

Software Requirements Specification

For

RentEase - Online Rental Marketplace

Prepared by : Fasih Shahid Abbasi , Ammar Ahmed , Musayyub Haider.
(22I-2235) (22I-1267) (22I-0862)

Submitted To : Ms. Javaria Imtiaz

March 23, 2025

VERSION 1.0 APPROVED.

[NUCES-Fast University]

Table of Contents

- [Revision History](#)
- [1. Introduction](#)
 - [1.1 Purpose](#)
 - [1.2 Document Conventions](#)
 - [1.3 Intended Audience and Reading Suggestions](#)
 - [1.4 Product Scope](#)
 - [1.5 References](#)
- [2. Overall Description](#)
 - [2.1 Product Perspective](#)
 - [2.2 Product Functions](#)
 - [2.3 User Classes and Characteristics](#)
 - [2.4 Operating Environment](#)
 - [2.5 Design and Implementation Constraints](#)
 - [2.6 User Documentation](#)
 - [2.7 Assumptions and Dependencies](#)
- [3. External Interface Requirements](#)
 - [3.1 User Interfaces](#)
 - [3.2 Hardware Interfaces](#)
 - [3.3 Software Interfaces](#)

- [3.4 Communications Interfaces](#)
- [4. System Features](#)
 - [4.1 User Registration and Profile Management](#)
 - [4.2 Item Listing and Management](#)
 - [4.3 Rental Booking and Processing](#)
 - [4.4 Payment Processing](#)
 - [4.5 Review and Rating System](#)
 - [4.6 Search and Browse Functionality](#)
- [5. Other Nonfunctional Requirements](#)
 - [5.1 Performance Requirements](#)
 - [5.2 Safety Requirements](#)
 - [5.3 Security Requirements](#)
 - [5.4 Software Quality Attributes](#)
 - [5.5 Business Rules](#)
- [6. Diagrams](#)
 - [6.1 Use Case Diagram](#)
 - [6.2 Activity Diagrams](#)
 - [6.3 Sequence Diagrams](#)
- [Appendix A: Glossary](#)
- [Appendix B: Analysis Models](#)
- [Appendix C: To Be Determined List](#)

Revision History

Name	Date	Reason for changes	Version
Team RentEase	23rd March 2025	Initial SRS Document	1.0

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the development of RentEase, an online rental marketplace. The document specifies the functional and non-functional requirements for the system to be developed as part of an introductory

software engineering course project. It will serve as the primary reference for the development team.

1.2 Document Conventions

The following conventions are used in this document:

- User stories are written in the format: "As a [type of user], I want [some goal] so that [some reason]."
- Requirements are labeled with unique identifiers: REQ-F# for functional requirements and REQ-NF# for non-functional requirements.
- Priorities are marked as High, Medium, or Low to indicate their relative importance.
- TBD (To Be Determined) indicates information that is currently unavailable but will be defined later.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Development Team: To understand what needs to be built
- Course Instructor: To evaluate the project requirements and design
- Peer Reviewers: To provide feedback on the requirements
- Future Maintainers: To understand the system's intended functionality

For a comprehensive understanding, readers should first review the Introduction and Overall Description sections. Developers should focus on System Features and External Interface Requirements. Course instructors may be particularly interested in the Design and Implementation Constraints and the provided diagrams.

1.4 Product Scope

RentEase is a web-based platform that connects people who want to rent items temporarily with those who have items available for rent. The platform aims to create a sharing economy where users can monetize their underutilized possessions while helping others avoid unnecessary purchases for temporarily needed items.

Key objectives include:

- Facilitating peer-to-peer rental transactions
- Providing a secure and transparent rental process
- Building trust through a review and rating system
- Creating economic opportunities for item owners
- Promoting sustainability by extending the utility of existing items

