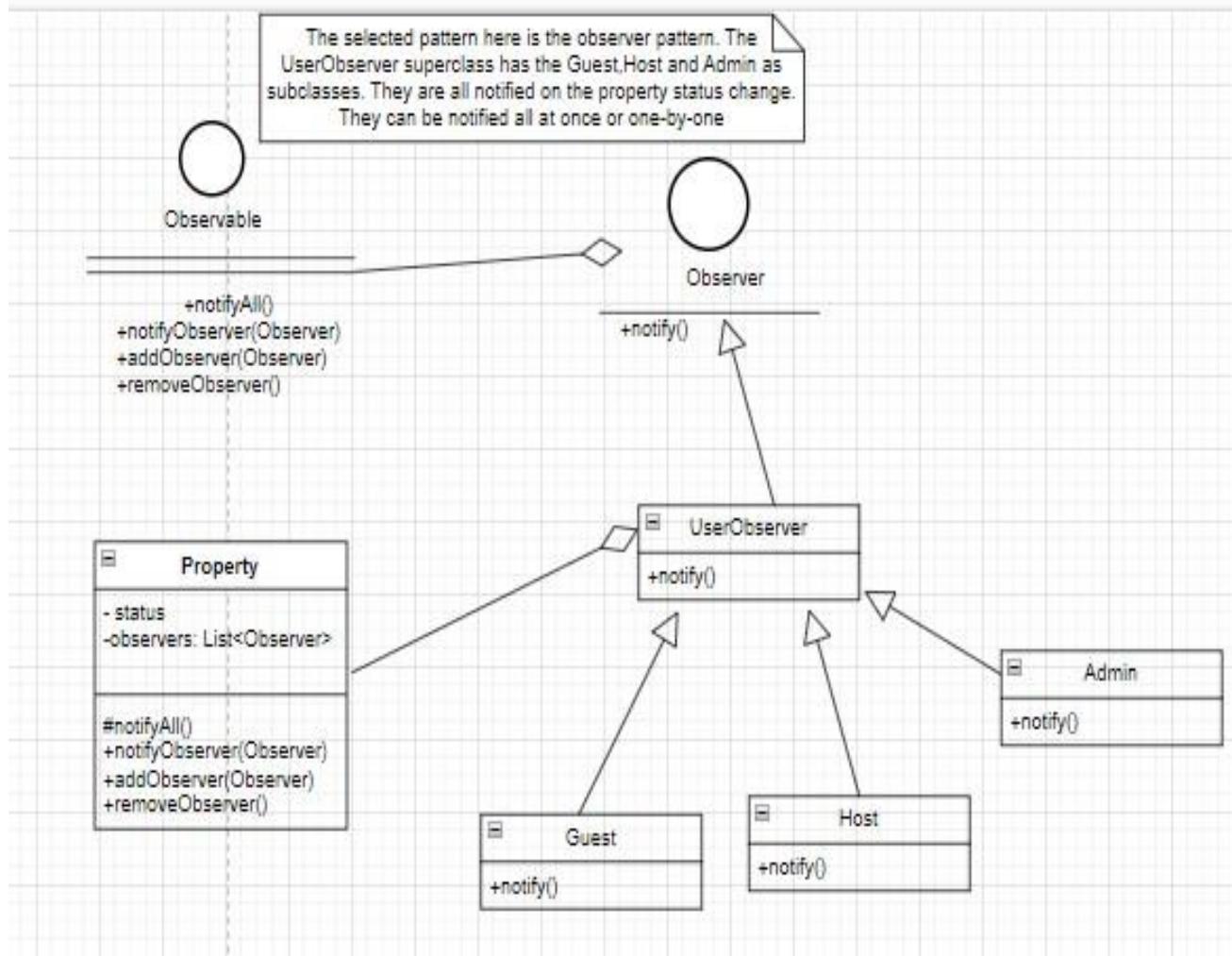


## 6. DESIGN PATTERNS

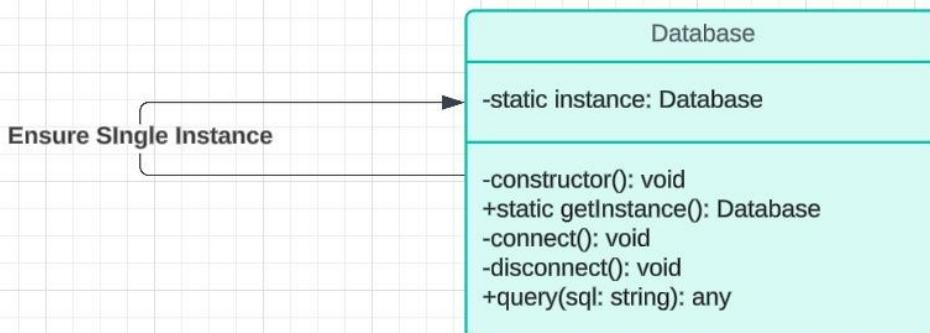
Builder pattern design for creating the admin, host and guest.



## Property Rentals Management System Requirements Specification



## SINGLETON



**APPENDIX**

**Appendix A. Definitions, Acronyms, and Abbreviations**

**PRMS -> PROPERTY RENTAL MANAGEMENT SYSTEM**

**Appendix B. References**

<https://github.com/alvinosh/PropertyRentEpokaSWE>