[Borrow Box]

SDA PROJECT REPORT

Submitted by :
22i-2371 & 22i-2340
Section J
Date 24/11/2024

Project Name: Borrow Box: A Peer-to-Peer Product Rental Platform

Table of Contents

1. Introduction

1.1 Purpose

The Borrow Box system is designed to facilitate seamless and efficient rental management for users and product owners. It enables users to rent, view, and manage products digitally while providing an organized and user-friendly platform for product owners to list and manage their inventory.>

1.2 Product Scope

Borrow Box is intended to serve as a complete solution for small-scale rental businesses. It allows users to browse available products, request rentals, leave feedback, and manage their rental history. Product owners can list items for rent, track active rentals, and review feedback to improve service quality

1.3 Title

Borrow Box: A Peer-to-Peer Product Rental Platform

1.4 Objectives

- Create a user-friendly platform for listing rental items.
- Provide secure and reliable transactions.
- Keeps records of rentals
- Enable notifications for product availability.
- Enhance the user experience through an intuitive and responsive interface.
- Enable product owners to manage listings and track rental activity effectively

1.5 Problem Statement

Traditional rental systems rely heavily on manual operations, resulting in inefficiencies, mismanagement, and limited scalability. Borrow Box resolves these challenges by digitizing rental operations, offering real-time updates, and ensuring secure transaction

2. Overall Description

2.1 Product Perspective

Borrow Box integrates essential features to streamline the rental process. It supports multi-user roles (Admin, Owner, User) and facilitates seamless communication between renters and owners. The system ensures data integrity, real-time availability checks, and reliable payment methods.

2.2 Product Functions

- User Registration:
- Product Listing:
- Rental Requests:.
- Payment Processing:
- Feedback System:
- User Login:
- Listing Products for Rent:
- Renting Products:
- Viewing Rental History:
- Generating Reports (Admin):
- Providing Feedback:
- Making Payments for Rentals:
- Reporting Product Issues:
- Admin Dashboard Management:

2.3 List of Use Cases

- Register a User
- Login a User:
- view product list
- list a product for rent
- verify user identity
- make payment for rentals
- leave a feedback for products
- cart management
- manage products
- database management
- UI navigation
- Admin Management

2.4 Extended Use Cases

1. **Register a User**

   Use Case Name: Register a User
   Scope: Online Rental System
   Level: User Goal

Primary Actor: User
Stakeholders and Interests:

1. **User:** Wants to create an account to access the system.
2. **System Owner:** Wants to increase the user base.

Preconditions: The user must have a valid email address.

Post conditions: A user account is created, and the user receives a confirmation email.

Main Success Scenario

Actor Action (or Intention)	**System Responsibility**
1. User navigates to the registration page.	2. System provides the registration form.
3. User fills in email, username, and password.	4. System validates the input data.
5. User submits the registration form.	6. System calls UserService.registerUser() to save the data in the database.
System validates email uniqueness and password confirmation.	7 System displays success or error messages.

Extensions:

- **1a. Invalid Email Format:**
  - User enters an invalid email.
  - System prompts user to correct the email format.
- **2a. Username Already Taken:**
  - User submits the form with an existing username.
  - System notifies the user that the username is taken.

2. log In User (Muhammad Fawaz)

Use Case Name: log In User
Scope: Online Rental System
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

User: Wants to log in to their account to access platform features.

System Owner: Wants to ensure secure logins.

Preconditions: The user must have a registered account with a valid email and password.

Postconditions: The user is logged into the system and has access to their account.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. User navigates to the "Login" page.	System provides fields for username and password.
2. User enters username and password.	System calls UserService.validateLogin() to check credentials.
3. User submits the form.	System checks user role (ADMIN or USER) using UserService.getUserRole().
4. If credentials are valid, system redirects user:	- To AdminDashboard for ADMIN.
	To ModeSelectionPage for USER.

Extensions:

1a. Invalid Email or Password:

- User enters an incorrect email or password.
- System displays an error message and prompts the user to retry or reset their password.

1b. Forgot Password:

- User clicks on the “Forgot Password” link.
- System redirects the user to the password recovery page to reset their password.