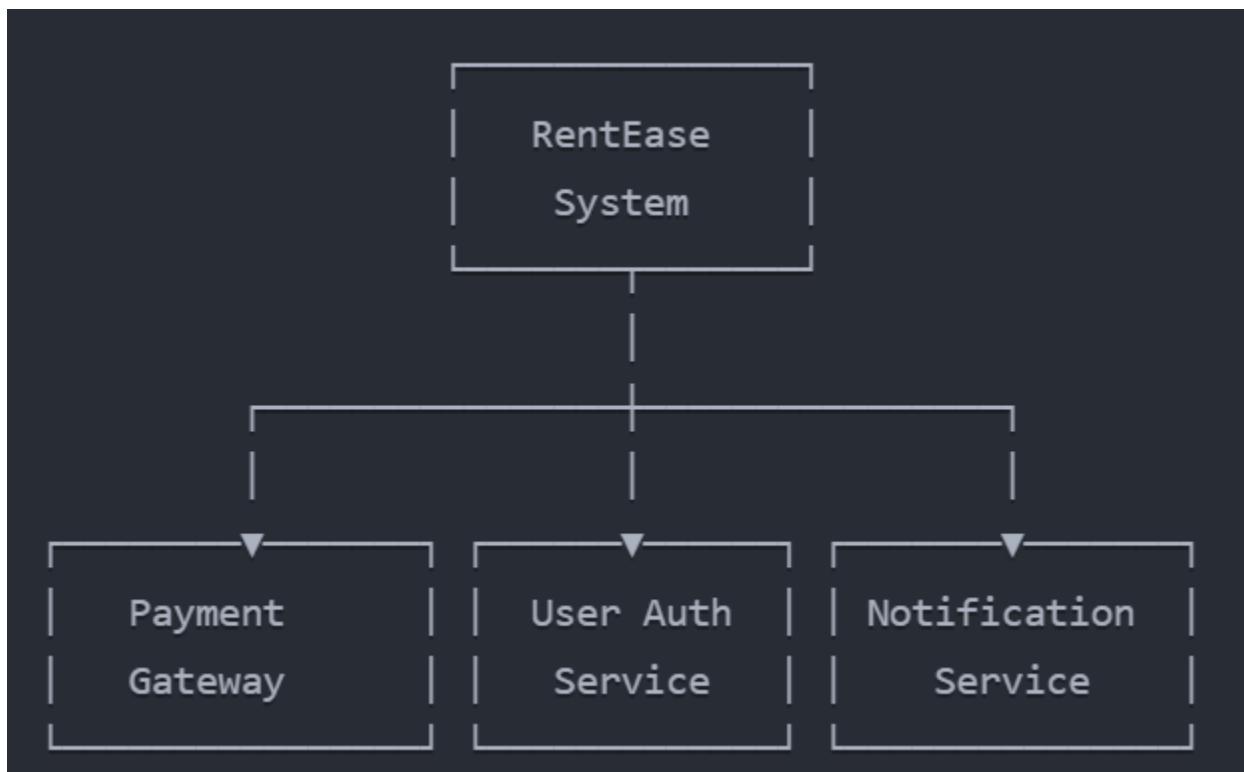
1.5 References

1. Course Assignment Description, Intro to Software Engineering, March 2025
2. SRS Template by Karl E. Wieggers, Version 1.0
3. IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

2. Overall Description

2.1 Product Perspective

RentEase is a new, self-contained product that will operate as a web application. It will leverage third-party services for user authentication, payment processing, and messaging. The system will initially be deployed as a standalone application but may later be integrated with mobile applications or other marketplaces.



2.2 Product Functions

The major functions of RentEase include:

- User account management (registration, login, profile management)
- Item listing and management (add, edit, remove rental items)
- Rental booking process (search, request, approve, track)
- Secure payment processing (rental fees, deposits)
- Review and rating system (for both renters and owners)
- Messaging system between renters and owners
- Dashboard for rental history and current rentals
- Search and filter functionality for browsing items
- Notification system for rental status updates

2.3 User Classes and Characteristics

1. Item Owners

- Users who list items for rent
- Need to effectively showcase their items and set rental terms
- Use the system to manage their listings and rental requests
- Frequency: Moderate to high usage
- Technical expertise: Varies

2. Renters

- Users looking to rent items temporarily
- Need to easily find and book desired items
- Use the system to search, book, and return rented items
- Frequency: Occasional to moderate usage
- Technical expertise: Varies

3. Administrators

- Staff who manage and maintain the platform
- Need comprehensive access to monitor and moderate content
- Use the system to resolve disputes and ensure smooth operation
- Frequency: High usage
- Technical expertise: High

4. Guest Users

- Unregistered visitors who browse the platform

- Need to view items but cannot book without registering
- Use the system to search and view rental items
- Frequency: Low usage
- Technical expertise: Minimal required

2.4 Operating Environment

- Web application accessible via standard web browsers (Chrome, Firefox, Safari, Edge)
- Responsive design for desktop and mobile devices
- Server environment: Linux-based hosting with HTTPS support
- Database: PostgreSQL or MySQL
- Compatible with modern operating systems (Windows 10+, macOS 10.15+, iOS 14+, Android 10+)

2.5 Design and Implementation Constraints

- Development timeframe limited by course schedule (approximately 8-10 weeks)
- Team size limited to assigned group members
- Development must use Git for version control
- Must implement responsive web design for mobile compatibility
- Payment gateway must comply with PCI DSS standards
- User data storage must comply with relevant privacy regulations
- Implementation should use modern web technologies (HTML5, CSS3, JavaScript)
- Backend implementation language will be determined by team's expertise

2.6 User Documentation

The following user documentation will be provided:

- Online help guide accessible within the application
- FAQ section addressing common questions and issues
- Video tutorials for key functions (listing an item, booking a rental)
- User guides for both item owners and renters
- Admin documentation for system administrators

2.7 Assumptions and Dependencies

- Users have access to a device with internet connectivity
- Users have basic familiarity with e-commerce websites
- Third-party payment processing services are available and functioning

- Email services are available for sending notifications
- Users have the means to upload digital images of their items
- Legal framework allows for peer-to-peer rental transactions
- Depends on availability of map/location services for item proximity search

3. External Interface Requirements

3.1 User Interfaces

The RentEase platform will feature a clean, intuitive user interface with the following characteristics:

- Responsive design that works across desktop and mobile devices
- Navigation menu with the following main sections:
 - Home/Dashboard
 - Browse Items
 - My Listings (for owners)
 - My Rentals (for renters)
 - Messages
 - Profile

Key screens include:

1. **Home Page:** Featured items, categories, and search functionality
2. **Item Detail Page:** Photos, description, availability calendar, pricing, owner details
3. **User Profile:** Personal information, ratings, rental history
4. **Listing Management:** Form for adding and editing item listings
5. **Booking Page:** Rental period selection, payment information, terms acceptance
6. **Dashboard:** Overview of active rentals, pending requests, and notifications

Standard elements across screens:

- Header with logo, search bar, and user account menu
- Footer with links to help, about, terms of service, and contact
- Consistent color scheme and typography following the design system
- Confirmation dialogues for important actions
- Toast notifications for system messages

3.2 Hardware Interfaces

The system does not require specific hardware interfaces beyond standard computing devices:

- Standard input devices (keyboard, mouse, touchscreen)
- Cameras for uploading item photos (mobile devices or webcams)
- Standard display screens with minimum resolution of 320px width

3.3 Software Interfaces

RentEase will interface with the following external software systems:

1. Payment Gateway API

- Purpose: Process rental payments and security deposits
- Data exchanged: Payment information, transaction status
- Format: REST API with JSON data format
- Timing constraints: Real-time processing with timeout handling

