



# Property Rental Management System

---

SWE 202– Software Modeling & Design

Faculty of Engineering and Architecture

Epoka University

Martin Vila

Ema Kuka

Evina Tershalla

Entea Bakiasi

Mikael Xhangolli

Ersi Majkaj

Klea Koxha

# **Property Rental Management System Requirements Specification**

**Version 3.0**

**March 25th, 2025**

This document serves as the official Requirements Specification for the Property Rental Management System. It defines both functional and non-functional requirements, along with their priorities and approval statuses. The document is structured to align with the project's needs, including an overview, product context, and key system features.

The format and organization of the requirements are tailored to best support the project, ensuring clarity and usability. This document will be continuously updated as necessary to reflect changes, enhancements, and progress. Each version shows the progress.

## Table of Contents

|                                       |              |
|---------------------------------------|--------------|
| <b>1. Executive Summary</b>           | <b>4</b>     |
| <b>1.1 Project Overview</b>           | <b>4</b>     |
| <b>2. Product/Service Description</b> | <b>4</b>     |
| <b>2.1 Product Context</b>            | <b>4</b>     |
| <b>2.2 User Characteristics</b>       | <b>5-8</b>   |
| <b>2.3 Assumptions</b>                | <b>8</b>     |
| <b>3. Requirements</b>                | <b>8-25</b>  |
| <b>3.1 Functional Requirements</b>    | <b>8-18</b>  |
| 3.2 Non-Functional Requirements       | 18           |
| <b>3.2.1 Product Requirements</b>     | <b>18</b>    |
| 3.2.1.1 Usability Requirements        | 14           |
| 3.2.1.2 Performance Requirements      | 19           |
| 3.2.1.3 Availability                  | 19           |
| 3.2.1.4 Security                      | 23           |
| 3.2.2 Organizational Requirements     | 24-25        |
| 3.2.3 External Requirements           |              |
| <b>4. User Scenarios/Use Cases</b>    | <b>25-73</b> |

## 1. Executive Summary

### 1.1 Project Overview

Our group has chosen the Rental Management System topic for our project. We will ensure the creation of a proper model encompassing all the steps necessary to fulfill its role as a design worthy of simplifying the work of the software engineers to turn the theoretical aspect into a practical one. Our duty in this project is for all the lessons of the SWE 202 course to be incorporated into our project, for example: Requirements Engineering, different Modelings, Design Patterns, etc.

Everything will be posted on our Github Repository:

[https://github.com/martinvila22/SoftwareModelingAndDesign\\_Project.git](https://github.com/martinvila22/SoftwareModelingAndDesign_Project.git)

## 2. Product/Service Description

The Rental Management System is a multipurpose software, designed to ensure the needs of all types of users. Our System will create a safe, interactive environment for the verified landlords to post their legally and physically verified product which might be: an apartment, a mansion, a land, any other type of real estate.

This safe environment will also benefit a verified customer, the interested party on renting a property. It is also in the scope of the purpose of our system to ensure an easily understandable multifunctional search mechanism for all customers to search a property in a certain area, city, street and with certain filtering options decided by him such as: type of property, surface area, rooms, budget, etc.

There would be ofcourse a need for contact between these two parts of a deal, so we will also integrate an inner chat between the customer and the landlord. Since the developers of a Rental Management System should have thorough knowledge on the properties and the property rental market, the system will give feedback to the customer about the best aspects of the property and if the price is below or above average (in trying not to undermine the benefits of the property). In the scope of our System is also a smooth payment transaction between the two sides of the deal either by cash or direct payment through our system. In either case the System will give both parts a pdf verifying being a third party.

### 2.1 Product Context

Our System will be an independent one, with no relation to other products similar to this one. Even so, we plan to implement an inner chat between the customer and the landlord. This will be an inside the System chat, designed to function within our System. To create the front-end we will use Angular for a dynamic user interface, for the back-end we will use Java and MySQL for the database part. Microsoft Azure as a Cloud Service for

## Property Rental Management System Requirements Specification

Hosting. PayPal will be our to-go method for payments, even though we might consider other options to be added later.

### 2.2 User Characteristics

#### 1. Tenant (Client)

**Purpose:** A person renting a property who interacts with the system to find, lease, and maintain their rental.

##### **Functionalities:**

- Apply for rental properties – View available listings and submit rental applications
  - Sign lease agreements – Review and digitally sign lease contracts.
  - Pay rent online and view payment history – Make payments through an integrated system and track past transactions.
  - Submit maintenance requests – Report issues (e.g., plumbing, electrical, heating) and monitor progress.
  - Receive notifications – Get reminders for rent due dates, lease renewals, and maintenance updates.
- 

#### 2. Landlord (Property Owner)

**Purpose:** Owns and rents out properties while managing tenants and overseeing property performance.

##### **Functionalities:**

- List properties for rent – Upload property details, set rental prices, and manage availability.
  - Approve or reject tenant applications – Review applications, conduct background checks, and finalize leases.
  - Monitor rental income and payments – Track payment status and outstanding balances.
  - Approve or deny lease renewals – Decide whether to extend or terminate tenant leases.
  - Review and manage maintenance requests – Oversee maintenance operations in the case where a Property Manager is not assigned.
- 

#### 3. Property Manager

## Property Rental Management System Requirements Specification

**Purpose:** Manages daily operations for rental properties on behalf of landlords if assigned by a landlord.

### **Functionalities:**

- Oversee property listings and tenant applications – Manage rental postings and applicants for the property/ies.
  - Handle maintenance requests and coordinate with vendors – Maintain the property and be responsible if a problem arises.
  - Track rental income and expenses – Maintain financial records for property-related income and costs.
  - Communicate with tenants and landlords – Address tenant concerns and report to landlords.
  - Manage lease renewals and terminations – Facilitate lease extensions or legal processes for tenant removal.
- 

## **4. Support Staff (Customer Service)**

**Purpose:** Assists tenants, landlords, and property managers with inquiries, complaints, and troubleshooting issues.

### **Functionalities:**

- Assist tenants with account, payment, and maintenance issues – Help tenants reset passwords, process payments, and report problems.
  - Handle tenant and landlord inquiries – Answer questions about property listings, lease terms, and rent payments.
  - Resolve disputes and complaints – Mediate issues between tenants and landlords or escalate to higher authorities.
  - Manage customer service tickets – Track and help solve technical or service-related problems.
- 

## **5. Administrator (System Admin)**

**Purpose:** Manages the platform's infrastructure, security, and user roles to ensure smooth system operations.

### **Functionalities:**

- Manages user accounts – Modifies accounts for tenants, landlords, and property managers, in case of disputes or violations.

## Property Rental Management System Requirements Specification

- Oversee system security, settings, and backups – Ensure data protection, perform system updates, and prevent security breaches.
  - Generate system reports for performance and security – Monitor system usage and generate analytics.
  - Assign roles and permissions to users – Control access to different system functionalities based on user roles.
- 

### 6. Legal Advisor/s

**Purpose:** Ensures compliance with property laws, handles disputes, and advises on legal matters.

**Functionalities:**

- Review and approve lease agreements – Ensures rental contracts comply with legal regulations.
  - Advise landlords and property managers on legal matters – Provides guidance on evictions, tenant disputes, and property laws.
  - Draft and modify contracts – Updates lease terms to align with local regulations and prevent legal conflicts.
  - Oversee legal processes related to property disputes – Represents landlords in legal actions if necessary.
  - Review Identification documents from users- Ensures the Identification forms provided by the users are correct and gives approval for the account creation.
- 

### 7. Finance Officer

**Purpose:** Manages all financial aspects of the rental business, including transactions, reports, and budgeting.

**Functionalities:**

- Generate financial reports – Track rental income, operational expenses, and profit margins.
- Handle rent payment processing – Ensure smooth payment transactions, including deposits and overdue payments.
- Track tenant deposits, refunds, and financial records – Manage security deposits and process refunds.
- Assist with financial forecasting and budgeting – Plan for maintenance costs and future financial needs.

## Property Rental Management System Requirements Specification

### 2.3 Assumptions

1. Users should access our System on Android with an Android 12 or higher or an IOS 16 or higher. Failing to comply with this assumption will make the System unresponsive to the User and not available.
2. We presume that the use of our System will be available only on Portrait Mode and not on Landscape mode.
3. The System will be supported with periodical updates, for the maintenance and bug fixes.
4. Users have two forms of payments: cash or paypal. Other forms will not be accepted by our System.

### 3. Requirements

#### 3.1 Do Functional Requirements

| Req#  | Requirement  | Comments                       | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|--|--------------------------------|--------------|--------------|-------------------------------|
| BR_01 | The system shall allow tenants and landlords to securely register and log in. All other users (e.g., finance officers, legal officers, administrators) shall not be allowed to self-register but must be created by an administrator, who assigns their roles. The system shall enforce role-based access control (RBAC) to ensure users can only access features and data relevant to their roles, maintaining data integrity, privacy, and security. | Ensures secure access control. | 1            | 10/03/25     | Evina<br>Tershalla            |



## Property Rental Management System Requirements Specification

| Req#  | Requirement   | Comments   | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|---|--|--------------|--------------|-------------------------------|
| BR_02 | The system shall allow an administrator to manually create, modify, and delete user accounts for internal roles such as finance officers, legal officers, and other staff. Administrators shall assign specific access levels to user roles, ensuring that users can only perform authorized actions based on their responsibilities within the platform. | This enables a flexible and secure way to manage user activities, ensuring that only authorized users can perform critical operations. | 1            | 10/03/25     | Evina<br>Tershalla            |
| BR_03 | The system shall provide Two-Factor Authentication (2FA) for all user logins and lock accounts after five failed login attempts. Locked users shall receive an email notification with recovery options to regain access.   | Enhances security by preventing unauthorized access and ensuring account recovery options are available.                               | 2            | 10/03/25     | Evina<br>Tershalla            |
| BR_04 | The system shall implement a user account recovery process that allows users to reset their passwords securely through email verification. This process shall include security questions or alternative verification methods to ensure the identity of the user requesting the reset.   | Prevents unauthorized access.  | 1            | 10/03/25     | Evina<br>Tershalla            |

## Property Rental Management System Requirements Specification

| Req#  | Requirement  | Comments  | Priority | Date Rvwd | SME Reviewed / Approved |
|-------|--|---|----------|-----------|-------------------------|
| BR_05 | The legal office shall be able to verify tenants, landlords, and property managers before they can create a new account, because this process ensures that only trusted and authenticated individuals are granted access to the software.    | Enhances security and ensures trust and reliability in the system.  | 1        | 10/03/25  | Ema Kuka                |
| BR_06 | Administrators shall be able to suspend or deactivate user accounts that violate platform policies because this helps maintain security and compliance.  | Ensures platform integrity by preventing misuse, enhancing security, and enforcing compliance with policies.                          | 1        | 10/03/25  | Ema Kuka                |
| BR_07 | The legal office shall conduct an additional verification step for new landlord accounts by reviewing both proof of property ownership and identity documents before account approval.   | Ensures that only legitimate property owners can list properties, preventing fraudulent listings and maintaining trust in the system. | 1        | 10/03/25  | Ema Kuka                |
| BR_08 | The system shall incorporate an event calendar feature that allows tenants and landlords to schedule and manage property viewings, maintenance appointments, and community events, improving organization and communication between parties. | This feature enhances scheduling efficiency, reduces conflicts, and improves communication between tenants and landlords.             | 3        | 10/03/25  | Mikael Xhangolli        |
| BR_09 | Landlords shall be able to list, update, or remove rental properties with photos, descriptions, rent prices, and amenities.  | Ensures updated and accurate listings.  | 2        | 10/03/25  | Evina Tershalla         |

## Property Rental Management System Requirements Specification

| Req#  | Requirement  | Comments   | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|--|--|--------------|--------------|-------------------------------|
| BR_10 | The system shall handle errors that occur during data processing, such as database connection failures, file read/write errors, or API call failures, without causing system crashes.  | Prevents crashes.  | 1            | 10/03/25     | Evina<br>Tershalla            |
| BR_11 | Landlords shall be able to handle maintenance or repairs for their properties through the system if they have appointed a property manager. If no property manager has been assigned, the landlord will be responsible for handling maintenance issues directly. | Ensures that landlords can efficiently address property maintenance issues, improving tenant experience and preserving property value. | 2            | 10/03/25     | Ema Kuka                      |
| BR_12 | Landlords shall be able to schedule property viewings and manage appointment slots through the system because this allows potential tenants to visit properties at convenient times, improving the rental process.   | Streamlines property visits, ensuring efficient coordination between landlords and tenants.  | 2            | 10/03/25     | Ema Kuka                      |
| BR_13 | Property managers shall be able to schedule and manage property viewings on behalf of the landlord, if permitted because this allows landlords to delegate property management tasks while maintaining an organized scheduling process.                          | Ensures that property managers can handle operational aspects efficiently, supporting landlords when needed.                           | 2            | 10/03/25     | Ema Kuka                      |

## Property Rental Management System Requirements Specification

| Req#  | Requirement   | Comments  | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|---|---|--------------|--------------|-------------------------------|
| BR_14 | The system shall allow landlords or property managers if they are given permission, to view summarized insights on tenant behavior, such as payment timeliness, maintenance request frequency, and feedback history, to help them make better rental decisions. | Enables landlords to see a summary of how reliable a tenant has been in the past.   | 2            | 10/03/25     | Ema Kuka                      |
| BR_15 | Tenants can search for rental properties using various filters such as price range, location, number of rooms, and amenities, allowing them to efficiently find suitable properties that meet their specific needs.   | Makes choices easier and more customized.   | 1            | 10/03/25     | Mikael Xhangolli              |
| BR_16 | Tenants shall be able to submit a rental application for a property by providing necessary documents and personal information.  | Defines tenant and finance office responsibilities, ensuring a streamlined rental process and proper payment validation.  | 1            | 10/03/25     | Ema Kuka                      |
| BR_17 | Landlords shall be able to review, approve, or reject rental applications submitted by tenants. Landlords may also delegate this responsibility to assigned Property Managers via the system.   | Landlords have control over tenant selection and flexibility to delegate this right, ensuring suitability and compliance. | 1            | 10/03/25     | Ema Kuka                      |

## Property Rental Management System Requirements Specification

| Req#  | Requirement   | Comments  | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|---|---|--------------|--------------|-------------------------------|
| BR_18 | Tenants shall be able to request lease renewals within 30 days of expiration through the system, and receive confirmation or rejection from the landlord or property manager.   | Allows tenants to express interest in staying in the same property and helps landlords easily review and respond to renewal requests. | 2            | 10/03/25     | Ema Kuka                      |
| BR_19 | The system shall allow tenants to view a rental history report, including past properties rented, payment records, and feedback from landlords.   | Helps tenants provide credibility when applying for new rentals.  | 2            | 10/03/25     | Evina<br>Tershalla            |
| BR_20 | The system shall allow landlords to set custom lease terms and conditions for each rental property, including pet policies, security deposit amounts, and maintenance responsibilities. Tenants must acknowledge and agree to these terms before submitting a rental application. | Improves lease management.  | 1            | 10/03/25     | Evina<br>Tershalla            |
| BR_21 | The system shall securely process tenant payments and generate electronic receipts upon successful payment.   | Prevents fraud and ensures safe processing of rent payments and generates receipts.   | 1            | 10/03/25     | Evina<br>Tershalla            |
| BR_22 | The system shall allow landlords to set automatic late fee penalties for overdue rent payments and notify tenants accordingly.  | Helps automate the enforcement of rental policies and improves financial accountability.  | 2            | 10/03/25     | Evina<br>Tershalla            |