3. List a Product for Rent

Use Case Name: List a Product for Rent
Scope: Online Rental System
Level: User Goal

Primary Actor: User (Product Owner)
Stakeholders and Interests:

User: Wants to list their product for rent.

System Owner: Wants to increase the number of products available for rent.

Preconditions: User must be logged into their account.

Post conditions: The product is listed and available for rental.

Main Success Scenario

Actor

Action (or Intention) **System Responsibility**

1. Owner navigates to the "Add Product" page.	System displays a form for product details.
2. Owner fills in product details (name, description, price, etc.).	System validates the input fields.
3. Owner submits the form.	3. Owner submits the form.
4. System saves the product and marks it as available.	System displays a success or error messag

Extensions:

1a. Missing Required Fields:

- User submits the form without filling all required fields.
- System prompts the user to complete all required fields.

2a. Invalid Price Format:

- User enters a price in an invalid format.
- System notifies the user to enter a valid price.

4. View Product List

Use Case Name: View Product List
Scope: Online Rental System
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

1. **User:** Wants to view the list of available products for rent**.**
2. **System Owner:** Wants to provide an easy and efficient way for users to browse available products.

Preconditions: The user must be logged into the system.
Postconditions: The user can see the list of available products.

Main Success Scenario

Actor

Action (or Intention) **System Responsibility**

1. User navigates to the product list page.	2. System displays the product list page.
3. User requests to view the list of products.	4. System retrieves the list of available products from the database.
5. User views the list of products.	6. System displays the list of products with details such as name, price, and availability.

Extensions:

Scenario	Action
1a. No Products Available:	System displays a message indicating that no products are available for rent.
2a. Database Connection Error:	System displays an error message indicating that the product list could not be retrieved.

5. Verify User Identity

Use Case Name: Verify User Identity

Scope: Online Rental System

Level: User Goal

Primary Actor: User

Stakeholders and Interests:

User: Wants to verify their identity to access secure features.

System Owner: Wants to ensure that only authenticated users can access certain features.

Preconditions: The user must be logged into the system.

Postconditions: The user's identity is verified, and they can access secure features.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. User navigates to a secure feature.	2. System prompts the user to verify their identity.
3. User provides identity verification information (e.g., password, OTP).	4. System validates the provided information.
5. User submits the verification information.	6. System verifies the user's identity and grants access to the secure feature.

Extensions:

Scenario	Action
Scenario	Action
1a. Invalid Verification Information:	System displays an error message indicating that the verification information is invalid and prompts the user to try again.
2a. Verification Timeout:	System displays a message indicating that the verification process has timed out and prompts the user to try again.

6. Make Payment for Rentals

Use Case Name: Make Payment for Rentals
Scope: Online Rental System
Level: User Goal

Primary Actor: User
Stakeholders and Interests:

1. **User:** Wants to make a payment for rented products.
2. **System Owner:** Wants to ensure that payments are processed securely and efficiently.

Preconditions: The user must have items in their cart and be logged into the system.
Postconditions: The payment is processed, and the rental transaction is completed.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. User navigates to the checkout page.	2. System displays the checkout page with the list of items to be rented.
3. User reviews the rental items and total cost.	4. System calculates the total cost and displays it.
5. User provides payment information (e.g., credit card details).	6. System validates the payment information.
7. User submits the payment.	8. System processes the payment through the payment gateway.
9. User receives a payment confirmation.	10. System confirms the payment and updates the rental status in the database.
	11. System sends a payment receipt to the user's email.

Extensions:

1a. Invalid Payment Information:	System displays an error message indicating that the payment information is invalid and prompts the user to correct it.
2a. Payment Processing Error:	System displays an error message indicating that the payment could not be processed and prompts the user to try again.
3a. Insufficient Funds:	System displays a message indicating that there are insufficient funds and prompts the user to use a different payment method.
1a. Invalid Payment Information:	System displays an error message indicating that the payment information is invalid and prompts the user to correct it.

7. Leave Feedback for Products

Use Case Name: Leave Feedback for Products
Scope: Online Rental System
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

1. **User:** Wants to leave feedback for a product they have rented.
2. **System Owner:** Wants to collect user feedback to improve product offerings and service quality.