2. Email Service API

- Purpose: Send notifications and alerts to users
- Data exchanged: Email content, recipient information
- Format: SMTP or provider-specific API
- Timing constraints: Near real-time delivery expected

3. User Authentication Service

- Purpose: Support OAuth login options
- Data exchanged: User credentials, authentication tokens
- Format: OAuth 2.0 protocol
- Timing constraints: Real-time authentication required

3.4 Communications Interfaces

The system will utilize the following communication protocols:

HTTPS :

- All web communication will be secured using TLS/SSL
- Certificates must be valid and renewed appropriately

4. System Features

4.1 User Registration and Profile Management

4.1.1 Description and Priority

This feature enables users to create accounts, manage their profiles, and build their reputation within the system. It is of **High** priority as it is fundamental to the platform's functionality.

4.1.2 User Stories

- As a new user, I want to register for an account so that I can rent items and list my own items for rent.
- As a registered user, I want to create and manage my profile so that I can build trust with other users.
- As a user, I want to view other users' profiles and ratings so that I can decide whether to rent to/from them.
- As a user, I want to update my contact information so that I can receive important notifications.

4.1.3 Functional Requirements

REQ-F1.1: The system shall allow users to register using email address or OAuth providers (Google, Facebook).

REQ-F1.2: The system shall require email verification before activating a new account.

REQ-F1.3: The system shall allow users to create and edit their profiles, including:

- Profile picture
- Name (First and Last)
- Contact information (email, phone)
- Brief bio/description
- Location information

REQ-F1.4: The system shall display a user's rental history and ratings on their profile page.

REQ-F1.5: The system shall provide password reset functionality.

REQ-F1.6: The system shall allow users to deactivate their accounts.

REQ-F1.7: The system shall provide options for notification preferences.

4.2 Item Listing and Management

4.2.1 Description and Priority

This feature enables users to create, edit, and manage rental listings for their items. It is of **High** priority as it constitutes the core inventory of the marketplace.

4.2.2 User Stories

- As an item owner, I want to create detailed listings for my items so that potential renters can find them.
- As an item owner, I want to set rental prices and availability for my items so that I can control when they are rented.
- As an item owner, I want to edit or remove my listings so that I can keep my information accurate and current.
- As an item owner, I want to upload multiple photos of my item so that renters can see it clearly.

4.2.3 Functional Requirements

REQ-F2.1: The system shall provide a form for users to create item listings with the following information:

- Item title
- Category
- Description
- Photos (up to 10)
- Rental price (daily, weekly, monthly options)
- Security deposit amount
- Location
- Availability calendar
- Pickup/delivery options

REQ-F2.2: The system shall allow users to edit or delete their own listings.

REQ-F2.3: The system shall allow users to mark items as temporarily unavailable without removing the listing.

REQ-F2.4: The system shall validate that required fields are completed before publishing a listing.

REQ-F2.5: The system shall optimize and store uploaded images.

REQ-F2.6: The system shall provide a dashboard view of all listings owned by a user.

REQ-F2.7: The system shall allow users to duplicate existing listings to create similar ones.

4.3 Rental Booking and Processing

4.3.1 Description and Priority

This feature enables the core rental transaction process, from requesting to returning items. It is of **High** priority as it represents the primary function of the platform.

4.3.2 User Stories

- As a renter, I want to request to rent an item for specific dates so that I can use it when needed.
- As an item owner, I want to approve or decline rental requests so that I can control who uses my items.
- As a renter, I want to receive confirmation when my rental request is approved so that I can plan accordingly.
- As a user, I want to track the status of my ongoing rentals so that I know when to return or expect items.
- As a renter, I want to extend my rental period if needed so that I can use the item longer.

4.3.3 Functional Requirements

REQ-F3.1: The system shall allow users to request rentals by selecting:

- Start and end dates
- Pickup or delivery preference
- Special requests or notes

REQ-F3.2: The system shall notify item owners of pending rental requests.

REQ-F3.3: The system shall allow item owners to approve or decline rental requests.

REQ-F3.4: The system shall notify renters of request approval or declination.

REQ-F3.5: The system shall generate a rental agreement when a request is approved.

REQ-F3.6: The system shall track the status of rentals (requested, approved, in progress, completed, cancelled).

REQ-F3.7: The system shall allow renters to request rental extensions.

REQ-F3.8: The system shall allow item owners to confirm item return and condition.

REQ-F3.9: The system shall maintain a history of all rental transactions.

REQ-F3.10: The system shall allow users to cancel rentals with different policies depending on timing.

4.4 Payment Processing

4.4.1 Description and Priority

This feature handles all financial transactions on the platform, including rental payments and security deposits. It is of **High** priority as it ensures the economic viability of the platform.

4.4.2 User Stories

- As a renter, I want to securely pay for my rentals so that I can access the items I need.
- As an item owner, I want to receive payments when my items are rented so that I can earn money.
- As a renter, I want my security deposit returned when I return an item in good condition so that I don't lose money.
- As an item owner, I want to claim the security deposit when an item is damaged so that I can cover repair costs.

4.4.3 Functional Requirements

REQ-F4.1: The system shall securely collect and store payment method information.

REQ-F4.2: The system shall process rental payments when a rental request is approved.

REQ-F4.3: The system shall hold security deposits separately from rental payments.

REQ-F4.4: The system shall automatically release security deposits to renters upon successful item return.

REQ-F4.5: The system shall allow item owners to file claims against security deposits for damaged items.

REQ-F4.6: The system shall provide payment receipts and transaction history.

REQ-F4.7: The system shall support multiple payment methods (credit/debit cards, PayPal, etc.).