## Property Rental Management System Requirements Specification

| Req#  | Requirement   | Comments  | Priority | Date Rvwd | SME Reviewed / Approved |
|-------|---|---|----------|-----------|-------------------------|
| BR_23 | Administrators shall be able to generate detailed reports on platform activity, including user logins, security logs, and system performance metrics because this helps monitor platform stability and detect anomalies.  | Provides insights into platform usage, helps identify trends, and ensures early detection of irregular activities for better decision-making. | 2        | 10/03/25  | Ema Kuka                |
| BR_24 | Tenants can receive automated payment reminders regarding upcoming rent due dates, and they can access their payment history, including transaction details and payment methods.  | This feature aids in tracking finances and promotes financial responsibility.   | 1        | 10/03/25  | Mikael Xhangolli        |
| BR_25 | The system shall allow users (tenants, landlords, etc.) to report any security incidents (e.g., hacking attempts, data breaches) directly to the administrator, ensuring prompt action and investigation.   | Helps keep the parties safe.  | 1        | 10/03/25  | Evina Tershalla         |
| BR_26 | The system shall provide a platform-wide notification system for users, including tenants, landlords, property managers, and administrators, to receive alerts about upcoming system maintenance, feature updates, or critical issues that may affect platform functionality. | This ensures users are informed in advance about system changes or downtime, improving transparency and reducing disruptions.                 | 3        | 10/03/25  | Evina Tershalla         |

## Property Rental Management System Requirements Specification

| Req#  | Requirement  | Comments  | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|--|---|--------------|--------------|-------------------------------|
| BR_27 | The system shall include an AI-powered chatbot that provides instant responses to frequently asked questions, assists with common rental-related inquiries, and escalates complex issues to human support staff when needed. | Provides support.   | 1            | 10/03/25     | Evina<br>Tershalla            |
| BR_28 | The system shall provide tenants with a secure messaging feature to communicate directly with landlords or property managers, improving communication regarding maintenance, lease terms, and other property-related issues. | Ensures the communication between parties,                            | 2            | 10/03/25     | Evina<br>Tershalla            |
| BR_29 | The system shall allow landlords to create and manage promotional offers or discounts on rental properties, such as reduced rent for the first month or referral bonuses, to attract potential tenants.                      | Marketing for landlords.  | 2            | 10/03/25     | Evina<br>Tershalla            |
| BR_30 | Tenants can provide feedback and rate their experience with landlords and property managers, including comments on maintenance responsiveness, property condition, and overall satisfaction.                                 | Admins can view and manage this feedback to ensure quality assurance. | 2            | 10/03/25     | Mikael<br>Xhangolli           |

## Property Rental Management System Requirements Specification

| Req#  | Requirement  | Comments  | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|--|---|--------------|--------------|-------------------------------|
| BR_31 | Users can select their preferred language for the interface and notifications, with options available in major regional languages. This feature enhances inclusivity for users from different demographics.  | This feature promotes inclusivity and improves user experience across different demographics.                                   | 2            | 10/03/25     | Mikael Xhangolli              |
| BR_32 | New users can complete an onboarding process that includes guided tutorials and tips on using various features when they first log in.   | Users can revisit these tutorials at any time from the help section to enhance their familiarity with the platform.             | 2            | 10/03/25     | Mikael Xhangolli              |
| BR_33 | The system shall include a frequently asked questions (FAQ) section where users can find quick answers to common inquiries about the rental process, payment methods, and system usage.  | Provides the finance office with necessary information to validate and confirm tenant payments efficiently.                     | 2            | 10/03/25     | Mikael Xhangolli              |
| BR_34 | The system shall be able to analyze rental prices and conditions to provide feedback to tenants on whether a property's price is below, above, or at the market average , while not undermining the positive sides because this helps tenants make informed financial decisions. | Enhances transparency in the rental market and ensures that tenants have insights into fair pricing before making a commitment. | 2            | 10/03/25     | Ema Kuka                      |



## Property Rental Management System Requirements Specification

| Req#  | Requirement   | Comments   | Priori<br>ty | Date<br>Rvwd | SME<br>Reviewed /<br>Approved |
|-------|---|--|--------------|--------------|-------------------------------|
| BR_35 | Landlords and property managers shall be able to generate detailed financial reports because tracking income and expenses is essential for managing property finances.  | Ensures financial transparency and easy tracking of income and costs.  | 1            | 10/03/25     | Evina Tershalla               |
| BR_36 | Support staff shall be able to track and help resolve technical support requests submitted by tenants, which must first be reviewed and approved by the landlord or an assigned property manager, to ensure proper handling and a smooth user experience. | Enhances user satisfaction by ensuring that technical issues are promptly identified and resolved, minimizing disruptions. | 2            | 10/03/25     | Ema Kuka                      |
| BR_37 | Administrators shall be able to remove policy-violating content, including property listings and tenant reviews, after receiving reports or detecting violations because this helps maintain platform integrity and security.                             | Helps maintain content accuracy and trustworthiness within the platform.   | 1            | 10/03/25     | Ema Kuka                      |
| BR_38 | Administrators shall be able to set up and manage platform-wide announcements, such as system updates, scheduled maintenance, and new feature releases because this keeps users informed and reduces unexpected disruptions.                              | Keeping users updated about changes improves communication and reduces confusion.  | 3            | 10/03/25     | Ema Kuka                      |

## Property Rental Management System Requirements Specification

| Req#  | Requirement   | Comments  | Priorty | Date Rvwd | SME Reviewed / Approved |
|-------|---|---|---------|-----------|-------------------------|
| BR_39 | Users can update their profile information, including contact details, preferences, and profile pictures.   | Changes will be saved and reflected in their accounts, allowing for user customization and maintaining accurate information.          | 1       | 10/03/25  | Mikael Xhangolli        |
| BR_40 | Legal advisors can mediate conflicts between landlords/property managers and tenants, providing suggested legal actions, mediation procedures, and potential settlements through the system.  | This feature ensures the Users of our role and makes the system more trustworthy.   | 2       | 10/03/25  | Mikael Xhangolli        |
| BR_41 | Users shall be able to upload, view, and manage important documents (such as ID proofs, lease agreements, ownership papers, and payment receipts) in a secure document storage area, where access is controlled based on the user's role. | Provides a central place for each user to safely manage documents, while ensuring that only authorized roles can view specific files. | 2       | 10/03/25  | Mikael Xhangolli        |

### 3.2 Non-Functional Requirements

#### 3.2.1 Product Requirements

Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.

##### 3.2.1.1 Usability Requirements

- The system shall have an intuitive and user-friendly interface to accommodate users with varying technical expertise.
- The interface shall follow Material Design principles to ensure consistency, clarity, and ease of use.

## Property Rental Management System Requirements Specification

- The system shall support responsive design, ensuring compatibility with mobile and tablet devices.
- The onboarding process for new users shall not exceed 5 minutes to facilitate quick adoption.
- Online documentation and a user manual shall be provided for assistance, ensuring learnability.
- System navigation shall require no more than three clicks to reach any key functionality.
- The map interface shall support zooming, panning and property selection within minimal lag across all devices.
- Users should be able to see a summary of the key points from review the system has made before clicking to read or see the full post.

### 3.2.1.2 Performance Requirements

- The system shall support a minimum of 1,000 concurrent users without performance degradation.
- 95% of transactions shall be processed within 2 seconds, ensuring smooth user experience.
- System response time for database queries shall not exceed 300 ms under normal load conditions.
- The system shall handle at least 10,000 property listings and 50,000 registered users efficiently.
- Peak load performance shall be tested to ensure the system can scale dynamically.
- The map should display up to 1000 listings without noticeable lag, ensuring property markers load within 5 seconds.
- Users should be able to see a review within 5 seconds ensuring no lag or delay.

### 3.2.1.3 Availability & Security

- The system shall maintain 99.9% uptime, ensuring that the platform is accessible 24/7 for property searches, bookings, payments, and other activities and also *it* must be able to handle peak traffic periods, such as weekends, holidays, and seasonal spikes, without performance degradation.
- The system must support at least 500 concurrent users for booking, payment, and search activities, without delays or system lag.
- The system must automatically update property availability when a property is booked or its price changes, ensuring that users see real-time updates.
- The system should allow users to set alerts for specific property types or location preferences, sending notifications for new listings as they become available.
- Users should receive real-time notifications via email, SMS, or mobile push notifications for property updates, such as booking confirmations, cancellations.
- If something goes wrong and the booking process is interrupted like during a network failure, system crash, or power outage users will be able to pick up right where they left

## Property Rental Management System Requirements Specification

off without losing any of their progress. All details will remain intact, except for sensitive information like credit card details, which are never saved for security reasons.

- The system must use two-factor authentication (2FA) for users when logging in. This means users will need to enter both their password and a one-time passcode (OTP) sent via email or -SMS.
- Passwords must be securely stored using strong encryption methods. Passwords must be at least 8 characters long, not contain personal information (such as name or surname), must contain a capital letter and must allow special characters.
- The system must limit login attempts to 5 failed tries within 30 minutes. After this, the system should temporarily lock the account for a predetermined period. Email notifications or phone messages should be sent to the user with the failed attempts and a link to regain access.
- The system must track user activity and extend the session only if there is continuous interaction. After 30 minutes of inactivity, users will be automatically logged out and prompted to re-authenticate. A warning message should appear 5 minutes before logout, allowing users to stay logged in if needed.
- The system must employ automated spam filters to detect and block inappropriate or irrelevant content submitted in reviews. Any review flagged by the system or users must be reviewed by a moderator within 24 hours to ensure compliance with content standards and guidelines.
- The system must ensure that sensitive data (user payment details, personal information) is only accessible to authorized personnel with the correct role permissions.
- All sensitive data, including payment information and user details, must be encrypted both in transit and at rest to protect it from unauthorized access.
- The system must log the IP address of users during login attempts and other sensitive actions (password changes, account settings modifications).

### 3.2.1.5 Reliability and Fault Tolerance

The system shall be designed to ensure high availability, fault tolerance, and reliability, minimizing service disruptions and data loss.

Reliability Measures:

- Uptime Guarantee: The system shall maintain 99.9% uptime, ensuring uninterrupted access for users.
- Redundant Infrastructure: Critical components such as databases, servers, and storage shall be replicated across multiple locations to prevent data loss and downtime.
- Data Integrity Checks: The system shall implement periodic data consistency checks and automatic error correction mechanisms to prevent corruption.

## **Property Rental Management System Requirements Specification**

### **Fault Tolerance Mechanisms:**

- Automatic Failover: In case of server failure, traffic shall be redirected to backup servers to ensure continuous operation.
- Graceful Degradation: If certain features (e.g., payment processing) become unavailable, the system shall ensure that core functionalities (e.g., property browsing, messaging) remain operational.

### **Error Detection and Recovery:**

- The system shall log and classify errors based on severity and trigger automated responses for minor issues.
- For critical failures, an alert shall be sent to system administrators for immediate intervention.
- Load Balancing: The system shall distribute traffic evenly across multiple servers to prevent overloading and maintain response times.

### **Testing for Reliability:**

- The system shall undergo stress testing, failover testing, and disaster recovery testing before deployment.
- It shall be capable of handling peak traffic loads without performance degradation.

### **3.2.1.6 System Maintenance**

The system shall follow a structured maintenance strategy to ensure security, performance, and feature updates without disrupting user experience.

#### **Maintenance Types:**

##### **1. Scheduled Maintenance**

- Updates, security patches, and performance optimizations shall be applied outside peak hours to minimize downtime.
- Users shall receive advance notifications via email, SMS, and in-app messages.
- A maintenance mode shall be activated, displaying a message to users during system updates.

##### **2. Unscheduled Maintenance (Emergency Fixes)**

- In case of security threats, major system failures, or critical bugs, emergency patches shall be deployed immediately.

## **Property Rental Management System Requirements Specification**

- Users shall be notified about the estimated resolution time and impact on functionality.

### **3. Automated Monitoring and Self-Healing**

- The system shall include real-time monitoring to detect performance issues, slow queries, or high server loads.
- Automated scripts shall restart failed services without manual intervention.

### **4. Backup and Disaster Recovery**

- Daily automatic backups of databases and configurations shall be performed.
- A disaster recovery plan shall ensure that the system can be restored within 30 minutes in case of data loss.
- Backups shall be stored in multiple geographically distributed data centers for redundancy.

### **5. Software Updates and Version Control**

- The system shall follow a CI/CD (Continuous Integration/Continuous Deployment) approach for smooth rollouts.
- A rollback mechanism shall be in place to revert updates if critical issues arise.

#### **System Maintenance Reports:**

- Detailed logs of maintenance activities shall be stored for compliance and auditing purposes.
- Performance reports shall be generated monthly to track system efficiency and identify areas for improvement.

#### **3.2.1.7 Logging**

- The system must log all login attempts, including successful and failed attempts, with details like timestamp, user ID, IP address, device type, and location. Failed attempts must also include the reason for failure (incorrect password).
- If an account is locked due to multiple failed login attempts, the system must log the event with details such as user ID, number of failed attempts, IP address, and timestamp.
- The system must log all user session activities, including the start time, end time, and actions performed during the session. It should also log instances where users are logged out due to inactivity or manual action.
- Whenever a review is flagged by the spam filter, the system must log the event, including review content, reason for flagging, and the timestamp.

## Property Rental Management System Requirements Specification

- Any changes made to user roles ( assigning admin privileges or changing user permissions) must be logged, including who performed the action, what change was made, and the timestamp.
- Any changes to access control settings (permission modifications, new role creations) should be logged, including who made the change, what was changed, and when.

### 3.2.2 Organizational Requirements

- **Code Review and QA Process:** The system must undergo both automated and manual testing to verify its functionality and quality. Automated tests should cover critical workflows and edge cases, while manual testing should focus on scenarios that require human judgment. Peer reviews of the code should be conducted to ensure adherence to coding standards, identify potential bugs, and ensure the system operates as expected before deployment.
- **Compliance with Internal Security Policies:** The system should comply with internal security policies, including secure user authentication methods, access control mechanisms, and the protection of sensitive data. This ensures the system is safeguarded against potential security threats and adheres to privacy and regulatory requirements.
- **System Documentation:** Detailed documentation of the system's architecture, database design, and APIs should be created and regularly updated. This documentation will serve as a reference for senior developers and help ensure the system is scalable and maintainable over time. It should be clear, organized, and comprehensive to facilitate future modifications and troubleshooting.
- **User roles and permissions:** The application should have predefined roles for tenants , landlords, property managers and administrators each with specific permissions.
- **Customer support and response time:** Support staff must respond to user inquiries within 24 hours.

### 3.2.3 External Requirements

#### External Requirements Overview

- **Social Media Integration** -The platform should allow users to share listings or bookings on popular social media platforms such as Facebook, Instagram, or Twitter, using seamless integration through their APIs.
- **External Service Integration**- The system should integrate with external communication platforms (e.g., email services, SMS providers) to send notifications, confirmations, and updates to users. APIs like SendGrid or Twilio can be used for this integration.
- **Consumer Protection Laws** -The platform must ensure that booking policies, cancellation fees, and refund processes comply with consumer protection laws in different regions. The system should allow for regional configuration to adhere to specific local regulations.
- **Two-Factor Authentication (2FA)**-The platform should support Two-Factor Authentication (2FA) for users accessing sensitive features, such as property

## **Property Rental Management System Requirements Specification**

management, payment processing, or personal account information. This provides an added layer of security.

- Multi-Language Support-The platform should offer multi-language support, enabling users from various regions to interact with the system in their preferred language. This should be configurable based on the user's region or preference.

### **3.2.3.3 Integration with External Systems**

#### **1. Payment Gateways**

- PayPal is supported in Albania, but not all users have it. Therefore, the system shall also integrate with:
- Bank Transfers (Raiffeisen, Credins, BKT, OTP, Intesa Sanpaolo) via local banking APIs.
- E-Wallets (Easypay, Paysera) for users preferring digital payments.
- Cash-on-Delivery (CoD): Many Albanians still prefer cash transactions. The system shall generate QR-code-based receipts for easy verification.

#### **2. Hosting & Compliance with Data Protection Laws**

- GDPR Compliance: Since Albania is a candidate for EU accession, the system must ensure that data processing follows European GDPR standards. (EU general data protection regulations)

#### **3. Property Listing Verification via Government Services**

- The system shall integrate with Albania's Property Registration Office (ZRPP) to verify property ownership before allowing landlords to list properties.
- Integration with e-Albania for digital identity verification shall ensure that landlords and tenants use real verified credentials.

#### **4. Communication & Notifications**

- SMS notifications should use local providers (e.g., Vodafone, ALBtelecom, One Albania) to ensure cost-effective and reliable messaging but also offer support for international numbers
- Multi-language support: The system shall offer Albanian (default) and English, as many users (e.g., expatriates, foreign renters) may prefer English.