Preconditions: The user must be logged into the system and have rented the product.
Postconditions: The feedback is saved in the system and associated with the product.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. User navigates to the feedback page.	2. System displays the feedback form.
3. User selects the product they want to leave feedback for.	4. System retrieves the product details.
5. User fills in the feedback form with rating and comments.	6. System validates the input data.
7. User submits the feedback form.	8. System calls FeedbackService.saveFeedback() to save the feedback in the database.
9. User receives a confirmation message.	10. System displays a success message indicating that the feedback has been submitted successfully.

Extensions:

Scenario	Action
1a. Invalid Input Data:	System displays an error message indicating that the input data is invalid and prompts the user to correct it.
2a. Feedback Submission Error:	System displays an error message indicating that the feedback could not be submitted and prompts the user to try again.
Scenario	Action

8. Cart Management

Use Case Name: Cart Management
Scope: Online Rental System
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

1. **User:** Wants to manage the items in their cart before proceeding to checkout.
2. **System Owner:** Wants to provide a seamless and efficient cart management experience to encourage rentals.

Preconditions: The user must be logged into the system.
Postconditions: The cart is updated based on the user's actions.

Main Success Scenario

1. User navigates to the cart page.	2. System displays the cart page with the list of items in the cart.
3. User adds a product to the cart.	4. System adds the selected product to the cart and updates the cart display.

Extensions:

1a. Cart Update Error:	System displays an error message indicating that the cart could not be updated and prompts the user to try again.

9. Manage Products

Use Case Name: Manage Products
Scope: Online Rental System
Level: User Goal
Primary Actor: Owner
Stakeholders and Interests:

1. **Owner:** Wants to manage the products available for rent.
2. **System Owner:** Wants to ensure that the product catalog is up-to-date and accurate.

Preconditions: The owner must be logged into the system.
Postconditions: The product catalog is updated based on the owner's actions.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. Owner navigates to the manage products page.	2. System displays the manage products page with the list of products.
3. Owner clicks on "Add Product" button.	4. System displays a form to enter product details.
5. Owner fills in the product details and submits the form.	6. System validates the input data and saves the new product to the database.

Extensions:

Scenario	Action
1a. No Products Available:	System displays a message indicating that no products are available for rent.
2a. Database Connection Error:	System displays an error message indicating that the product list could not be retrieved.

10. Database Management

Use Case Name: Database Management
Scope: Online Rental System
Level: System Goal
Primary Actor: System
Stakeholders and Interests:

1. **System Owner:** Wants to ensure that the database is properly managed and accessible.
2. **Developers:** Want to ensure that the database operations are performed efficiently and correctly.

Preconditions: The system must be running and have access to the database.
Postconditions: The database operations are performed successfully.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. System attempts to connect to the database.	2. System establishes a connection to the database.
3. System executes a database query.	4. System retrieves the query results from the

	database.
5. System processes the query results.	6. System updates the database based on the processed results.
7. System closes the database connection.	8. System ensures that the connection is properly closed and resources are released.

Extensions:

Scenario	Action
1a. Database Connection Failed:	System displays an error message indicating that the database connection failed and logs the error.
2a. Query Execution Error:	System displays an error message indicating that the query execution failed and logs the error.
3a. Data Retrieval Error:	System displays an error message indicating that data retrieval failed and logs the error.

11. UI Navigation

Use Case Name: UI Navigation
Scope: Online Rental System
Level: User Goal
Primary Actor: User
Stakeholders and Interests:

1. **User:** Wants to navigate through different pages of the application easily.
2. **System Owner:** Wants to provide a seamless and intuitive navigation experience for users.

Preconditions: The user must be logged into the system.
Postconditions: The user successfully navigates to the desired page.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. User clicks on a navigation link or button.	2. System captures the navigation request.
3. User requests to navigate to a specific page.	4. System loads the requested page.
5. User views the requested page.	6. System displays the requested page with the relevant content.