REQ-F4.8: The system shall handle refunds for cancelled rentals according to cancellation policies.

REQ-F4.9: The system shall calculate and display platform fees transparently.

4.5 Review and Rating System

4.5.1 Description and Priority

This feature enables users to rate and review each other after completed transactions. It is of **Medium** priority as it builds trust within the community.

4.5.2 User Stories

- As a renter, I want to rate and review item owners so that others know about my experience.
- As an item owner, I want to rate and review renters so that other owners know about their reliability.

- As a user, I want to see ratings and reviews for potential rental partners so that I can make informed decisions.
- As a user, I want to respond to reviews about me so that I can address any concerns.

4.5.3 Functional Requirements

REQ-F5.1: The system shall prompt users to review each other after a rental is completed.

REQ-F5.2: The system shall allow users to rate each other on a 5-star scale.

REQ-F5.3: The system shall allow users to write text reviews to accompany ratings.

REQ-F5.4: The system shall display average ratings on user profiles.

REQ-F5.5: The system shall allow users to report inappropriate reviews.

REQ-F5.6: The system shall allow users to respond to reviews about them.

REQ-F5.7: The system shall prevent users from reviewing others without completing a transaction.

REQ-F5.8: The system shall calculate and display the number of successful transactions for each user.

4.6 Search and Browse Functionality

4.6.1 Description and Priority

This feature enables users to discover items available for rent. It is of **High** priority as it facilitates finding relevant items efficiently.

4.6.2 User Stories

- As a renter, I want to search for specific items so that I can find what I need quickly.
- As a renter, I want to filter search results by location, price, and category so that I can find relevant options.
- As a renter, I want to browse items by category so that I can discover available options.
- As a renter, I want to see items near my location so that pickup is convenient.

4.6.3 Functional Requirements

REQ-F6.1: The system shall provide keyword search functionality for items.

REQ-F6.2: The system shall allow filtering search results by:

- Category
- Price range
- Location/distance
- Availability dates
- Rating
- Delivery options

REQ-F6.3: The system shall display search results in a grid or list view with thumbnail images.

REQ-F6.4: The system shall provide category browsing with hierarchical organization.

REQ-F6.5: The system shall support location-based searching using postal codes or geolocation.

REQ-F6.6: The system shall allow sorting of results by relevance, price, distance, and rating.

REQ-F6.7: The system shall provide pagination or infinite scrolling for search results.

REQ-F6.8: The system shall highlight featured or promoted listings in search results.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

REQ-NF1.1: The system shall load pages within 3 seconds under normal load conditions.

REQ-NF1.2: The system shall support at least 1000 concurrent users without degradation in performance.

REQ-NF1.3: The system shall process search queries and return results within 2 seconds.

REQ-NF1.4: The system shall process payment transactions within 5 seconds.

REQ-NF1.5: The system shall be able to handle peak loads during promotional periods with only 10% increase in response time.

REQ-NF1.6: The database shall be optimized to handle at least 100,000 item listings.

REQ-NF1.7: Image uploads shall be processed and displayed within 10 seconds, even for large files.

5.2 Safety Requirements

REQ-NF2.1: The system shall implement measures to prevent fraudulent listings and scams.

REQ-NF2.2: The system shall verify user identities through email confirmation at minimum.

REQ-NF2.3: The system shall protect against common web vulnerabilities (XSS, CSRF, SQL injection).

REQ-NF2.4: The system shall maintain logs of all transactions for dispute resolution.

REQ-NF2.5: The system shall implement content moderation for item descriptions and images.

5.3 Security Requirements

REQ-NF3.1: The system shall encrypt all sensitive user data, including personal information and payment details.

REQ-NF3.2: The system shall implement secure authentication with password complexity requirements.

REQ-NF3.3: The system shall use HTTPS for all communications.

REQ-NF3.4: The system shall implement session timeout after 30 minutes of inactivity.

REQ-NF3.5: The system shall comply with relevant data protection regulations.

REQ-NF3.6: The system shall implement rate limiting to prevent brute force attacks.

REQ-NF3.7: The system shall provide audit trails for all administrator actions.

REQ-NF3.8: The payment processing shall comply with PCI DSS standards.

5.4 Software Quality Attributes

REQ-NF4.1: **Usability:** The system shall be intuitive enough that new users can complete a basic rental transaction without training.

REQ-NF4.2: **Reliability:** The system shall have an uptime of at least 99.5%.

REQ-NF4.3: **Maintainability:** The system shall be built with modular architecture to facilitate future enhancements.

REQ-NF4.4: **Portability:** The system shall function correctly on all major browsers and device types.

REQ-NF4.5: **Scalability:** The system architecture shall allow for horizontal scaling to accommodate growth.

REQ-NF4.6: **Testability:** The system shall include automated testing for critical components.

REQ-NF4.7: **Accessibility:** The system shall comply with WCAG 2.1 Level AA standards.

5.5 Business Rules

BR-1: The platform shall collect a service fee of 10% on each rental transaction.

BR-2: Security deposits must be between 20-100% of the item's stated value.

BR-3: Rental prices must not exceed the item's stated market value.

BR-4: Users must be at least 18 years old to register.

BR-5: Cancellation policies shall be tiered based on timing:

- More than 7 days before: 100% refund
- 3-7 days before: 50% refund
- Less than 3 days before: No refund

BR-6: Owners cannot modify pricing or availability for approved rentals.

BR-7: Disputes must be filed within 48 hours of item return.