#### **5. Legal Compliance & Identity Verification**

- Tenant Contracts Must Follow Albanian Law



## Property Rental Management System Requirements Specification

- The system shall ensure lease agreements comply with Law No. 7850/1994 (Albanian Civil Code) regarding rental contracts.
- The system shall provide contract templates with terms aligned with Albanian rental laws.
- Identity Verification via National ID (Integrating with e-Albania) to prevent fraud.

### 4. User Scenarios/Use Cases

|                                  |   |
|----------------------------------|---|
| UC Name                          | <b><u>UC-01: User Registration and Role Based Access</u></b>  |
| Summary                          | This use case describes how tenants and landlords can self-register and log in to the system. Other roles cannot self register and must be created by an administrator. The system enforces role based access control to restrict access based on user roles.   |
| Dependency                       | <b>UC-02:</b> Login authentication<br><b>UC-03:</b> Admin User Management   |
| Actors                           | <b>Primary Actor:</b> Tenant, Landlord<br><b>Secondary Actors:</b> System(validates registration),<br>Administrator(manages restricted roles)   |
| Preconditions                    | <ul style="list-style-type: none"><li>- The system is online and operational.</li><li>- The user is not already registered.</li></ul>   |
| Description of the Main Sequence | <ol style="list-style-type: none"><li>1. User selects 'Sign Up' on the login page.</li><li>2. The system displayed the registration form with only the roles Tenant and Landlord as options(restricted roles are not shown.)</li><li>3. The user enters their details.</li><li>4. The system validates the input:<ol style="list-style-type: none"><li>a. Checks all required fields are filled.</li><li>b. Ensures email is in valid format.</li></ol></li><li>5. The user submits the form.</li><li>6. The system checks if the email is unique:<ol style="list-style-type: none"><li>a. If the email is already in use, an error is displayed.</li><li>b. If the email is available it proceeds.</li></ol></li><li>7. The system stores the data and sends a confirmation email.</li><li>8. The system redirects the uder to login page.</li></ol> |
| Description of the Alternative   | <b>A1: User enters an existing email:</b> <ol style="list-style-type: none"><li>1. The user enters an email that it already exists.</li><li>2. The system displays an error "Email already exists. Please log</li></ol>   |

## Property Rental Management System Requirements Specification

|                             |   |
|-----------------------------|---|
| Sequence                    | <p>in or try a different email.”</p> <p><b>A2: User Enters Invalid Input:</b></p> <ol style="list-style-type: none"><li>1. The user leaves required fields blank or has invalid format</li><li>2. The system displays error messages.</li></ol>   |
| Non functional requirements | <p><u>Usability Requirements</u></p> <p>The system shall have an intuitive and user-friendly interface to accommodate users with varying technical expertise.</p> <p>The onboarding process for new users shall not exceed 5 minutes to facilitate quick adoption.</p> <p>(Referenced from 3.2.1.1 - Usability Requirements.)</p> <p><u>Performance Requirements</u></p> <p>The system shall handle at least 50,000 registered users efficiently.</p> <p>(Referenced from 3.2.1.2 - Performance Requirements.)</p> <p><u>Security &amp; Availability Requirements</u></p> <p>The system must use Two-Factor Authentication (2FA) for user logins.</p> <p>Passwords must be securely stored and encrypted and must be at least 8 characters long, include a capital letter, and allow special characters.</p> <p>The system must limit login attempts to 5 failed tries within 30 minutes.</p> <p>(Referenced from 3.2.1.3 - Availability &amp; Security.)</p> |
| Postconditions              | <ul style="list-style-type: none"><li>- A tenant or landlord account is successfully created.</li><li>- The user can now log in.</li><li>- Other roles cannot self-register and must be created by an administrator</li></ul>   |

|               |   |
|---------------|---|
| UC Name       | <b><i>UC-02 – Role-Based Access Control &amp; User Role Assignment</i></b>  |
| Summary       | This use case describes how administrators can assign specific access levels to users, ensuring that users can only perform actions and access data relevant to their roles within the platform. The system enforces role-based access control (RBAC) to ensure users can only access features and data authorized based on their assigned roles. |
| Dependency    | <b>UC-01:</b> User Registration & Role-Based Access (as users need to be registered and authenticated before role assignments can be made).   |
| Actors        | <p><b>Primary Actor:</b> Administrator</p> <p><b>Secondary Actor:</b> System (ensures correct role assignment and access control)</p>   |
| Preconditions | <ul style="list-style-type: none"><li>- The user is already registered and logged in.</li></ul>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | <ul style="list-style-type: none"> <li>- The user is an administrator with the appropriate permissions to assign roles.</li> <li>- The user does not have a restricted role assigned to them already.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The administrator logs into the system with admin credentials.</li> <li>2. The system validates the administrator's credentials and grants access to the admin dashboard.</li> <li>3. The administrator navigates to the user management page and selects the user to assign a role to.</li> <li>4. The system displays a list of registered users who need role assignment.</li> <li>5. The administrator selects the user and proceeds to the role assignment section.</li> <li>6. The system displays available roles (e.g., Tenant, Landlord, Finance Officer, Legal Officer) with permissions associated with each role.</li> <li>7. The administrator selects the appropriate role based on the user's responsibilities.</li> <li>8. The system assigns the selected role to the user and updates the user's role in the database.</li> <li>9. The system notifies the user of their new role and access permissions via email.</li> <li>10. The system ensures the user now has access to only authorized features based on their role.</li> </ol> |
| Description of the Alternative Sequence | <p><b>A1: Administrator Tries to Assign an Invalid Role</b></p> <ol style="list-style-type: none"> <li>1. The administrator selects a role that the user is not authorized to receive (e.g., assigning a restricted role to a non-admin).</li> <li>2. The system displays an error message: <ul style="list-style-type: none"> <li>• "This role cannot be assigned to the user. Please select a valid role."</li> </ul> </li> </ol> <p><b>A2: Administrator Assigns a Role to a User Without Permissions</b></p> <ol style="list-style-type: none"> <li>1. The administrator does not have permission to assign the selected role (e.g., restricted role).</li> <li>2. The system displays an error message: <ul style="list-style-type: none"> <li>• "You do not have the necessary permissions to assign this role."</li> </ul> </li> </ol>   |
| Non functional requirements             | <p><u>Usability Requirements</u></p> <p>The administrator interface shall be user-friendly and allow easy navigation between user management sections.</p> <p>The role assignment process should not exceed 2 minutes to ensure quick and efficient management of users.</p> <p>(Referenced from 3.2.1.1 - Usability Requirements.)</p>   |

## Property Rental Management System Requirements Specification

|                |  |
|----------------|--|
|                | <p><u>Performance Requirements</u></p> <p>The system shall be able to process role assignments efficiently, with the operation completing in less than 2 seconds.</p> <p>The system shall handle at least 500 active administrators performing role assignments simultaneously.</p> <p>(Referenced from 3.2.1.2 - Performance Requirements.)</p> <p><u>Security &amp; Availability Requirements</u></p> <p>The system shall ensure role-based access control (RBAC) to restrict access based on the assigned role, ensuring only authorized users can perform sensitive actions.</p> <p>All role assignments must be logged and auditable for security purposes.</p> <p>(Referenced from 3.2.1.3 - Availability &amp; Security.)</p> |
| Postconditions | <ul style="list-style-type: none"><li>- The user's role is successfully updated, and they now have access to the features authorized by their role.</li><li>- The administrator receives a confirmation that the role has been assigned.</li><li>- The user is restricted from accessing features not authorized for their role.</li></ul>   |

|                                  |   |
|----------------------------------|---|
| UC Name                          | <b><u>UC-03: Two-Factor Authentication and Account Lockout</u></b>  |
| Summary                          | This use case describes how the system implements Two-Factor Authentication(2FA) for all user logins to increase security and automatically locks accounts after five failed login attempts. Locked users will be notified via email and provided with recovery options to regain access. |
| Dependency                       | <b>UC-01:</b> Login authentication<br><b>UC-02:</b> Admin User Management<br><b>UC-04:</b> User account Recovery<br><b>UC-25:</b> Security Incident Reporting   |
| Actors                           | <b>Primary Actor:</b> Users<br><b>Secondary Actors:</b> System(enforces 2FA,locks accounts and sends recovery notifications)  |
| Preconditions                    | <ul style="list-style-type: none"><li>- The user is registered and logged in with valid credentials.</li><li>- The user has enabled the 2FA.</li><li>- The user has not reached the maximum failed logged attempts(5).</li></ul>  |
| Description of the Main Sequence | <ol style="list-style-type: none"><li>1. The users enter their login credentials.</li><li>2. The system validates the credentials. If the credentials are correct, the user prompted to complete 2FA.</li></ol>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | <ol style="list-style-type: none"> <li>3. The system sends a one-time passcode to the user via the selected 2FA way.</li> <li>4. The user enters the OTP. If correct the user is logged in.</li> <li>5. The system logs the successful login attempt for security auditing.</li> </ol>  |
| Description of the Alternative Sequence | <p><b><i>A1: Invalid Login Credentials:</i></b></p> <ol style="list-style-type: none"> <li>1. The users enters incorrect credentials.</li> <li>2. The system displays error.</li> <li>3. The system increments the failed attempts.</li> </ol> <p><b><i>A2: Failed 2FA:</i></b></p> <ol style="list-style-type: none"> <li>1. The user enters the wrong OTP.</li> <li>2. The system displays the error.</li> <li>3. Increments the counter.</li> </ol> <p><b><i>A3: User Reaches Maximum Failed Login Attempts</i></b></p> <ol style="list-style-type: none"> <li>1. If user fails 5 login attempts, the system will automatically lock the account.</li> <li>2. The system displays a message.</li> <li>3. The system sends a notification to the user by email for recovery.</li> <li>4. The system logs the failed login and locks the account for a time.</li> </ol>  |
| Non functional requirements             | <p><u>Usability Requirements</u><br/> The 2FA prompt should be clear and easy to understand for users.<br/> The lockout notification email should be easy to follow.<br/> The recovery step should be quick and efficient, allowing users to gain access within 5 minutes.(Referenced from 3.2.1.1-Usability Requirements)</p> <p><u>Performance Requirements</u><br/> The system should process login attempts and 2FA in less than 2 seconds to ensure minimal delay.<br/> The system should handle up to 10,000 concurrent login attempts and 2FA validation without performance degradation.(Referenced from 3.2.1.2-Performance Requirements)</p> <p><u>Security and Availability Requirements</u><br/> The system must ensure 2FA is enabled for all user logins, ensuring enhanced security.Account lockout should automatically trigger after 5 failed attempts and notify the user promptly.<br/> Sensitive data like passwords otps etc must be encrypted in transit and at rest t prevent unauthorized access.(Referenced from 3.2.1.3- Availability and Security)</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- If the user passes 2FA, they are successfully logged in and granted access based on their role.</li> </ul>   |

## Property Rental Management System Requirements Specification

|  |  |
|--|--|
|  | <ul style="list-style-type: none"><li>- If the user fails to pass 2FA or the login credentials, the system locks the account after 5 failed attempts and notifies the user of the lockout via email.</li><li>- The system logs all failed login attempts and ensures the account is locked for safety reasons.</li></ul> |
|--|--|

|   |  |
|---|--|
| UC Name                                 | <b><u>UC-04: User Account Recovery</u></b>   |
| Summary                                 | This use case describes the process for user account recovery through email verification. If a user forgets their password they can request a password reset by verifying their identity via email. The process may also include security questions or other ver methods to ensure the identity of the user.   |
| Dependency                              | <b>UC-01:</b> Login authentication<br><b>UC-02:</b> Admin User Management  |
| Actors                                  | <b>Primary Actor:</b> Users<br><b>Secondary Actors:</b> System(handles the password reset process)   |
| Preconditions                           | <ul style="list-style-type: none"><li>- The user must have a registered account in the system,</li><li>- The user must have access to the email they have used to register.</li><li>- The system must have the user's security questions or other 2FA set up.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The user selects 'Forgot Password' option on the login page.</li><li>2. The system asks the user's registered email address to start the recovery.</li><li>3. The user enters their email address and submits request.</li><li>4. The system sends an email to the user with a password reset link with expiration.</li><li>5. The user clicks the link which opens the new password page.</li><li>6. The system validates the new password.</li><li>7. Once the password is reset the system logs the user out from any active devices for security purposes.</li><li>8. The user is redirected to the login page.</li><li>9. The system logs the auction.</li></ol> |
| Description of the Alternative Sequence | <b>A1: Invalid Email Address</b> <ol style="list-style-type: none"><li>1. If the email is not registered, the system shows: "No account found. Please try again."</li><li>2. Prompts user to enter a valid email or contact support.</li></ol> <b>A2: Failed Email Delivery</b> <ol style="list-style-type: none"><li>1. If email delivery fails, the system shows: "Unable to send email. Please try again later."</li></ol>  |

## Property Rental Management System Requirements Specification

|                             |  |
|-----------------------------|--|
|                             | <p>2. Logs the failed attempt for troubleshooting.</p> <p><b>A3: Security Questions Verification</b></p> <ol style="list-style-type: none"> <li>1. User answers previously set security questions.</li> <li>2. If correct, password reset proceeds. If incorrect, prompts user to try again or contact support.</li> </ol> <p><b>A4: Expired Reset Link</b></p> <ol style="list-style-type: none"> <li>1. If the link is expired, the system shows: "This link has expired. Request a new one."</li> <li>2. User can request a new reset link.</li> </ol>  |
| Non functional requirements | <p><u>Usability Requirements</u></p> <p>Email notifications should include clear instructions for resetting the password and links that are easy to follow.</p> <p>The onboarding and recovery process should be designed to allow the user to recover their password within 5 minutes.</p> <p>(Referenced from 3.2.1.1 – Usability Requirements)</p> <p><u>Performance Requirements</u></p> <p>The system should send the password reset email within 2 seconds of the user's request.</p> <p>Password reset page loading should be completed within 2 seconds to avoid user frustration.(Referenced from 3.2.1.2 – Performance Requirements)</p> <p><u>Security &amp; Availability Requirements</u></p> <p>All password reset links must expire within a set time frame (e.g., 1 hour) to prevent unauthorized access.</p> <p>Sensitive data (such as new passwords) must be encrypted during transmission to protect it from interception.</p> <p>(Referenced from 3.2.1.3 – Availability &amp; Security)</p> |
| Postconditions              | <ul style="list-style-type: none"> <li>- If the user successfully completes the recovery process,their password is reset.</li> <li>- The system logs the recovery attempt for security auditing and analysis.</li> <li>- If the recovery attempt fails the system ensures the user is notified and given the option to retry the process or contact support.</li> </ul>  |

|            |   |
|------------|---|
| UC Name    | <b><u>UC-05: Legal Verification for User Registration</u></b>   |
| Summary    | This use case describes the process in which the legal office verifies the identity of users (tenants, landlords and property managers) before allowing them to register a new account in the system. |
| Dependency | <b>UC-41:</b> Document Upload process   |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | <b>Extends into UC-07:</b> Additional Landlord Verification  |
| Actors                                  | <b>Primary:</b> Legal Office<br><b>Secondary:</b> Tenants, Landlords, Property Managers  |
| Preconditions                           | <ul style="list-style-type: none"><li>- User has submitted a registration request.</li><li>- User has uploaded required identification and supporting documentation.</li></ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. User submits a registration request and uploads identification documents.</li><li>2. System alerts the Legal Office of a pending verification.</li><li>3. Legal Office reviews the documents.</li><li>4. Legal Office confirms user identity.</li><li>5. For landlords, verification process is triggered (via UC07).</li><li>6. If verified, system marks user as approved and enables account creation.</li><li>7. User receives a verification success notification.</li></ol> |
| Description of the Alternative Sequence | <p><b>A1. Missing or Invalid Documents:</b></p> <ol style="list-style-type: none"><li>1. The Legal Office marks the request as incomplete.</li><li>2. System notifies the user to upload missing or the right documents.</li></ol> <p><b>A2. Failed Verification:</b></p> <ol style="list-style-type: none"><li>1. The Legal Office marks the request as rejected.</li><li>2. System notifies the user with a rejection reason and logs the attempt.</li></ol>   |
| Non functional requirements             | The system shall securely handle user data in line with GDPR standards during verification, with documents encrypted both in transit and at rest. It shall integrate with national ID systems (e.g., e-Albania), provide a user-friendly, responsive interface, log all actions, and remain available and efficient during peak usage.   |
| Postconditions                          | <ul style="list-style-type: none"><li>- Verified users can proceed to complete registration and log in.</li><li>- Rejected or incomplete users remain blocked from account creation until the issue is resolved.</li></ul>   |



## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| UC Name                                 | <b><u>UC06 - Account Suspension &amp; Deactivation</u></b>  |
| Summary                                 | This use case describes the process in which the administrator suspends or deactivates user accounts that violate platform policies to maintain system security and compliance.   |
| Dependency                              | <b>UC-01:</b> User management features like role-based access and login   |
| Actors                                  | <b>Primary:</b> Administrator<br><b>Secondary:</b> Affected users (tenants, landlords, etc.)  |
| Preconditions                           | <ul style="list-style-type: none"><li>- The user account exists in the system.</li><li>- The violation or reason for suspension has been recorded or reported.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. Administrator identifies or receives a report of a user violation.</li><li>2. Administrator accesses the user management interface.</li><li>3. Administrator selects the account.</li><li>4. Administrator applies suspension or deactivation.</li><li>5. System logs the action and updates the user's account status.</li><li>6. User receives a notification about the action taken.</li></ol>    |
| Description of the Alternative Sequence | <p><b>A1. Mistaken Suspension:</b></p> <ol style="list-style-type: none"><li>1. If the account was suspended in error, the administrator reactivated the account.</li><li>2. The system logs the change and notifies the user.</li></ol> <p><b>A2. Appeal Process (Future Scope):</b></p> <ol style="list-style-type: none"><li>1. User submits an appeal request.</li><li>2. Admin reviews and restores or upholds the suspension.</li></ol> |
| Non functional requirements             | The system shall ensure secure and audible handling of account status changes. All actions must be logged and affected users notified. The admin interface shall remain responsive and available during peak usage.   |
| Postconditions                          | <ul style="list-style-type: none"><li>- The user account is marked as suspended or deactivated.</li><li>- Access to the system is blocked for the suspended/deactivated user.</li><li>- Logs are available for future auditing.</li></ul>   |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| UC Name                                 | <b><u>UC 07 - Landlord Account Verification</u></b>  |
| Summary                                 | The system provides a verification process for <b>landlords</b> to ensure they are authorized property owners before they can list properties. Landlords must submit proof of property ownership and identification for review.  |
| Dependency                              | <b>UC-01: Secure User Registration</b> → A landlord must first create an account.<br><b>UC-02: Role-Based Access Control</b> → Only users with the <b>landlord role</b> can proceed with verification.   |
| Actors                                  | <b>Primary actor:</b> Legal Office<br><b>Secondary actor:</b> System   |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The landlord <b>must have created an account</b></li> <li>- The account <b>must be assigned to the “landlord” role</b></li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The landlord <b>logs into the system</b>.</li> <li>2. The landlord navigates to the <b>“Verification”</b> section.</li> <li>3. The landlord <b>uploads ownership and identity documents</b>.</li> <li>4. The system <b>submits the documents</b> for review.</li> <li>5. The <b>legal office reviews and approves/rejects</b> the verification request.</li> <li>6. The landlord is <b>notified of the verification outcome</b></li> </ol> |
| Description of the Alternative Sequence | <b>A1:</b> If the <b>uploaded documents are invalid</b> , the system <b>rejects the request</b> and prompts the landlord to <b>resubmit valid documents</b> .<br><br><b>A2:</b> If the <b>legal office does not approve within 48 hours</b> , the system <b>sends a reminder notification</b> .  |
| Non functional requirements             | <u>Security &amp; Privacy:</u> Only authorized legal personnel can <b>review documents</b> .<br><br><u>Availability:</u> Verification requests must be <b>processed within 48 hours</b> to avoid delays.   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- <b>Approved landlords</b> gain access to the platform’s property listing features.</li> <li>- <b>Rejected landlords</b> must resubmit <b>valid documentation</b> for verification.</li> </ul>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| UC Name                                 | <b><u>UC 08 - Event Calendar Management</u></b>   |
| Summary                                 | The system provides an <b>event calendar</b> for <b>tenants and landlords</b> to <b>schedule property viewings, maintenance appointments, and community events</b> .  |
| Dependency                              | <b>UC-01: Secure User Registration</b><br><b>UC-05: Property Viewing Scheduling</b>   |
| Actors                                  | <b>Primary Actor:</b> Tenant, Landlord<br><b>Secondary Actor:</b> System  |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The <b>user must have registered and be logged in</b>.</li> <li>- The <b>event scheduling module must be active</b>.</li> <li>- If scheduling a <b>property viewing</b>, there must be an <b>existing viewing request</b>.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The user logs in.</li> <li>2. The user navigates to the <b>Event Calendar</b> section.</li> <li>3. The user selects a <b>date and time</b> for an event.</li> <li>4. The system <b>validates and saves</b> the event.</li> <li>5. The system <b>sends reminders</b> to relevant users.</li> </ol> |
| Description of the Alternative Sequence | <p><b>A1:</b> If the <b>event date conflicts with another scheduled event</b>, the system <b>suggests alternative time slots</b>.</p> <p><b>A2:</b> If the <b>event is canceled</b>, the system <b>removes the event and notifies attendees</b>.</p>  |
| Non functional requirements             | <p><u>Usability:</u> The system must have a <b>clear and user-friendly interface</b>.</p> <p><u>Security:</u> Event changes must be <b>logged to prevent disputes</b>.</p> <p><u>Performance:</u> The calendar must be <b>updated in real-time</b> to avoid conflicts.</p>  |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The event is <b>scheduled successfully and notifications are sent</b>.</li> <li>- If <b>canceled</b>, the event is <b>removed from the system</b>.</li> </ul>  |

|            |  |
|------------|--|
| UC Name    | <b><u>UC 09 - Rental Property Listing Management</u></b>   |
| Summary    | The system allows <b>landlords</b> to <b>list, edit, and remove rental properties</b> while ensuring all listings meet system requirements.  |
| Dependency | <p><b>UC-07: Landlord Account Verification</b> → Only <b>verified landlords</b> can list properties.</p> <p><b>UC-10: Rental Application Submission</b> → Listings must be linked to</p> |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | rental applications.  |
| Actors                                  | <b>Primary Actor:</b> Landlord<br><b>Secondary Actor:</b> System  |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The <b>landlord must have a verified account</b>.</li> <li>- The landlord must be logged in.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The landlord <b>logs in</b> to the system.</li> <li>2. The landlord navigates to “<b>Property Management</b>”.</li> <li>3. The landlord <b>adds/edit property details</b>.</li> <li>4. If the <b>property is newly listed</b>, the system <b>validates verification status</b>.</li> <li>5. The system <b>saves the listing</b> and updates the rental application database.</li> </ol> |
| Description of the Alternative Sequence | <p><b>A1:</b> If <b>property verification is missing (UC-07)</b>, the system <b>denies listing submission</b>.</p> <p><b>A2:</b> If <b>mandatory fields are missing</b>, the system <b>prompts the landlord to fill them</b>.</p>   |
| Non functional requirements             | <p><u>Performance:</u> Listings must be <b>approved within 2 hours</b>.</p> <p><u>Security:</u> Unauthorized users <b>cannot modify or delete</b> listings.</p>   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The <b>property is successfully listed</b>.</li> <li>- The listing is <b>visible to potential tenants</b>.</li> <li>- If <b>submission fails</b>, the landlord must correct errors and retry.</li> </ul>   |

|               |  |
|---------------|--|
| UC Name       | <b><u>UC 10 - Error Handling in Data Processing</u></b>  |
| Summary       | The system detects, logs, and manages <b>data processing errors</b> to <b>prevent system failures</b> .  |
| Dependency    | <p><b>UC-09: Rental Property Listing Management</b> → Errors may occur when managing property listings.</p> <p><b>UC-12: Property Availability Updates</b> → Availability updates must be processed correctly to avoid system inconsistencies.</p> |
| Actors        | <b>Primary Actor:</b> System<br><b>Secondary Actor:</b> Administrator  |
| Preconditions | <ul style="list-style-type: none"> <li>- The system is actively processing data related to property listings or availability updates.</li> </ul>   |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The system <b>detects an error</b>.</li> <li>2. If the error is in <b>rental property listing (UC-09)</b>, the system <b>rolls back changes</b>.</li> <li>3. If the error is in <b>availability updates (UC-12)</b>, the system <b>prevents duplicate or conflicting updates</b>.</li> <li>4. The system <b>logs the error</b> and attempts <b>automatic recovery</b>.</li> <li>5. If recovery <b>fails</b>, the administrator is notified.</li> </ol>     |
| Description of the Alternative Sequence | <p><b>A1:</b> If an <b>API failure occurs</b>, the system <b>retries up to 3 times before sending an alert</b>.</p> <p><b>A2:</b> If the <b>database connection is lost</b>, the system <b>queues pending transactions</b> and retries every 5 minutes.</p> <p><b>A3:</b> If a <b>user submits incorrect data</b>, the system <b>displays a validation error message and prompts for correction</b>.</p>   |
| Non functional requirements             | <p><u>Performance:</u> The system must <b>recover from errors within 30 seconds</b> whenever possible.</p> <p><u>Security:</u> Error logs must be <b>encrypted and stored securely</b> to prevent unauthorized access.</p> <p><u>Maintainability:</u> The system should <b>generate detailed error logs</b> for easier debugging and support.</p> <p><u>Availability:</u> The system must have a <b>99.9% uptime guarantee</b>, meaning errors should not disrupt services for extended periods.</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- When the error occurs, the system <b>recovers automatically or alerts an administrator</b>.</li> <li>- If successful, <b>data integrity is maintained</b>.</li> <li>- If unsuccessful, <b>administrators must manually intervene</b>.</li> </ul>  |

|                    |   |
|--------------------|---|
| UC Name            | <b><u>UC - 11: Maintenance and Repair Management</u></b>  |
| Summary            | The system allows <b>tenants</b> to <b>submit maintenance requests</b> , which are managed by <b>landlords or property managers</b> . |
| Dependency         | <b>UC-09: Rental Property Listing Management</b> → Only active rental properties can have maintenance requests.                       |
| Actors             | <b>Primary Actor:</b> Tenant<br><b>Secondary Actor:</b> Landlord, System  |
| Preconditions      | <ul style="list-style-type: none"> <li>- The tenant must <b>have an active rental agreement</b>.</li> </ul>                           |
| Description of the | <ol style="list-style-type: none"> <li>1. The tenant logs in.</li> </ol>  |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| Main Sequence                           | <ol style="list-style-type: none"> <li>2. The tenant selects “<b>Maintenance Requests</b>”.</li> <li>3. The tenant <b>submits a request</b>, selecting the <b>associated rental property (UC-09)</b>.</li> <li>4. The landlord assigns a <b>technician</b>.</li> <li>5. The system <b>logs and tracks progress and requests until completion</b>.</li> </ol>   |
| Description of the Alternative Sequence | <p><b>A1:</b> If the <b>landlord does not respond within 24 hours</b>, the system <b>sends an escalation alert</b>.</p> <p><b>A2:</b> If the <b>assigned technician is unavailable</b>, the system <b>suggests an alternative technician</b>.</p> <p><b>A3:</b> If the <b>tenant submits an incomplete request</b>, the system <b>prompts them to provide missing details</b> before submitting.</p> <p><b>A4:</b> If the repair requires additional approvals (e.g., major damages), the system <b>flags the request for manual review by the landlord</b>.</p> |
| Non functional requirements             | <p><u>Availability:</u> Maintenance requests must be <b>acknowledged within 24 hours</b>.</p> <p><u>Usability:</u> The maintenance request form must be <b>simple and easy to fill out</b>.</p>  |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The <b>maintenance request is resolved</b>, and the tenant is notified.</li> <li>- If the issue remains unresolved, the request <b>remains open</b> for further action.</li> </ul>  |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><u>UC - 12: Property Availability Updates</u></b>   |
| Summary                          | The system detects, logs, and manages <b>data processing errors</b> to <b>prevent system failures</b> .  |
| Dependency                       | <b>UC-09: Rental Property Listing Management</b> → Availability updates can only be made on <b>listed properties</b> .   |
| Actors                           | <b>Primary Actor:</b> System<br><b>Secondary Actor:</b> Administrator  |
| Preconditions                    | <ul style="list-style-type: none"> <li>- The <b>property must be an active rental listing (UC-09)</b>.</li> <li>- The <b>landlord must be logged in and verified</b>.</li> </ul>   |
| Description of the Main Sequence | <ol style="list-style-type: none"> <li>1. The landlord <b>logs in</b>.</li> <li>2. The landlord navigates to “<b>My Properties</b>”.</li> <li>3. The landlord selects a <b>property and updates its availability</b>.</li> <li>4. The system <b>validates the update and prevents conflicts</b> (linked</li> </ol> |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | to <b>UC-10: Error Handling</b> ).<br>5. The system <b>notifies interested tenants</b> .   |
| Description of the Alternative Sequence | <p><b>A1:</b> If the <b>property is not listed (UC-09)</b>, the system <b>prevents availability updates</b>.</p> <p><b>A2:</b> If the <b>landlord tries to mark a rented property as available</b>, the system <b>prompts for lease termination confirmation</b>.</p> <p><b>A3:</b> If there is a <b>server delay</b>, the system <b>temporarily stores the update and applies it when the connection is restored</b>.</p> |
| Non functional requirements             | <p><u>Performance:</u> Updates must reflect <b>instantly</b>.</p> <p><u>Data Consistency:</u> Updates should be <b>synchronized across all rental platforms</b> (if integrated).</p> <p><u>Security:</u> Only <b>verified landlords</b> should be able to update availability.</p> <p><u>Usability:</u> The update process should be <b>quick and require minimal steps</b>.</p>   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The <b>updated availability status is reflected</b> in the system.</li> <li>- If a listing update fails, the landlord receives a <b>notification with troubleshooting steps</b></li> </ul>  |

|                                  |   |
|----------------------------------|---|
| UC Name                          | <b><i>UC-13: Property Managers Scheduling Viewings</i></b>  |
| Summary                          | Property managers can schedule and manage property viewings on behalf of the landlord if permitted, ensuring an organized scheduling process.   |
| Dependency                       | This use case depends on the landlord granting permission for property managers to manage viewings.   |
| Actors                           | Property Manager, Landlord  |
| Preconditions                    | <ul style="list-style-type: none"> <li>- The property must be listed for rent.</li> <li>- The landlord must grant permission for property managers to handle viewings.</li> </ul>   |
| Description of the Main Sequence | <ol style="list-style-type: none"> <li>1. Property manager logs into the system.</li> <li>2. Property manager selects a listed property.</li> <li>3. Property manager checks the availability of viewing slots.</li> <li>4. Property manager schedules a viewing for potential tenants.</li> <li>5. System notifies the landlord and potential tenants of the scheduled viewing.</li> </ol> |
| Description of the Alternative   | <ol style="list-style-type: none"> <li>1. Property manager attempts to schedule a viewing.</li> <li>2. System verifies if the landlord has granted permission.</li> </ol>   |