Extensions:

Scenario	Action
1a. Page Not Found:	System displays an error message indicating that the requested page could not be found.
2a. Navigation Error:	System displays an error message indicating that there was an error during navigation and prompts the user to try again.

12 Admin management

Use Case Name: Admin management
Scope: Online Rental System
Level: User Goal
Primary Actor: Admin
Stakeholders and Interests:

1. **Admin:** Wants to manage users, manage products, view reports, and perform administrative tasks.
2. **System Owner:** Wants to ensure that the system is properly managed and maintained.

Preconditions: The admin must be logged into the system with appropriate permissions.
Postconditions: The admin successfully performs administrative tasks.

Main Success Scenario

Actor Action (or Intention)	System Responsibility
1. Admin navigates to the admin dashboard.	2. System displays the admin dashboard with available options.
3. Admin selects "Manage Users".	4. System navigates to the Manage Users page.
5. Admin selects "Manage Products".	6. System navigates to the Manage Products page.
7. Admin selects "View Reports".	8. System navigates to the View Reports page.
9. Admin selects "Logout".	10. System logs out the admin and navigates to the Login page.

Extensions:

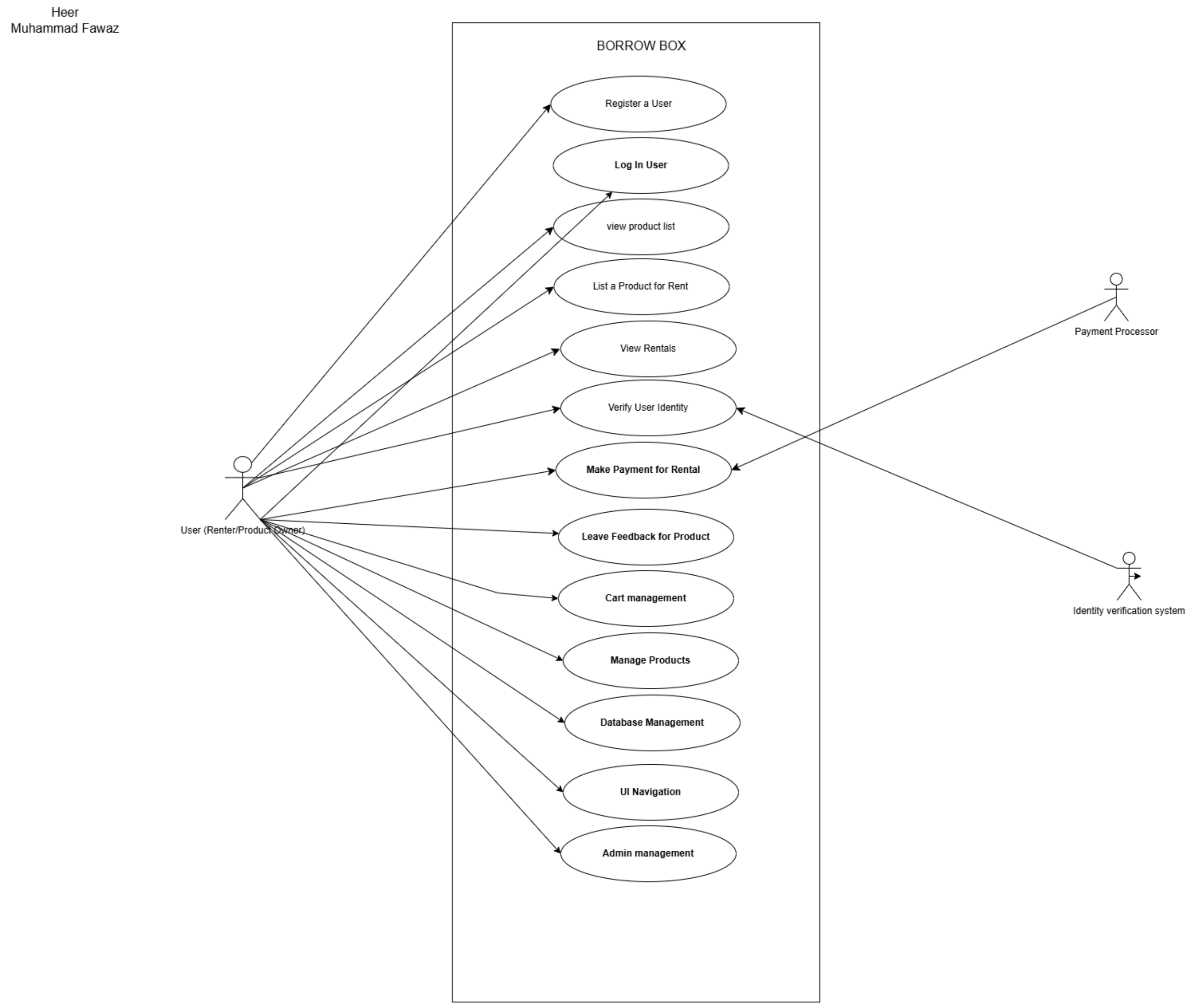
Scenario	Action

Actor Action (or Intention)	System Responsibility
1. User navigates to the registration page.	2. System provides the registration form.
3. User fills in email, username, and password.	4. System validates the input data.
5. User submits the registration form.	6. System creates the user account.
	7. System sends a confirmation email to the user.

Extensions:

- **1a. Invalid Email Format:**
  1. User enters an invalid email.
  2. System prompts user to correct the email format.
- **2a. Username Already Taken:**