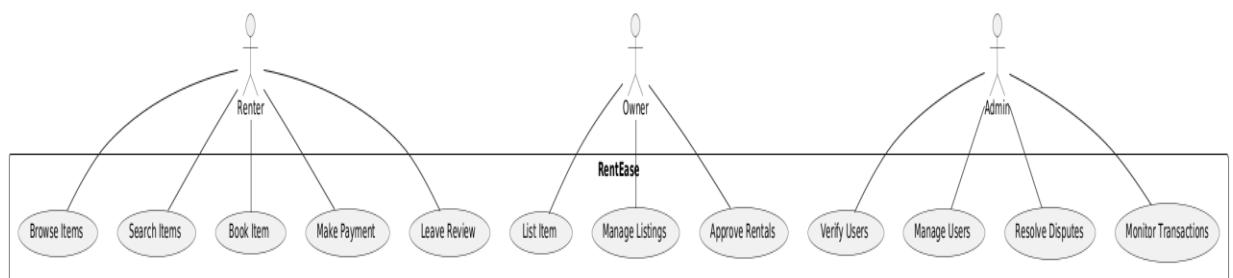
BR-8: Prohibited items (weapons, illegal items, etc.) may not be listed.

6. Diagrams

6.1 Use Case Diagram

A comprehensive use case diagram will be created to illustrate the interactions between users and the RentEase system, including:

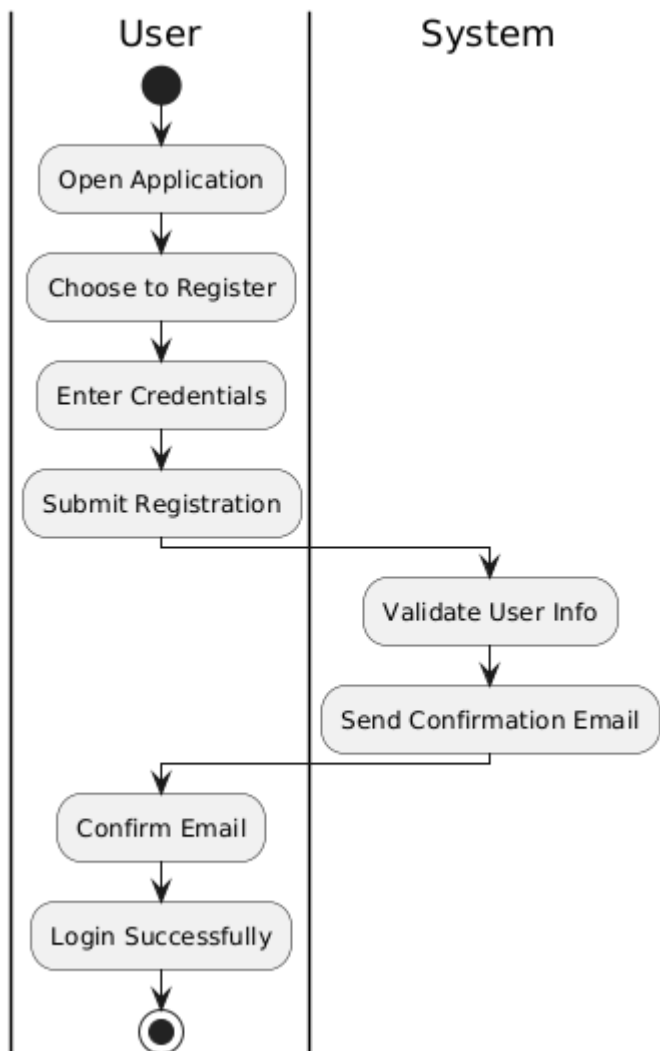
- User registration and authentication
- Item listing and management
- Rental booking and processing
- Payment transactions
- Review and rating
- Search and browsing functions