## Property Rental Management System Requirements Specification

|                             |   |
|-----------------------------|---|
| Sequence                    | <ol style="list-style-type: none"><li>3. If permission is not granted, the system prevents scheduling and notifies the property manager.</li><li>4. If the selected time slot is unavailable, the system suggests alternative slots.</li><li>5. If the property manager cancels a scheduled viewing, the system notifies the landlord and tenants.</li></ol>  |
| Non functional requirements | <p><u>Logging number 5:</u> Any changes made to user roles ( assigning admin privileges or changing user permissions) must be logged, including who performed the action, what change was made, and the timestamp.</p> <p><u>Logging number 6:</u> Any changes to access control settings (permission modifications, new role creations) should be logged, including who made the change, what was changed, and when.</p> |
| Postconditions              | <ul style="list-style-type: none"><li>- The property viewing is scheduled successfully.</li><li>- All involved parties (landlord, property manager, and potential tenants) receive notifications.</li><li>- The system updates the schedule to reflect the new booking.</li></ul>   |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><i><u>UC - 14: Tenant Behavior Insights for Landlords</u></i></b>   |
| Summary                          | The system allows landlords or property managers (if given permission) to view summarized insights on tenant behavior, such as payment timeliness, maintenance request frequency, and feedback history.  |
| Dependency                       | <b>UC-01:</b> Secure Login & Role-Based Access Control<br><b>UC-02:</b> User Role Access Management<br><b>UC-19:</b> Tenant Rental History Report<br><b>UC-23:</b> Platform Activity & Security Reports  |
| Actors                           | Landlord<br>Property Manager (with permission)   |
| Preconditions                    | <ul style="list-style-type: none"><li>- The landlord or property manager must have an active account.</li><li>- The landlord/property manager must have permission to access tenant insights.</li><li>- Tenant data must be available in the system.</li></ul>   |
| Description of the Main Sequence | <ol style="list-style-type: none"><li>1. The landlord/property manager logs into the system.</li><li>2. They navigate to the "Tenant Insights" section.</li><li>3. The system retrieves tenant behavior data from the database.</li><li>4. A summary of key metrics (payment timeliness, maintenance requests, feedback history) is displayed.</li><li>5. The landlord/property manager reviews the insights to make rental decisions.</li></ol> |



## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. If the user does not have permission, an error message is displayed.</li> <li>2. If no tenant data is available, the system notifies the user that no records exist.</li> <li>3. If the tenant's data is incomplete or inconsistent, a partial or incomplete summary of insights is displayed.</li> </ol>  |
| Non functional requirements             | <p><u>Usability number 1:</u> The system shall have an intuitive and user-friendly interface to accommodate users with varying technical expertise.</p> <p><u>Performance number 3:</u> System response time for database queries shall not exceed 300 ms under normal load conditions.</p> <p><u>Availability&amp;Security number 12:</u> The system must ensure that sensitive data (user payment details, personal information) is only accessible to authorized personnel with the correct role permissions.</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The landlord/property manager has viewed the tenant behavior summary.</li> <li>- The system logs the access request for security and auditing purposes.</li> </ul>  |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><u>UC - 15: Rental Property Search Filters</u></b>  |
| Summary                          | Tenants can search for rental properties using various filters such as price range, location, number of rooms, and amenities, allowing them to efficiently find suitable properties that meet their specific needs. This makes choices easier and more customized.   |
| Dependency                       | <b>UC-0:</b> Secure Login & Role-Based Access Control<br><b>UC-02:</b> User Role Access Management<br><b>UC-13:</b> Property Managers Scheduling Viewings<br><b>UC-23:</b> Platform Activity & Security Reports  |
| Actors                           | Tenant   |
| Preconditions                    | <ul style="list-style-type: none"> <li>- The tenant must have an active account and be logged into the system.</li> <li>- Rental properties must be available in the system.</li> <li>- Property details (price, location, number of rooms, amenities) must be accurately stored in the database.</li> </ul> |
| Description of the Main Sequence | <ol style="list-style-type: none"> <li>1. The tenant logs into the system.</li> <li>2. The tenant navigates to the property search section.</li> <li>3. The system displays a search interface with multiple filter options (price, location, rooms, amenities).</li> </ol>                                  |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | <ol style="list-style-type: none"> <li>4. The tenant selects their desired filters.</li> <li>5. The system retrieves matching properties based on the selected filters.</li> <li>6. The system displays the filtered list of properties to the tenant.</li> <li>7. The tenant reviews the list of properties and makes a decision.</li> </ol>  |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. If no properties match the selected filters, the system notifies the tenant and suggests modifying the search criteria.</li> <li>2. If the system fails to retrieve property data, the tenant is informed of the issue and is prompted to try again.</li> </ol>  |
| Non functional requirements             | <p><u>Usability number 2:</u> The interface shall follow Material Design principles to ensure consistency, clarity, and ease of use.</p> <p><u>Availability number 4:</u> The system should allow users to set alerts for specific property types or location preferences, sending notifications for new listings as they become available.</p> <p><u>Availability number 3:</u> The system must automatically update property availability when a property is booked or its price changes, ensuring that users see real-time updates.</p> <p><u>Security number 2:</u> The system must support at least 500 concurrent users for booking, payment, and search activities, without delays or system lag.</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The tenant views the list of rental properties that meet their search criteria.</li> <li>- The system logs the search request for future recommendations and improvements.</li> </ul>   |

|            |  |
|------------|--|
| UC Name    | <b><u>BR - 16: Rental Application &amp; Payment Validation</u></b>   |
| Summary    | Tenants shall be able to submit a rental application for a property by providing necessary documents and personal information. The process defines tenant and finance office responsibilities, ensuring a streamlined rental process and proper payment validation.  |
| Dependency | <b>UC-05:</b> Legal Verification for User Registration<br><b>UC-21:</b> Secure Tenant Payment Processing & Receipts<br><b>UC-22:</b> Automatic Late Fee Penalties<br><b>UC-25:</b> Security Incident Reporting<br><b>UC-28:</b> Secure Tenant-Landlord Messaging<br><b>UC-17:</b> Landlord Control Over Applications |
| Actors     | Tenant, Finance Office   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| Preconditions                           | <ul style="list-style-type: none"><li>- The tenant must have an active account and be logged into the system.</li><li>- The rental property must be available and open for applications.</li><li>- The tenant must have access to the necessary documents for the application.</li><li>- The finance office must have access to the payment system to validate payments.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The tenant logs into the system and navigates to the rental property page.</li><li>2. The tenant selects the property and begins the application process.</li><li>3. The system prompts the tenant to submit necessary documents (e.g., identification, proof of income, references).</li><li>4. The tenant submits the required documents and personal information.</li><li>5. The system validates the information and prepares the application.</li><li>6. The system requests the first payment (e.g., security deposit, first month's rent).</li><li>7. The tenant processes the payment through the system.</li><li>8. The finance office receives the payment for validation.</li><li>9. Once validated, the system confirms the application submission and payment receipt.</li><li>10. The tenant receives confirmation of application acceptance and payment validation.</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the tenant does not provide the necessary documents, the system prompts the tenant to upload missing documents.</li><li>2. If the payment fails, the system notifies the tenant and requests an alternative payment method.</li><li>3. If the tenant does not meet the property's requirements (e.g., income verification fails), the system notifies the tenant that their application has been rejected.</li></ol>  |
| Non functional requirements             | <p><u>Usability number 1:</u> The system shall have an intuitive and user-friendly interface to accommodate users with varying technical expertise.</p> <p><u>Performance number 2:</u> 95% of transactions shall be processed within 2 seconds, ensuring smooth user experience.</p> <p><u>Availability number 3:</u> The system must automatically update property availability when a property is booked or its price changes, ensuring that users see real-time updates.</p>  |

## Property Rental Management System Requirements Specification

|                |  |
|----------------|--|
|                | <p><u>Availability number 2:</u> The system must support at least 500 concurrent users for booking, payment, and search activities, without delays or system lag.</p> <p><u>Security number 13:</u> All sensitive data, including payment information and user details, must be encrypted both in transit and at rest to protect it from unauthorized access.</p> <p><u>Fault Tolerance Mechanisms number 1:</u> In case of server failure, traffic shall be redirected to backup servers to ensure continuous operation.</p> <p><u>Payment Gateways number 2&amp;3:</u> -Bank Transfers (Raiffeisen, Credins, BKT, OTP, Intesa Sanpaolo) via local banking APIs.<br/>-E-Wallets (Easypay, Paysera) for users preferring digital payments.</p> |
| Postconditions | <ul style="list-style-type: none"> <li>- The tenant's rental application is submitted with the required documents.</li> <li>- The payment is processed and validated by the finance office.</li> <li>- The system logs the transaction and application details.</li> </ul>   |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><u>UC - 17: Landlord Control Over Applications</u></b>  |
| Summary                          | Landlords shall be able to review, approve, or reject rental applications submitted by tenants. They may also delegate this responsibility to assigned Property Managers via the system.   |
| Dependency                       | <b>UC-01:</b> Secure Login & Role-Based Access Control<br><b>UC-02:</b> User Role Access Management<br><b>UC-13:</b> Property Managers Scheduling Viewings<br><b>UC-23:</b> Platform Activity & Security Reports   |
| Actors                           | Landlords, Property Managers, Tenants  |
| Preconditions                    | <ul style="list-style-type: none"> <li>- The landlord must be verified in the system.</li> <li>- The rental application must be submitted and validated.</li> <li>- The property must be listed and available for rent.</li> <li>- If a Property Manager is handling approvals, delegation settings must be configured.</li> </ul> |
| Description of the Main Sequence | <ol style="list-style-type: none"> <li>1. The landlord/property manager logs into the system.</li> <li>2. They navigate to the "Rental Applications" section.</li> <li>3. The system displays a list of pending applications.</li> </ol>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | <ol style="list-style-type: none"> <li>4. The landlord/property manager selects an application to review.</li> <li>5. The system retrieves and displays tenant details, rental history, and application status.</li> <li>6. The landlord/property manager approves or rejects the application.</li> <li>7. The system updates the application status and notifies the tenant.</li> </ol>  |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. If there are no applications, the system notifies the landlord/property manager.</li> <li>2. If the landlord has delegated authority, the system ensures the property manager has the correct permissions before allowing access.</li> </ol>  |
| Non functional requirements             | <p><u>Security number 14:</u> The system must log the IP address of users during login attempts and other sensitive actions ( password changes, account settings modifications).</p> <p><u>Usability number 1:</u> The system shall have an intuitive and user-friendly interface to accommodate users with varying technical expertise.</p> <p><u>Security number 7:</u> The system must use two-factor authentication (2FA) for users when logging in. This means users will need to enter both their password and a one-time passcode (OTP) sent via email or -SMS.</p> <p><u>Availability number 1:</u> The system shall maintain 99.9% uptime, ensuring that the platform is accessible 24/7 for property searches, bookings, payments, and other activities.</p> <p><u>Logging number 3:</u> The system must log all user session activities, including the start time, end time, and actions performed during the session.</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The rental application status is updated (approved/rejected).</li> <li>- The tenant is notified of the decision.</li> <li>- The system logs the decision for record-keeping.</li> </ul>  |

|         |   |
|---------|---|
| UC Name | <b><u>UC - 18: Lease Renewal Requests</u></b>   |
| Summary | Tenants shall be able to request lease renewals within 30 days of expiration through the system, and receive confirmation or rejection from the landlord or property manager. This allows tenants to express interest in staying in the same property and helps landlords easily review and |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | respond to renewal requests.   |
| Dependency                              | <b>UC-05:</b> Legal Verification for User Registration<br><b>UC-06:</b> Account Suspension & Deactivation<br><b>UC-19:</b> Tenant Rental History Report<br><b>UC-21:</b> Secure Tenant Payment Processing & Receipts<br><b>UC-31:</b> Multi-Language Support   |
| Actors                                  | Tenant, Landlord, Property Manager   |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The tenant must have an active lease agreement that is nearing expiration.</li> <li>- The tenant must be logged into the system.</li> <li>- The landlord or property manager must be able to review the renewal request.</li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The tenant logs into the system.</li> <li>2. The tenant navigates to the "Lease Renewal" section.</li> <li>3. The tenant submits a lease renewal request.</li> <li>4. The system notifies the landlord or property manager of the request.</li> <li>5. The landlord/property manager reviews the renewal request and decides to approve or reject it.</li> <li>6. The system notifies the tenant of the approval or rejection of the renewal request.</li> </ol>   |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. If the tenant submits the renewal request after the lease expiration, the system notifies them that the request is no longer valid.</li> <li>2. If the landlord or property manager has not responded to the renewal request within a specified time frame, the system sends a reminder notification.</li> </ol>   |
| Non functional requirements             | <p><u>Performance number 2&amp;3:</u> The system shall process lease renewal requests within 2 seconds. System response time for database queries related to lease renewals shall not exceed 300 ms under normal load conditions.</p> <p><u>Availability number 1:</u> The system shall maintain 99.9% uptime, ensuring uninterrupted access for tenants to submit renewal requests and for landlords or property managers to respond to them.</p> <p><u>Testing&amp;Reliability number 2:</u> The system shall be capable of handling peak traffic loads without performance degradation.</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The tenant receives a confirmation or rejection of the lease renewal.</li> </ul>  |

## Property Rental Management System Requirements Specification

|  |   |
|--|---|
|  | <ul style="list-style-type: none"><li>- The system updates the lease status based on the decision (renewed or terminated).</li><li>- The system logs the renewal request and response for record-keeping.</li></ul> |
|--|---|

|   |  |
|---|--|
| UC Name                                 | <b><u>UC - 19: View Rental History Report</u></b>  |
| Summary                                 | This use case describes the process in which tenants can view a detailed rental history report, including a list of past properties rented, payment records, and feedback from landlords.  |
| Dependency                              | <b>UC-01:</b> Secure Login & Role-Based Access Control<br><b>UC-02:</b> User Role Access Management<br><b>UC-15:</b> Rental Property Search Filters<br><b>UC-21:</b> Secure Tenant Payment Processing & Receipts<br><b>UC-30:</b> Tenant Feedback & Ratings  |
| Actors                                  | <b>Primary Actor:</b> Tenant<br><b>Secondary Actors:</b> System, Landlord  |
| Preconditions                           | <ul style="list-style-type: none"><li>- The tenant must be logged into the system.</li><li>- The tenant has a history of past rentals.</li><li>- The tenant has access to the rental history feature.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The tenant logs into the system using secure credentials.</li><li>2. The tenant navigates to the "Rental History" section in their profile.</li><li>3. The system authenticates the tenant's access based on their role and retrieves past rental information, including properties rented, payment history, and landlord feedback.</li><li>4. The system displays the rental history report to the tenant, which includes a list of properties, payment records, and feedback from landlords.</li><li>5. The tenant can view the full details of each record, including payment dates and amounts, as well as feedback from landlords.</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the tenant has no rental history, the system will display a message indicating no history available.</li><li>2. If the tenant is not logged in, they will be redirected to the login page.</li></ol>   |
| Non functional requirements             | <u>Usability Requirements</u><br>The rental history report interface must be intuitive and easy to navigate. It should clearly display rental properties, payments, and feedback. Search and filter should be responsive for quick record retrieval.   |

## Property Rental Management System Requirements Specification

|                |  |
|----------------|--|
|                | <p>Tenants should be able to download or print the report.</p> <p><u>Performance Requirements</u><br/>Rental history data should load within 2 seconds.<br/>The system must handle up to 10,000 tenant records and 100,000 properties efficiently.</p> <p><u>Availability &amp; Security Requirements</u><br/>Tenant data, including rental history and payments, must be securely encrypted.<br/>Only authorized tenants can access their rental history reports.</p> |
| Postconditions | <ul style="list-style-type: none"><li>- The tenant has successfully viewed their rental history, including past properties rented, payment records, and landlord feedback.</li><li>- If the tenant has no rental history or feedback, the system provides an appropriate message.</li><li>- The system logs the action of the tenant viewing their rental history for security auditing purposes.</li></ul>  |

|                         |  |
|-------------------------|--|
| UC Name                 | <b><i>UC - 20: Landlord Custom Lease Terms and Conditions</i></b>  |
| Summary                 | This use case describes how landlords can define and set custom lease terms and conditions for each rental property. The custom terms may include pet policies, security deposit amounts, and maintenance responsibilities. Tenants must review, acknowledge, and agree to these terms before submitting a rental application.   |
| Dependency              | <b>UC-01:</b> Secure Login & Role-Based Access Control<br><b>UC-10:</b> Rental Application Submission  |
| Actors                  | <b>Primary Actor:</b> Landlord<br><b>Secondary Actor:</b> Tenant (interacts indirectly when submitting a rental application)<br><b>Supporting Actor:</b> System (stores lease terms and handles tenant acknowledgment)   |
| Preconditions           | <ul style="list-style-type: none"><li>- The landlord is logged in and has access to the property management interface.</li><li>- The tenant is registered and has access to the rental application page.</li><li>- The landlord has at least one rental property listed on the platform.</li><li>- The system must have a valid lease template and the ability to save custom terms for each property.</li></ul> |
| Description of the Main | 1. <b>Landlord logs in</b> with their credentials and accesses the <b>landlord dashboard</b> .   |



## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| Sequence                                | <ol style="list-style-type: none"> <li>2. The <b>landlord selects a property</b> and views current lease terms, with options to <b>edit or add new terms</b>.</li> <li>3. The landlord clicks “<b>Set Lease Terms</b>”, fills out the <b>form</b> with pet policy, security deposit, and maintenance responsibilities, and clicks <b>Save</b>.</li> <li>4. The system <b>saves the terms</b> and confirms the update.</li> <li>5. When a <b>tenant views the property</b>, they must <b>acknowledge</b> the lease terms before submitting the <b>rental application</b>.</li> <li>6. The tenant <b>reviews and agrees</b> to the terms by clicking “<b>I Agree</b>”.</li> <li>7. The system <b>records the agreement</b> and allows the tenant to proceed with the application.</li> </ol> |
| Description of the Alternative Sequence | <p><b>A1: Landlord Does Not Set Custom Lease Terms</b><br/>The landlord skips setting custom terms, and the system defaults to a standard lease template. The landlord is notified that terms are not set. Tenants viewing the property see the standard lease and proceed as usual.</p> <p><b>A2: Tenant Does Not Agree to Lease Terms</b><br/>The tenant reviews the terms but does not agree. The system prevents submission of the rental application until the tenant clicks “I Agree,” reminding them to acknowledge the terms.</p>  |
| Non functional requirements             | <p><u>Usability Requirements</u><br/>The landlord interface should be intuitive, allowing landlords to easily set and update lease terms for properties.<br/>(Referenced from 3.2.1.1 - Usability Requirements)</p> <p><u>Performance Requirements</u><br/>The system must update and save custom lease terms within 2 seconds to ensure efficiency.<br/>(Referenced from 3.2.1.2 - Performance Requirements)</p> <p><u>Security &amp; Availability Requirements</u><br/>The system shall ensure that lease terms are stored securely and are accessed only by authorized users (landlords and tenants).<br/>(Referenced from 3.2.1.3 - Availability &amp; Security)</p>   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- Custom lease terms for the property are successfully set and saved in the system.</li> <li>- The tenant acknowledges and agrees to the terms before proceeding with their rental application.</li> <li>- The tenant’s agreement is stored in the system, linked to their rental application.</li> </ul>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| UC Name                                 | <b><u>UC - 21: Secure Payment Processing and Receipt Generation</u></b>   |
| Summary                                 | This use case describes how the system processes tenant payments securely and generates electronic receipts upon successful payment.  |
| Dependency                              | <b>UC-16:</b> Submitting a rental application<br><b>UC-17:</b> Approving/rejecting rental applications  |
| Actors                                  | <b>Primary Actor:</b> Tenant<br><b>Secondary Actor:</b> System (handles payment processing)   |
| Preconditions                           | <ul style="list-style-type: none"><li>- The tenant has an active rental agreement.</li><li>- The tenant has a valid payment method linked.</li><li>- The system is connected to the payment gateway.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The tenant navigates to the payment section and selects the due payment.</li><li>2. The system displays available payment methods.</li><li>3. The tenant selects a payment method and confirms the transaction.</li><li>4. The system securely processes the payment through the payment gateway.</li><li>5. Upon successful transaction, the system generates an electronic receipt.</li><li>6. The receipt is sent via email and stored in the tenant's payment history.</li></ol> |
| Description of the Alternative Sequence | <b>Payment Failure</b> <ol style="list-style-type: none"><li>1. The tenant's payment fails due to insufficient funds or a declined card.</li><li>2. The system displays an error message and suggests an alternative payment method.</li><li>3. The tenant attempts another payment method.</li></ol>   |
| Nonfunctional requirements              | <p><u>Availability &amp; Security (3.2.1.3)</u> - The system must ensure that sensitive data (user payment details, personal information) is only accessible to authorized personnel.</p> <p><u>Performance (3.2.1.2)</u> - 95% of transactions shall be processed within 2 seconds.</p> <p><u>Reliability &amp; Fault Tolerance (3.2.1.5)</u> - The system shall log and classify errors based on severity and trigger automated responses.</p>  |

## Property Rental Management System Requirements Specification

|                |  |
|----------------|--|
| Postconditions | <ul style="list-style-type: none"> <li>- The payment is recorded.</li> <li>- The tenant receives an electronic receipt.</li> <li>- The landlord is notified of the payment.</li> </ul> |
|----------------|--|

|   |   |
|---|---|
| UC Name                                 | <b><u>UC - 22: Automatic Late Fee Penalty and Notifications</u></b>   |
| Summary                                 | This use case describes how landlords can set automatic late fee penalties for overdue rent payments, and how the system notifies tenants accordingly.  |
| Dependency                              | <b>UC-21:</b> Secure Payment Processing   |
| Actors                                  | <b>Primary Actor:</b> Tenant<br><b>Secondary Actor:</b> System (handles payment processing)   |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The tenant has a rental agreement with predefined late payment terms.</li> <li>- The payment due date has passed without payment being received.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The system checks if the tenant has missed the payment deadline.</li> <li>2. If overdue, the system applies the pre-set late fee to the tenant's balance.</li> <li>3. The system generates an automatic notification informing the tenant of the late fee.</li> <li>4. The notification is sent via email and displayed in the tenant's account dashboard.</li> </ol> |
| Description of the Alternative Sequence | <b>Tenant Pays Before Grace Period Ends</b> <ol style="list-style-type: none"> <li>1. The tenant makes a payment within the grace period.</li> <li>2. The system does not apply a late fee.</li> </ol>  |
| Non functional requirements             | <u>Availability &amp; Security (3.2.1.3)</u> - The system must ensure that sensitive data (user payment details, personal information) is only accessible to authorized personnel.<br><br><u>Logging (3.2.1.7)</u> - Any changes to access control settings (such as late fees applied) should be logged.   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The late fee is applied if necessary.</li> <li>- The tenant is notified.</li> <li>- The system updates the payment balance. Tenant Pays Before Grace Period Ends</li> </ul>  |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| UC Name                                 | <b><u>UC-23: Platform Activity &amp; Security Reports</u></b>  |
| Summary                                 | This use case describes the process in which administrators generate detailed reports on user logins, security logs, and system performance to monitor platform stability and detect anomalies.  |
| Dependency                              | <b>UC-01:</b> Secure User Login - for login activity data<br><b>UC-06:</b> Account Suspension & Deactivation - for suspension-related logs   |
| Actors                                  | <b>Primary:</b> Administrator  |
| Preconditions                           | <ul style="list-style-type: none"><li>- The system has been logging platform activities.</li><li>- The administrator has the appropriate role and access rights to reporting tools.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. Administrator accesses the reporting module.</li><li>2. Administrator selects the report type (system performance, security logs, login activity).</li><li>3. Filters (date range, user role, specific events) are applied if needed.</li><li>4. System complies and generates the report.</li><li>5. Administrator views, downloads or exports the report if needed.</li></ol> |
| Description of the Alternative Sequence | <p><b>A1. No data available:</b></p> <ol style="list-style-type: none"><li>1. If there are no records for the selected filters or time range, the system displays a “No Data Found” message.</li></ol> <p><b>A2. Report generation fails:</b></p> <ol style="list-style-type: none"><li>1. System displays an error message; administrator can retry or report the issue to support staff.</li></ol>                     |
| Non functional requirements             | The system shall ensure that all reports are generated securely and within acceptable response times. Reports must be accessible only to authorised users and remain available during high-load periods.   |
| Postconditions                          | <ul style="list-style-type: none"><li>- A detailed activity report is generated and available for download or review.</li><li>- Reports can be stored or exported for auditing and compliance purposes.</li></ul>  |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| UC Name                                 | <b><u>UC - 24: Automated Payment Reminders and Payment History</u></b>  |
| Summary                                 | This use case describes how tenants receive automated reminders for upcoming rent payments and can access their payment history.  |
| Dependency                              | <b>UC-21:</b> Secure Payment Processing   |
| Actors                                  | <b>Primary Actor:</b> Tenant<br><b>Secondary Actor:</b> System (handles payment processing)   |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The tenant has an active rental agreement.</li> <li>- The system has recorded upcoming payment dates.</li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The system schedules payment reminders based on the due date.</li> <li>2. A notification is sent via email and displayed in the tenant's account.</li> <li>3. The tenant logs into the system and views their payment history.</li> <li>4. The tenant sees past payments, upcoming payments, and transaction details</li> </ol> |
| Description of the Alternative Sequence | <b>Tenant Opts Out of Notifications</b> <ol style="list-style-type: none"> <li>1. The tenant modifies notification preferences to disable reminders.</li> </ol>   |
| Non functional requirements             | <u>Availability &amp; Security (3.2.1.3)</u> - The system must ensure that sensitive data (user payment details, personal information) is only accessible to authorized personnel.<br><br><u>Logging (3.2.1.2)</u> - The system shall support at least 50,000 registered users efficiently.   |
| Postconditions                          | The tenant receives reminders.<br>The tenant can view their payment history.  |

|            |   |
|------------|---|
| UC Name    | <b><u>UC - 25: Reporting Security Incidents</u></b>   |
| Summary    | This use case describes how users (tenants, landlords) can report security incidents such as hacking attempts, data breaches, or suspicious activities directly to the system administrator for prompt action and investigation.  |
| Dependency | <b>UC-1:</b> Ensures that the system's authentication and authorization mechanisms are secure, allowing only authorized users to report security incidents.<br><b>UC-23:</b> This requirement pertains to logging and tracking actions within the system, which ensures that all security incident submissions are logged for |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | <p>auditing and future reference. This will be crucial for keeping track of all actions related to security incidents.</p> <p><b>UC-25:</b> This pertains to the system's notification system, which will notify the administrator and the appropriate users when a security incident has been reported, ensuring a timely response and escalation.</p>  |
| Actors                                  | <p><b>Primary Actor:</b> User (tenant, landlord, property manager, or administrator)</p> <p><b>Secondary Actor:</b> System (supports the reporting process and routes reports to the administrator)</p>  |
| Preconditions                           | <ul style="list-style-type: none"><li>- The user is logged into the platform and has access to the "Report Security Incident" functionality</li><li>- The user is aware of the security issue ( noticing suspicious activity, receiving notifications about potential security breaches).</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The user accesses the "Security Incident Reporting" section from their dashboard or via the user settings.</li><li>2. The user fills in a report form detailing the incident, including any relevant information such as time, type of issue, and evidence (screenshots, logs).</li><li>3. The system validates the input for completeness and correctness (ensuring a description of the incident is included).</li><li>4. The system submits the report to the system administrator for investigation.</li><li>5. The user receives a confirmation message indicating that the incident report has been successfully submitted.</li><li>6. The administrator receives the incident report and begins the investigation process.</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the user is unable to provide sufficient details, the system prompts the user to provide more information before submitting the report.</li><li>2. If the report contains suspicious content or violates system rules ( false reporting), the system flags it for manual review by the administrator.</li><li>3. If the user fails to submit the form or closes the page before completing the report, the incident is not reported.</li></ol>   |
| Non functional requirements             | <ol style="list-style-type: none"><li>1. Incident reports should be processed and submitted within 2 seconds, ensuring a responsive and smooth user experience during high-traffic periods.</li><li>2. Incident report data, including sensitive information, must be encrypted both in transit and at rest to prevent unauthorized access.</li><li>3. The system must ensure that no data is lost during the report</li></ol>   |

## Property Rental Management System Requirements Specification

|                |  |
|----------------|--|
|                | <p>submission process, with automatic saving of progress in case of failure.</p> <ol style="list-style-type: none"><li>4. The incident reporting interface should be intuitive, requiring no more than 3 clicks to submit a report, ensuring ease of use for users with varying technical expertise.</li><li>5. The system should be able to handle multiple incident reports concurrently from at least 1,000 users without degradation in performance.</li></ol> |
| Postconditions | <ul style="list-style-type: none"><li>- The security incident report has been successfully submitted and routed to the administrator.</li><li>- The system logs the incident report submission for auditing and review purposes.</li><li>- The user is notified of the report submission and can track the progress of the investigation.</li></ul>  |

|            |   |
|------------|---|
| UC Name    | <b><u>UC - 26: Platform-wide Notification System</u></b>  |
| Summary    | This use case describes the platform-wide notification system, which allows users (tenants, landlords, property managers, and administrators) to receive alerts about system maintenance, feature updates, or critical issues that could affect the platform's functionality.   |
| Dependency | <p><b>UC-1:</b> Ensures that only authorized landlords with appropriate permissions can access and create promotional offers. This requirement is essential to secure the feature and prevent unauthorized access to modify promotional offers.</p> <p><b>UC-24:</b> Relates to the system's ability to handle the creation and updating of promotional offers, including validating offers within the allowed parameters. This ensures that the promotional offers comply with any predefined rules, including legal limits or system constraints.</p> <p><b>UC-38:</b> Pertains to the availability and responsiveness of the system, ensuring that landlords can create or modify promotions without system downtime. This requirement ensures that the platform is always available for landlords to manage their promotions effectively.</p> |
| Actors     | <p><b>Primary Actor:</b> User (tenant, landlord, property manager, or administrator)</p> <p><b>Secondary Actor:</b> System (handles sending notifications)</p>  |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| Preconditions                           | <ul style="list-style-type: none"><li>- The user is logged into the platform and has an active account.</li><li>- The platform has scheduled system maintenance, feature updates, or critical issues that need to be communicated to users.</li></ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The platform detects an event that requires user notification (e.g., system maintenance, feature updates, critical platform issues).</li><li>2. The system generates a notification with relevant details, including the nature of the event and any actions required by the user.</li><li>3. The system sends the notification to all relevant users based on their roles (e.g., tenants, landlords, property managers, administrators).</li><li>4. Users receive notifications through multiple channels, including in-app messages, emails, and/or push notifications (depending on their preferences).</li><li>5. The user reads the notification, which may include more details or a link to a help page for further instructions.</li><li>6. The system logs the notification delivery status (whether successfully delivered or failed).</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the notification fails to be delivered to a user due to technical issues ( network failure), the system will retry delivery up to three times within a 24-hour period.</li><li>2. If the user does not have push notifications enabled, the system will only send an email or an in-app notification.</li><li>3. If a user opts out of receiving specific types of notifications (maintenance updates), they will not receive those types of alerts, and the system logs this preference.</li></ol>   |
| Non functional requirements             | <ul style="list-style-type: none"><li>● The chatbot should respond to user queries within 1 second to ensure a fast and smooth interaction.</li><li>● The AI chatbot must ensure user interactions are logged securely, with sensitive data being encrypted and stored according to data protection regulations.</li><li>● The chatbot should provide correct and relevant responses to at least 95% of common inquiries, escalating complex issues to human support.</li><li>● The chatbot interface should be easy to use, with a user-friendly design allowing seamless interactions, even for non-technical users.</li><li>● The system should handle thousands of simultaneous chatbot interactions without performance degradation.</li></ul>   |
| Postconditions                          | <ul style="list-style-type: none"><li>- The system has successfully sent notifications to all users based on the event or trigger.</li><li>- Users have been informed of the system maintenance, feature updates, or critical issues, and they can take any necessary actions.</li><li>- The system logs the notification delivery status and user interactions for future reference and troubleshooting.</li></ul>   |



## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| UC Name                                 | <b><u>UC - 27: AI-Powered Chatbot</u></b>  |
| Summary                                 | This use case describes the integration of an AI-powered chatbot within the platform, which provides instant responses to frequently asked questions, assists with common rental-related inquiries, and escalates complex issues to human support staff when needed.   |
| Dependency                              | <b>UC-33:</b> The FAQ section will reduce the number of queries for the AI chatbot by providing users with quick answers, improving chatbot efficiency.<br><br><b>UC-36:</b> Technical support requests raised by the chatbot will be tracked and handled by support staff, ensuring prompt resolution.  |
| Actors                                  | <b>Primary Actor:</b> Tenant, Landlord, Property Manager (user interacting with the chatbot).<br><b>Secondary Actor:</b> AI Chatbot, Support Staff (human escalations)   |
| Preconditions                           | <ul style="list-style-type: none"><li>- The user (tenant, landlord, or property manager) is logged into the system.</li><li>- The AI chatbot is enabled and trained to answer rental-related questions.</li></ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The user accesses the AI chatbot through the platform interface (web/mobile app).</li><li>2. The user asks a question or submits an inquiry regarding the rental process, lease terms, property availability, etc.</li><li>3. The AI chatbot processes the query, searching its database for relevant information.</li><li>4. The chatbot provides an instant response based on its training and available resources.</li><li>5. If the inquiry is complex or not in the chatbot's knowledge base, the system escalates the issue to a human support staff member.</li><li>6. The user is notified that their query has been escalated and will be addressed by support.</li><li>7. The support staff responds to the user's inquiry.</li><li>8. The user is notified once the issue has been resolved.</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the AI chatbot is unable to understand the user query, it will politely inform the user that it couldn't find an answer and escalate the inquiry to human support.</li><li>2. If the user chooses to escalate the issue directly, the chatbot will hand over the conversation to support without attempting to answer.</li></ol>   |
| Non functional requirements             | <ul style="list-style-type: none"><li>● Feedback submissions should be processed and stored within 2 seconds to ensure an immediate reflection of tenant reviews.</li></ul>  |

## Property Rental Management System Requirements Specification

|                |   |
|----------------|---|
|                | <ul style="list-style-type: none"><li>• All tenant feedback should be securely stored and anonymized, ensuring tenant privacy is preserved.</li><li>• The system should ensure that feedback is reliably recorded and stored without data corruption, with regular checks to ensure data consistency.</li><li>• The feedback interface should be simple, allowing tenants to provide ratings with minimal effort ( 1-5 stars) and to add optional comments.</li><li>• The feedback system should be robust enough to handle feedback from tens of thousands of tenants without performance degradation during peak usage times.</li></ul> |
| Postconditions | <ul style="list-style-type: none"><li>- The user has received a response to their inquiry, either from the chatbot or a human support representative.</li><li>- The chatbot logs all user interactions for future reference or improvements.</li></ul>  |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><u>UC - 28: Secure Messaging Feature for Tenants and Landlords</u></b>  |
| Summary                          | This use case allows tenants to securely communicate directly with landlords or property managers regarding maintenance, lease terms, or other property-related issues. The messaging feature ensures privacy and facilitates smooth communication between tenants and property managers or landlords.   |
| Dependency                       | <p><b>UC-1:</b> Secure login and authentication for both tenants and landlords ensures that communication via the messaging system is protected.</p> <p><b>UC-11:</b> Landlords are responsible for maintenance, and the messaging system facilitates communication about maintenance requests and updates.</p> <p><b>UC-18:</b> The system allows tenants to request lease renewals, and messaging provides a secure way to communicate such requests directly to landlords or property managers.</p> |
| Actors                           | <p><b>Primary Actor:</b> Tenant</p> <p><b>Secondary Actor:</b> Landlord, Property Manager</p>  |
| Preconditions                    | <ul style="list-style-type: none"><li>- Both the tenant and the landlord/property manager are registered and logged into the platform.</li><li>- Both parties have an active communication channel enabled and can send and receive messages.</li></ul>  |
| Description of the Main Sequence | <ol style="list-style-type: none"><li>1. The tenant logs into the system and accesses the messaging feature.</li><li>2. The tenant selects the landlord or property manager they wish to</li></ol>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | <p>communicate with from the available list (usually based on the property they are renting).</p> <ol style="list-style-type: none"><li>3. The tenant types their message (a maintenance request, clarification on lease terms, or any other property-related inquiry).</li><li>4. The tenant sends the message, and the system delivers it instantly.</li><li>5. The landlord or property manager receives the message within their messaging inbox. They can read the message and respond within the platform's secure messaging system.</li><li>6. The tenant and landlord/property manager exchange messages as needed.</li><li>7. If necessary, the tenant or landlord can escalate the issue to the appropriate department ( maintenance staff, customer support, etc.), either by attaching the message to a support request or forwarding the message to the department directly.</li><li>8. After the issue is resolved, the tenant and landlord can close the conversation or continue communication if needed.</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the tenant or landlord experiences network connectivity issues, they will receive a notification informing them that the message has not been delivered, with a prompt to retry.</li><li>2. If a message is not received within the platform, the system should attempt to re-send the message and notify the sender of the failure after an attempted resend.</li><li>3. If the message fails after multiple attempts, the system should ask the tenant or landlord to check their internet connection and reattempt communication.</li></ol>  |
| Non functional requirements             | <ul style="list-style-type: none"><li>• Messages should be delivered within 2 seconds of being sent, ensuring near-instant communication between tenants and landlords.</li><li>• All messages must be encrypted end-to-end to protect the content of conversations from unauthorized access.</li><li>• The system should ensure that conversations are not lost, even if there is a network issue during message delivery.</li><li>• The messaging interface should be simple and easy to navigate, requiring minimal effort to send or read messages.</li><li>• The system should support up to 1,000 simultaneous users without a decrease in performance.</li></ul>   |
| Postconditions                          | <ul style="list-style-type: none"><li>- After the message exchange is completed, the tenant and landlord/property manager have successfully communicated, and any necessary follow-up actions (maintenance requests, lease clarifications) are recorded in the system.</li><li>- All conversations are stored securely, ensuring that both the tenant and the landlord can reference them in the future if needed.</li></ul>  |

## Property Rental Management System Requirements Specification

|                                  |   |
|----------------------------------|---|
| UC Name                          | <b><u>UC - 29: Landlord Promotional Offers</u></b>  |
| Summary                          | This use case allows landlords to create and manage promotional offers or discounts on their rental properties. These promotions can include reduced rent for the first month, referral bonuses, or discounts for long-term leases, which are intended to attract potential tenants.  |
| Dependency                       | <p><b>UC-09:</b> Landlords are able to list, update, or remove rental properties, and the promotional offers system integrates with property listings to attract potential tenants.</p> <p><b>UC-15:</b> Tenants can search for rental properties using various filters, and promotional offers can be applied to property listings to enhance visibility and appeal.</p> <p><b>UC-26:</b> The system-wide notification system ensures that tenants and landlords are informed about new promotional offers, updates, and discounts.</p>  |
| Actors                           | <p><b>Primary Actor:</b> Landlord</p> <p><b>Secondary Actor:</b> System (which processes and displays the promotional offers to tenants)</p>  |
| Preconditions                    | <ul style="list-style-type: none"><li>- The landlord must be logged into the platform.</li><li>- The landlord must have available properties listed for rent.</li><li>- The landlord must have appropriate permissions to create and modify promotional offers.</li></ul>   |
| Description of the Main Sequence | <ol style="list-style-type: none"><li>1. The landlord logs into the platform and navigates to the property management interface.</li><li>2. The landlord selects a property they wish to add a promotional offer for.</li><li>3. The landlord chooses the option to create a promotional offer for the selected property.</li><li>4. The landlord inputs the details of the promotional offer, including the type (reduced rent, referral bonuses, long-term lease discounts), duration, and terms of the offer.</li><li>5. The system validates the promotional offer to ensure it meets legal requirements and platform guidelines ( no offer exceeds any discount limits).</li><li>6. The system applies the promotion to the property listing and displays the offer on the property page, visible to potential tenants.</li><li>7. The landlord can later return to the system to update or remove the promotion as necessary.</li></ol> |
| Description of the Alternative   | <ol style="list-style-type: none"><li>1. If the landlord tries to create a promotional offer outside the allowed parameters ( a discount that exceeds the platform's or legal limits), the system will display an error message notifying the landlord of the</li></ol>   |

## Property Rental Management System Requirements Specification

|                             |  |
|-----------------------------|--|
| Sequence                    | <p>violation.</p> <p>2. The system will also provide guidance on offer restrictions ( maximum allowable discount).</p>   |
| Non functional requirements | <ul style="list-style-type: none"> <li>• Promotional offers should be updated in real-time, reflecting immediately on property listings.</li> <li>• The system should process the creation of promotional offers within 3 seconds to provide a responsive user experience.</li> <li>• The system must ensure that only authorized landlords can create or modify promotional offers.</li> <li>• The interface for creating promotional offers should be simple and intuitive, with easy-to-understand fields for landlords.</li> <li>• The system should be able to handle multiple promotional offers simultaneously across different properties without performance issues.</li> </ul> |
| Postconditions              | <ul style="list-style-type: none"> <li>- The promotional offer is successfully applied to the selected property, and the property listing is updated with the offer details. Potential tenants can view the promotion when browsing the property.</li> <li>- The landlord can manage or update the promotion at any time through the platform, ensuring that offers can be adjusted as needed.</li> </ul>  |

|               |  |
|---------------|--|
| UC Name       | <b><u>UC - 30: Tenant Feedback and Rating System</u></b>   |
| Summary       | This use case allows tenants to provide feedback and rate their experience with landlords and property managers. Tenants can leave comments on aspects such as maintenance responsiveness, property condition, and overall satisfaction, helping future tenants make informed decisions.   |
| Dependency    | <p><b>UC-29:</b> The ability to create and manage promotional offers could impact tenant feedback, as tenants might leave reviews based on whether the promotional offers were honored or effectively communicated.</p> <p><b>UC-17:</b> Landlords review rental applications, and feedback from tenants could influence future rental decisions. Positive or negative feedback might be factored in when evaluating prospective tenants.</p> <p><b>UC-23:</b> Administrators generating detailed reports could use tenant feedback to assess overall satisfaction, identify areas for improvement, and ensure platform policies align with tenant expectations.</p> |
| Actors        | <p><b>Primary Actor:</b> Tenant</p> <p><b>Secondary Actors:</b> Landlord,Property Manager</p>  |
| Preconditions | <ul style="list-style-type: none"> <li>- The tenant has completed their stay or rental experience with a</li> </ul>  |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | <p>landlord or property manager.</p> <ul style="list-style-type: none"><li>- The tenant is logged into the system and has access to the feedback feature for the relevant property and landlord.</li></ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. The tenant logs into the platform and navigates to the feedback section.</li><li>2. The tenant selects the landlord or property manager they wish to rate based on their completed rental experience.</li><li>3. The tenant rates their experience by choosing a rating and leaving comments about their experience with the landlord/property manager (maintenance responsiveness, property condition, overall satisfaction).</li><li>4. The tenant submits the feedback.</li><li>5. The system stores the feedback securely and updates the landlord's or property manager's profile with the rating and comments.</li><li>6. The feedback is displayed on the landlord's or property manager's profile for future tenants to view.</li></ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"><li>1. If the tenant experiences any issues during the submission process ( a network error), the system will display an error message, and the tenant will be prompted to retry submitting the feedback.</li><li>2. If the system encounters an issue storing the feedback ( technical error), it will notify the tenant and ask them to retry submitting their review.</li></ol>   |
| Non functional requirements             | <ul style="list-style-type: none"><li>● Feedback submissions should be processed and stored within 2 seconds, allowing for immediate updates to the landlord's or property manager's profile.</li><li>● All feedback and ratings should be securely stored, ensuring tenant privacy and preventing unauthorized access.</li><li>● The system should ensure data consistency, ensuring that feedback cannot be corrupted during the submission process.</li><li>● The interface for submitting feedback should be intuitive, requiring minimal effort for tenants to submit ratings and comments.</li><li>● The system should be able to handle thousands of feedback submissions concurrently without performance degradation.</li></ul>   |
| Postconditions                          | <ul style="list-style-type: none"><li>- The tenant's feedback and rating are securely stored and added to the landlord's or property manager's profile.</li><li>- The feedback is publicly visible on the landlord's or property manager's profile, helping future tenants make informed decisions about renting the property.</li></ul>   |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| UC Name                                 | <b><u>UC - 31: Multi-Language Support</u></b>  |
| Summary                                 | Users can select their preferred language for the interface and notifications.   |
| Dependency                              | <b>UC-26:</b> Platform-wide notifications  |
| Actors                                  | <b>Primary Actor:</b> All Users<br><b>Secondary Actor:</b> System  |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The user is registered and logged in.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The user navigates to settings.</li> <li>2. The user selects a preferred language from the available options.</li> <li>3. The system updates the interface and future notifications to the selected language.</li> </ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. Language Not Available</li> <li>2. The system displays a message indicating the requested language is not supported.</li> </ol>  |
| Non functional requirements             | <u>Usability (3.2.1.1)</u> - The system must provide an intuitive and user-friendly interface.<br><u>External Requirements (3.2.3)</u> - The system shall offer multi-language support.  |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The user's language preference is applied.</li> <li>- Future notifications are sent in the selected language.</li> </ul>  |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><u>UC - 32: Onboarding Tutorials for New Users</u></b>  |
| Summary                          | New users complete an onboarding process with guided tutorials.  |
| Dependency                       | <b>UC-01:</b> User Registration  |
| Actors                           | <b>Primary Actor:</b> New User<br><b>Secondary Actor:</b> System   |
| Preconditions                    | The user has successfully registered and logged in for the first time.   |
| Description of the Main Sequence | <ol style="list-style-type: none"> <li>1. The system detects a new user login and launches the tutorial.</li> <li>2. The user is guided through system features.</li> <li>3. The user can complete the tutorial or skip it.</li> </ol> |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | 4. The user can revisit the tutorial from the help section later.   |
| Description of the Alternative Sequence | <b>User Skips the Tutorial</b><br>1. The user selects 'Skip' and exits the tutorial.  |
| Non functional requirements             | <u>Usability (3.2.1.1)</u> - The onboarding process must not exceed 5 minutes.  |
| Postconditions                          | <ul style="list-style-type: none"><li>- The user is familiar with system features.</li><li>- The user can access tutorials later.</li></ul> |

|                                  |   |
|----------------------------------|---|
| UC Name                          | <b><u>UC - 33: FAQ Section for Common Inquiries</u></b>   |
| Summary                          | Users can access an FAQ section to find quick answers.  |
| Dependency                       | <b>UC-32:</b> Onboarding Tutorials  |
| Actors                           | <b>Primary Actor:</b> All Users<br><b>Secondary Actor:</b> System   |
| Preconditions                    | The user has access to the system.  |
| Description of the Main Sequence | <ol style="list-style-type: none"><li>1. The user navigates to the FAQ section.</li><li>2. The user searches for a question or browses categories.</li><li>3. The system displays relevant answers.</li></ol> |
| Non functional requirements      | <u>Usability (3.2.1.1)</u> - FAQs must be accessible within 2 clicks.   |
| Postconditions                   | <ul style="list-style-type: none"><li>- The user is familiar with system features.</li><li>- The user can access tutorials later.</li></ul>   |

|            |  |
|------------|--|
| UC Name    | <b><u>UC-34: Rental Price Analysis &amp; Market Feedback</u></b>   |
| Summary    | This use case describes the process in which the system analyses rental prices and informs tenants whether a property's price is below, at, or above the market average. |
| Dependency | <b>UC-09:</b> Property Listing Data - price, description, size, etc.<br><b>UC-15:</b> Supports features like property search   |



## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| Actors                                  | <b>Primary:</b> Tenant  |
| Preconditions                           | <ul style="list-style-type: none"><li>- The system has access to current rental listings and historical pricing data.</li><li>- The tenant is browsing or viewing property details.</li></ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"><li>1. Tenant views a property listing.</li><li>2. System analyses the property's price based on location, size and features.</li><li>3. System compares the price with the market average.</li><li>4. System displays a label or feedback (e.g., "Below Market", "Above Market") on the listing.</li></ol> |
| Description of the Alternative Sequence | <b>Insufficient Market Data:</b> If the system lacks enough data for the location or property type, it notifies the tenant that feedback is unavailable at the moment.  |
| Non functional requirements             | The system shall provide price analysis results in under 2 seconds, ensuring a seamless user experience. The analysis must be based on up-to-date data and maintain high accuracy.  |
| Postconditions                          | <ul style="list-style-type: none"><li>- The tenant receives market feedback for the viewed property.</li><li>- The feedback may influence rental decision-making or comparisons.</li></ul>  |

|            |   |
|------------|---|
| UC Name    | <b><u>UC-35: Financial Reports for Landlords &amp; Managers</u></b>   |
| Summary    | This use case describes the process in which landlords or property managers generate detailed financial reports to monitor income, expenses and manage property-related finances. |
| Dependency | <b>UC-21:</b> Process Tenant Payments<br><b>UC-29:</b> Promotional Offers   |
| Actors     | <b>Primary:</b> Landlords, Property Managers.   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| Preconditions                           | <ul style="list-style-type: none"> <li>- The user is a verified landlord or property manager.</li> <li>- Financial transactions (e.g., rent payments, expenses) have been recorded in the system.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. Landlord or property manager navigates to the financial reports section.</li> <li>2. They select the type or report (e.g., income summary, expense report).</li> <li>3. Optional filters (e.g., date range, property) are applied.</li> <li>4. System generates the report.</li> <li>5. User reviews, downloads or exports the report.</li> </ol> |
| Description of the Alternative Sequence | <b>No Available Financial Data:</b> System notifies the user that no report can be generated due to insufficient data for the selected filters or timeframe.  |
| Non functional requirements             | The system shall generate financial reports securely and within 2 seconds. Reports must be accessible only to authorised users and remain available during high traffic periods. Generated reports shall be exportable in standard formats (PDF, Excel).  |
| Postconditions                          | <ul style="list-style-type: none"> <li>- A financial report is generated and available for viewing, download, or export.</li> <li>- The report can be used for personal tracking or shared externally for accounting purposes.</li> </ul>   |

|            |   |
|------------|---|
| UC Name    | <b><u>UC-36: Technical Support Request Tracking</u></b>   |
| Summary    | This use case describes the process in which tenants submit technical support requests that are reviewed and approved by landlords or assigned property managers before being handled by support staff. |
| Dependency | <b>UC-13:</b> Property manager acts on behalf of landlord if assigned<br><b>UC-11:</b> Landlord manages issues directly if no manager is assigned   |
| Actors     | <b>Primary:</b> Support Staff   |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
|   | <b>Secondary:</b> Tenants, Landlords, Property Managers  |
| Preconditions                           | <ul style="list-style-type: none"> <li>- A support request has been submitted by a tenant.</li> <li>- The associated property has either a landlord or assigned property manager to review the request.</li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. Tenant submits a technical support request through the system.</li> <li>2. The system routes the request to the assigned property manager, or to the landlord if no manager is assigned.</li> <li>3. The landlord or property manager reviews and if the request is approved, it is forwarded to the support staff.</li> <li>4. Support staff investigates the issue and updates the status.</li> <li>5. Support staff resolves the issue and marks the request as completed.</li> <li>6. System notifies the tenant of the resolution.</li> </ol> |
| Description of the Alternative Sequence | <p><b>A1. Request Rejected by Landlord/Property Manager</b></p> <ol style="list-style-type: none"> <li>1. Request is rejected with a reason.</li> <li>2. The system notifies the requester with a reason for rejection.</li> </ol> <p><b>A2. No Response to Request</b></p> <ol style="list-style-type: none"> <li>1. If the landlord or manager does not act within a defined time window (e.g., 48 hours), the system sends a reminder or escalates to admin (optional future enhancement).</li> </ol>   |
| Non functional requirements             | The system shall securely track all support requests with real-time status updates. Role-based access shall control who can approve and act on requests. All actions must be logged for accountability, and support staff must acknowledge new requests within 24 hours.   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The support request is approved and resolved or rejected with a reason.</li> <li>- A full activity log is stored for future reference and accountability.</li> </ul>  |

|         |  |
|---------|--|
| UC Name | <b><u>UC - 37: Administrators Removing Policy-Violating Content</u></b>                                      |
| Summary | Policy-Violating Content received from reviews is removed after thorough investigation by the administrator. |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
| Dependency                              | <b>UC-30:</b> since tenant reviews are at stake.<br><b>UC-09:</b> since property listings are at stake.   |
| Actors                                  | <b>Primary Actor:</b> Administrator is responsible for checking and removing policy-violating content.<br><b>Secondary Actors:</b> Landlords and tenants.They are the ones to post this content.  |
| Preconditions                           | <ul style="list-style-type: none"> <li>- There must exist at least one policy-violating content for the administrator to check and remove.</li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The administrator is presented with content suspected to violate our privacy.</li> <li>2. During the process of checking he is able to access our General Policy material directly from the system.</li> <li>3. The administrator removes the content from the System and initiates an automated warning message to the actor responsible.</li> </ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. The administrator is presented with the content suspected to violate our privacy.</li> <li>2. During the process of checking he is able to access our General Policy material directly from the system.</li> <li>3. He/she decides that particular content does not violate our General Policy.</li> </ol>  |
| Non functional requirements             | Availability & Security non-functional requirement nr 11-Any privacy-violation report captured by the System, must be sent to the Administrator for review.   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- The System is secure and cleansed from privacy-violating content.</li> <li>- Violating reports are reviewed and answered.</li> </ul>   |

|               |  |
|---------------|--|
| UC Name       | <b><u>UC - 38: Platform Wide announcements by Administrators</u></b>   |
| Summary       | The Administrator will make Platform-Wide announcements for all types of Users for important changes and updates of the System.                                    |
| Dependency    | <b>UC-26:</b> The Administrator depends on the System's notification functionality to spread announcements.  |
| Actors        | <b>Primary Actor:</b> Administrator. The only actor in this user case is the Administrator which notifies all Users about major updates and changes to the System. |
| Preconditions | <ul style="list-style-type: none"> <li>- There must exist a change or update.</li> </ul>   |

## Property Rental Management System Requirements Specification

|   |   |
|---|---|
|   | <ul style="list-style-type: none"> <li>- The notification functionality of the System must work for all Users.</li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The Administrator has prepared a change or update for the System.</li> <li>2. The Administrator notifies all Users about this change or update via the notification functionality of the System.</li> </ol>   |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. The Administrator has prepared a change or update for the System.</li> <li>2. The Administrator does not notify all users immediately, but sets a schedule for the notification prior to the time of the change or update.</li> <li>3. Once the scheduled date and time has arrived, the System sends a notification to all Users.</li> </ol> |
| Non functional requirements             | Availability & Security nr 5: Users should receive real time notifications via email, SMS or mobile push notifications.   |
| Postconditions                          | <ul style="list-style-type: none"> <li>- All Users are notified about changes or updates to the System.</li> </ul>  |

|                                  |  |
|----------------------------------|--|
| UC Name                          | <b><u>UC - 39: User Profile Updates and Customization</u></b>  |
| Summary                          | Users are able to update their profile information including contact details and preferences.  |
| Dependency                       | <p><b>UC-01:</b> Users need to firstly create their profiles, before continuing to modify it.</p> <p><b>UC-05:</b> Only a verified user is able to modify his profile, whereas for unverified profiles this option is locked.</p>  |
| Actors                           | <b>Primary Actor:</b> This user case only has one primary Actor, the user which might be a: tenant, landlord, property manager, legal advisor, support staff member, etc.  |
| Preconditions                    | <ul style="list-style-type: none"> <li>- The user must have a created and verified account, otherwise he will not be allowed to modify his profile information.</li> </ul>   |
| Description of the Main Sequence | <ol style="list-style-type: none"> <li>1. The user clicks on his profile.</li> <li>2. In personal profile information for example: contact number an edit button appears and he presses it.</li> <li>3. Inserts the new value, which will be automatically verified if it has a valid format.</li> <li>4. Press complete.</li> </ol> |

## Property Rental Management System Requirements Specification

|   |  |
|---|--|
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. The user clicks on his profile.</li> <li>2. He scrolls down to edit all information.</li> <li>3. Edits all the information he intends to which will be format checked by the System.</li> <li>4. Press Complete.</li> </ol>                      |
| Non functional requirements             | <p>Usability nr.1:The system must have a user-friendly interface,to accommodate the user in its use.</p> <p>Usability nr.5:Users will be presented with a user manual.</p> <p>Usability nr.6:System navigation shall require no more than three clicks to reach any key functionality.</p> |
| Postconditions                          | <ul style="list-style-type: none"> <li>- Profile information that the user intends to change is updated.</li> <li>- This updated information will be visible on his profile for other users to see.</li> </ul>   |

|   |  |
|---|--|
| UC Name                                 | <b><u>UC - 40: Legal Advisors Mediating Conflicts</u></b>  |
| Summary                                 | Legal Advisors will help in the mediation of the conflicts between landlords/property managers and tenants through our System.   |
| Dependency                              | <p><b>UC-20:</b> Requesting maintenance or repairs by the tenant side or landlord/property manager side may oblige the legal party to solve the contract aspect of the responsible party for this action.</p> <p><b>UC-20:</b> Contract terms are potential conflicts between the two parties , which can be resolved by the Legal Advisors.</p>   |
| Actors                                  | <p><b>Primary Actor:</b> Legal Advisor.They mediate the conflict.</p> <p><b>Secondary Actors:</b> Landlord/Property Manager/tenant:They are parties of the conflict waiting to be resolved.</p>  |
| Preconditions                           | <ul style="list-style-type: none"> <li>- A deal must have been arranged by the two sides.</li> <li>- Legal terms must exist in the first place between the parties.</li> <li>- A conflict must have arisen beforehand.</li> </ul>  |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The legal Officer accesses the conflict and the legal documents of the deal.</li> <li>2. The Legal Officer may require the history of the tenant and landlord to judge their behaviour.</li> <li>3. The Officer makes a decision about the case and justifies it with legal arguments.</li> <li>4. The Officer notifies the two parties about the decision.</li> </ol> |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. The legal Officer accesses the conflict and the legal documents of the deal.</li> <li>2. The Legal Officer may require the history of the tenant and</li> </ol>  |

## Property Rental Management System Requirements Specification

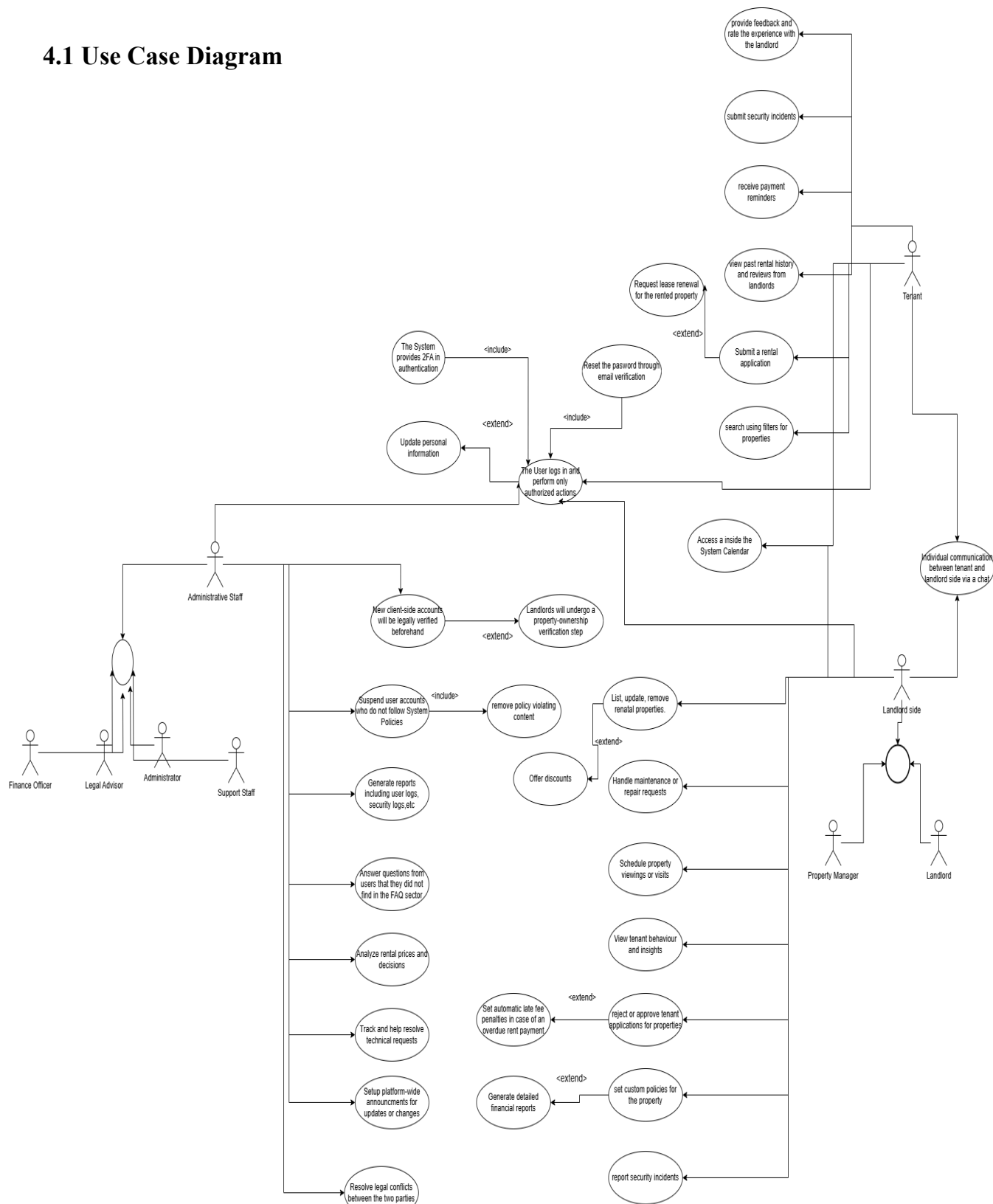
|                             |   |
|-----------------------------|---|
|                             | <p>landlord to judge their behaviour.</p> <ol style="list-style-type: none"> <li>3. The Officer may decide to demand further documents by any or both the parties to make a thorough and complete decision.</li> <li>4. The conflict is left at a pending state.</li> </ol> |
| Non functional requirements | None  |
| Postconditions              | <ul style="list-style-type: none"> <li>- The Conflict is solved by the Legal Officer.</li> <li>- Both Users have been notified of the decision.</li> </ul>  |

|   |  |
|---|--|
| UC Name                                 | <b><i>UC - 41: Secure Document Management</i></b>  |
| Summary                                 | Users are provided with a save inside the system Drive where they can upload secure documents, where access is personal.   |
| Dependency                              | <b>UC-03:</b> Verifying users before they log in is important for a Secure Document functionality so only the real users can access it.  |
| Actors                                  | <b>Primary Actor:</b> User who can be a: landlord, tenant, property manager, etc is the sole actor of this user case, the one that uploads the documents and edits them or deletes them.   |
| Preconditions                           | <ul style="list-style-type: none"> <li>- The User must have created an account.</li> <li>- The account must have been verified by the System and the competences charged with verifying it.</li> </ul>   |
| Description of the Main Sequence        | <ol style="list-style-type: none"> <li>1. The User logs in to his verified profile.</li> <li>2. He presses on his profile.</li> <li>3. He presses the My Drive button.</li> <li>4. In his My Drive interface he can add, edit or remove documents.</li> </ol>  |
| Description of the Alternative Sequence | <ol style="list-style-type: none"> <li>1. The User logs in to his verified profile.</li> <li>2. Via the search bar, he searches "My Drive".</li> <li>3. In the dropdown options he presses "My Drive".</li> <li>4. In his My Drive interface he can add, edit or remove documents.</li> </ol>  |
| Non functional requirements             | <p><u>Usability nr.1:</u> The System must have a user-friendly interface to navigate easily to any functionality.</p> <p><u>Usability nr.5:</u> Users will be granted a user manual to accommodate them with different functionalities.</p> <p><u>Availability &amp; Security nr.7:</u> Users will be faced with 2FA when logging in to ensure security.</p> |

# Property Rental Management System Requirements Specification

|                |   |
|----------------|---|
|                | <u>Availability &amp; Security nr.12:</u> Sensitive data such as content inside the Drive will be only accessible to the User.  |
| Postconditions | <ul style="list-style-type: none"> <li>- User has a personal inside the System Drive to store documents.</li> <li>- This Drive is secure for unauthorized personnel.</li> </ul> |

## 4.1 Use Case Diagram





# **Property Rental Management System Requirements Specification**