  1. User submits the form with an existing username.
  2. System notifies the user that the username is taken.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

1. The system should process user login within 2 seconds.
2. Search for products should return results within 3 seconds.
3. The system should support up to 100 concurrent users without performance degradation.
4. Data retrieval from the database should not exceed 2 seconds per query.

3.2 Safety Requirements

1. The system should provide an error recovery mechanism in case of unexpected shutdowns.
2. Data entered during a session must not be lost due to abrupt termination or power failure.
3. Only verified users can access the system to prevent misuse.

3.3 Security Requirements

1. All passwords should be stored in a hashed format in the database.
2. Implement SSL/TLS encryption for data transmitted over the network.
3. Users should be locked out of their accounts after 5 failed login attempts to prevent brute-force attacks.
4. Admin access must be restricted and verified with two-factor authentication (2FA).

3.4 Software Quality Attributes

1. **Usability:** The system should provide a user-friendly and intuitive interface for users with minimal technical expertise.
2. **Reliability:** The system should have an uptime of 99.9%.
3. **Scalability:** The system should handle a growing number of users and transactions without a significant decrease in performance.
4. **Maintainability:** The system should be modular, allowing developers to update components without impacting the entire system.

3.5 Business Rules

1. Each product listed for rental must belong to a registered owner.
2. Products marked as unavailable should not appear in search results.

3. Rentals must be approved by the owner before being marked as active.
4. Late returns will incur a penalty fee, which is calculated automatically.