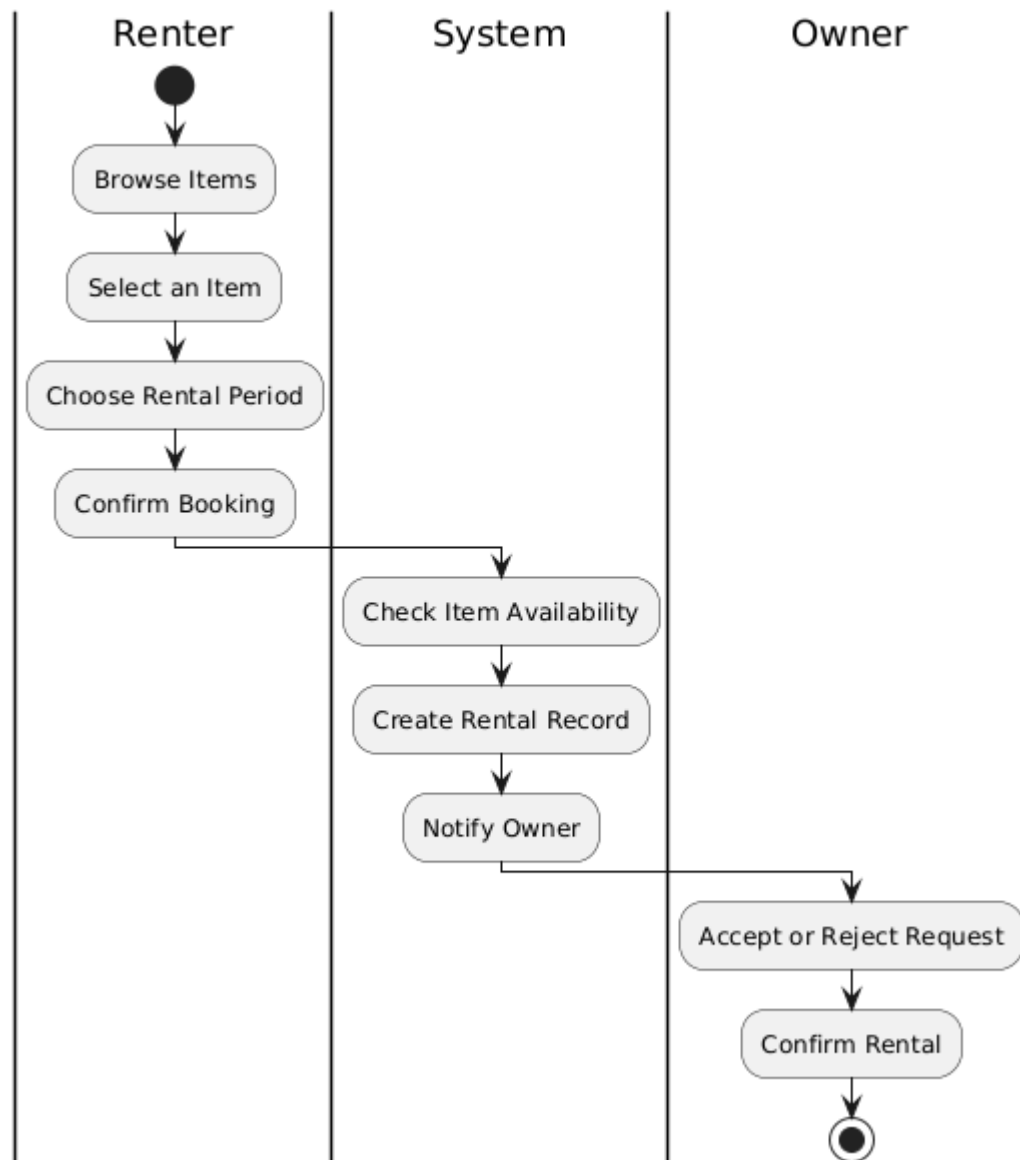
6.2 Activity Diagrams

Activity diagrams will be created for three main processes:

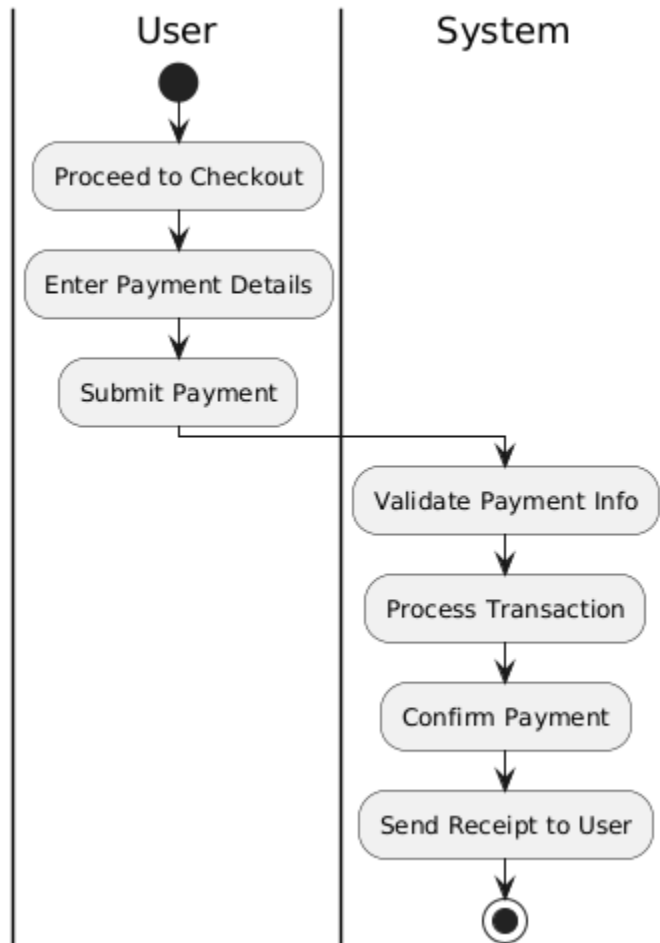
1. User Registration



2. Item Booking



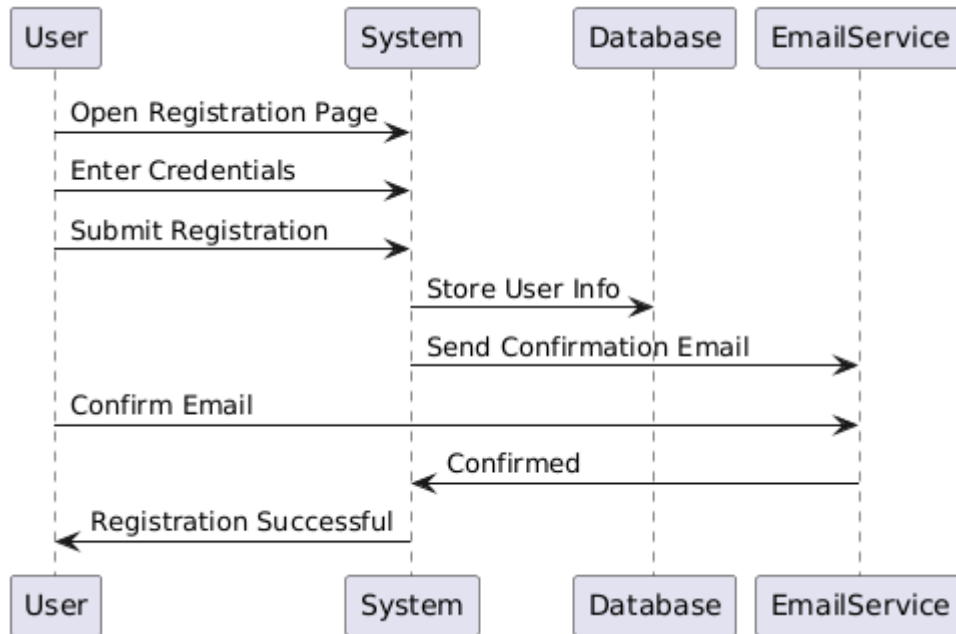
3. Payment processing



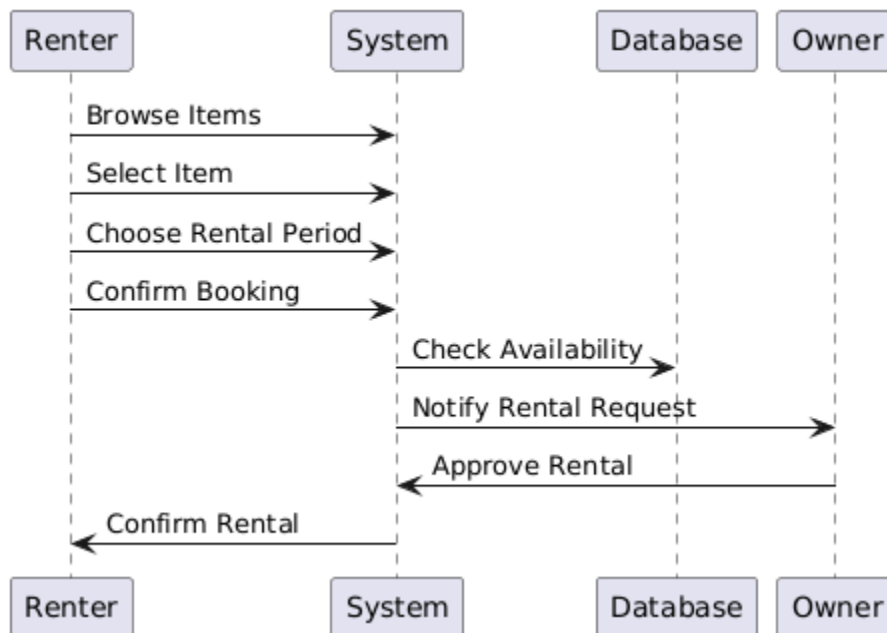
6.3 Sequence Diagrams

Sequence diagrams will be created for three main activities:

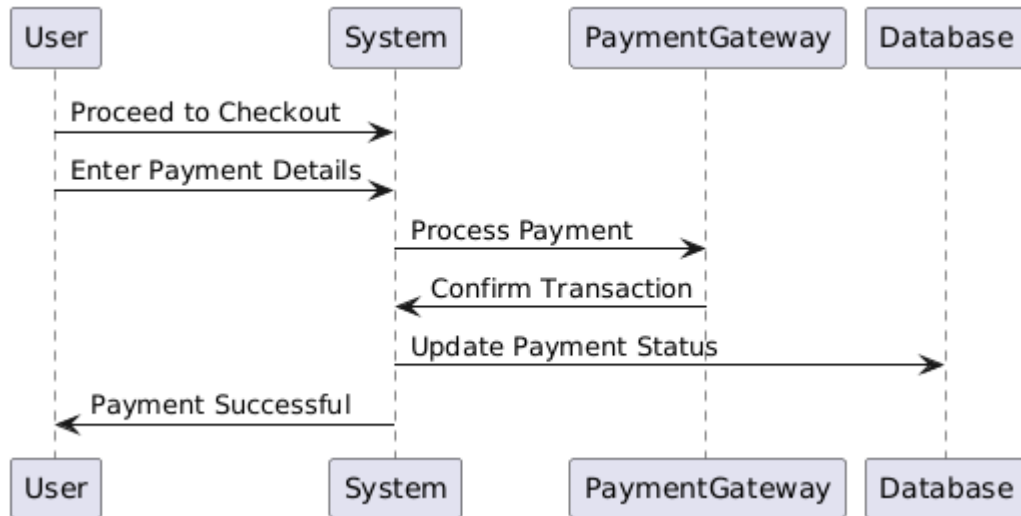
1. User Registration



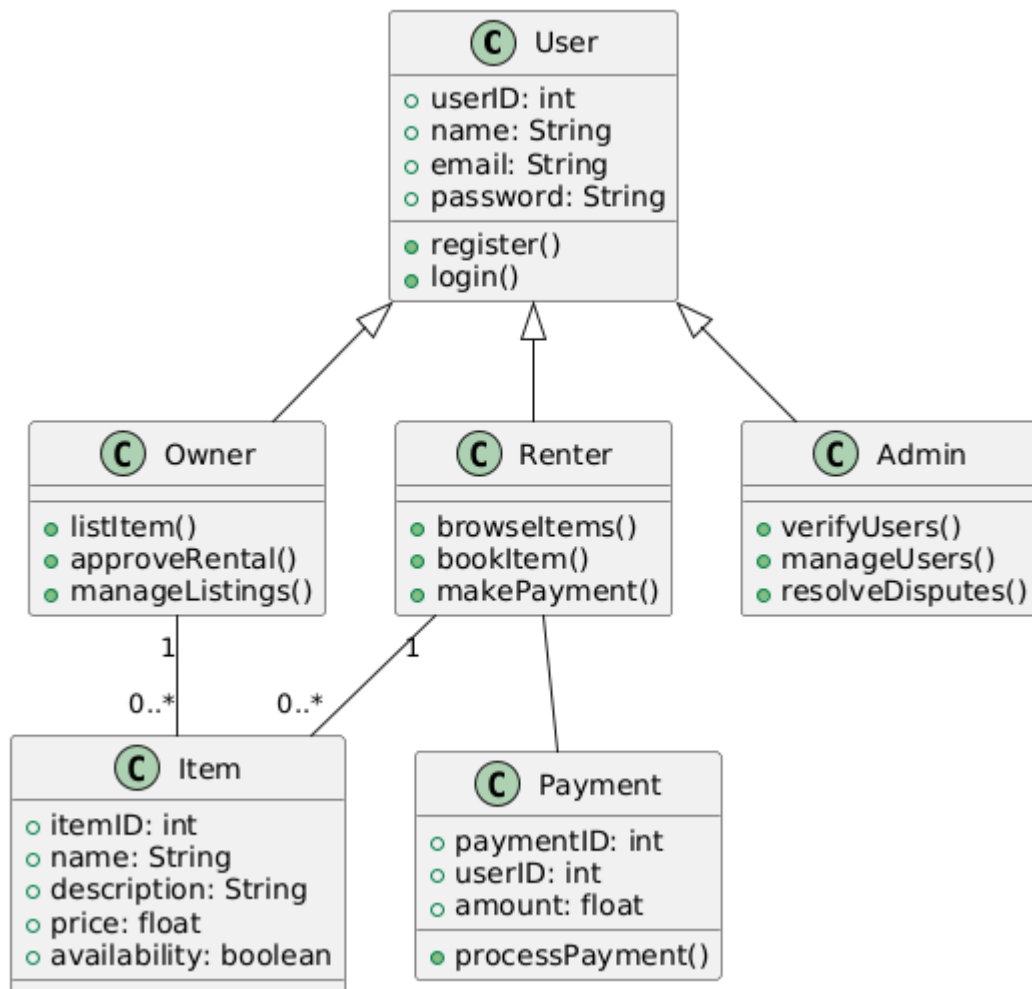
2. Item Booking



3. Payment Processing



6.4 Class Diagram



Appendix A: Glossary

- **Item Owner:** A user who lists items for rent on the platform
- **Renter:** A user who rents items from other users
- **Listing:** An item offered for rent, including its description, photos, and rental terms
- **Security Deposit:** A refundable amount held to cover potential damage to a rented item
- **Rental Period:** The duration for which an item is rented
- **Transaction Fee:** The percentage taken by the platform for facilitating rentals
- **Rating:** Numerical score given to users based on rental experiences
- **Review:** Written feedback about rental experiences
- **Dashboard:** Personalized user interface showing rentals, listings, and activity
- **OAuth:** Open standard for access delegation, used for third-party login

Appendix B: Analysis Models

Class diagrams and entity-relationship diagrams will be included in the final SRS document.

Appendix C: To Be Determined List

1. TBD-1: Specific payment gateway integration (to be selected)
2. TBD-2: Exact fee structure and transaction percentages
3. TBD-3: Legal requirements for rental agreements in target markets
4. TBD-4: Implementation details for messaging system
5. TBD-5: Specific technologies for backend development

Github repository Link: <https://github.com/ammarahmad01/RentEase>

Screenshots Trello Board

Trello

Workspaces ▾

Recent ▾

Starred ▾

Templates ▾

T

Trello Workspace

Premium

<

Boards

Members

+

Workspace settings

▾

PREMIUM

Workspace views

+

Table

Calendar

Your boards

+

RentEase

RentEase

☆

Workspace visible

Product Backlog

↔ ⋮

User Registration

User Login

List an Item for Rent

Browse items

Item Search

Item Booking

Payment Integration

Rental History

Review and Rating System

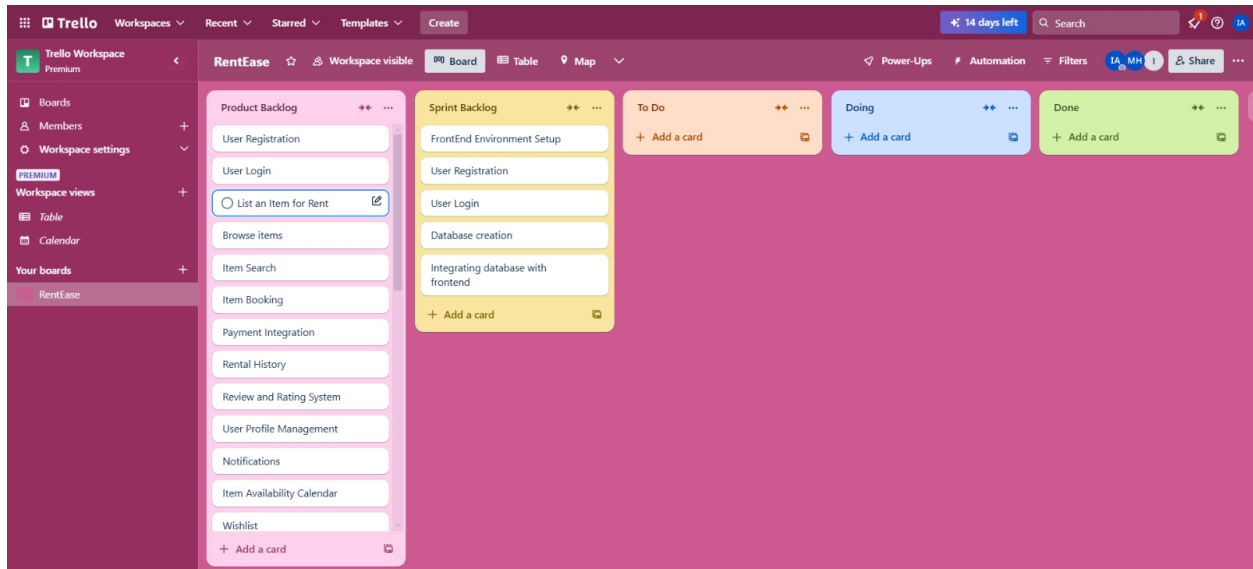
User Profile Management

Notifications

Item Availability Calendar

Wishlist

+ Add a card



Trello – Product Backlog, Sprint Backlog and LeftOver from Sprint 1