3.6 Operating Environment

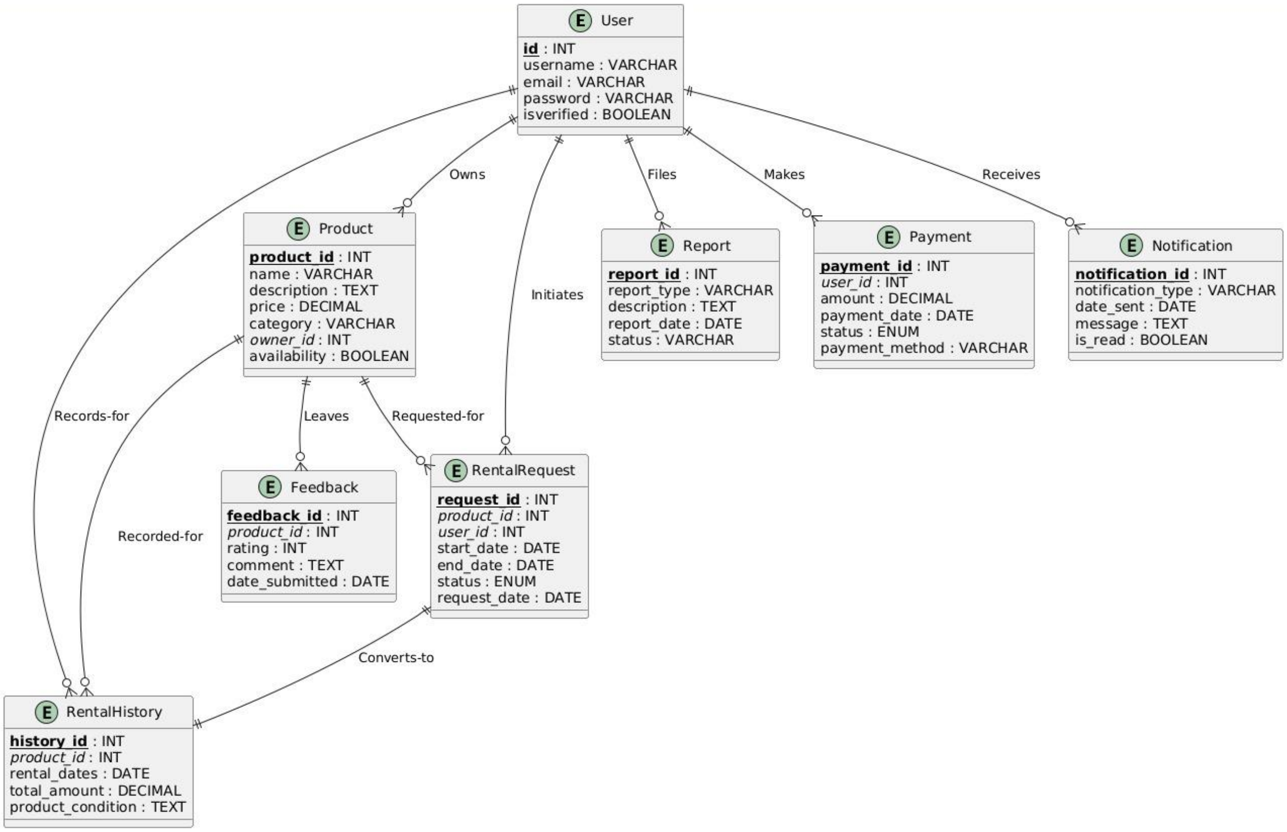
1. The system should be compatible with Windows, macOS, and Linux operating systems.
2. The system should support major browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
3. The database will run on MySQL Server version 8.0 or higher.

3.7 User Interfaces

1. A dashboard for product owners to manage their listings.
2. A cart system for users to view and check out their selected products.
3. Admin tools for viewing and managing rentals, user accounts, and reports.

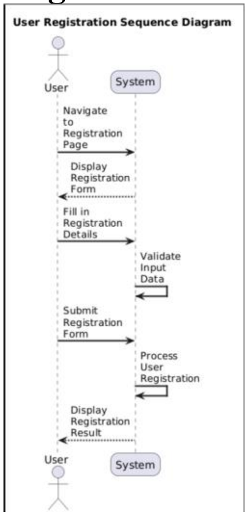
Let me know if you want to adjust or expand on any of these points!

4. Domain Model

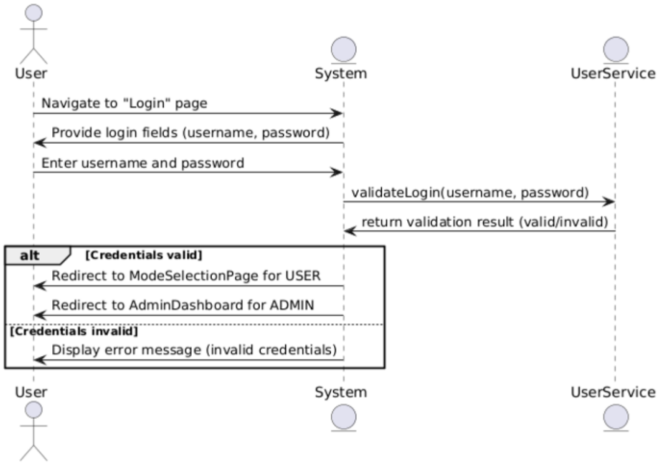


5. System Sequence Diagram

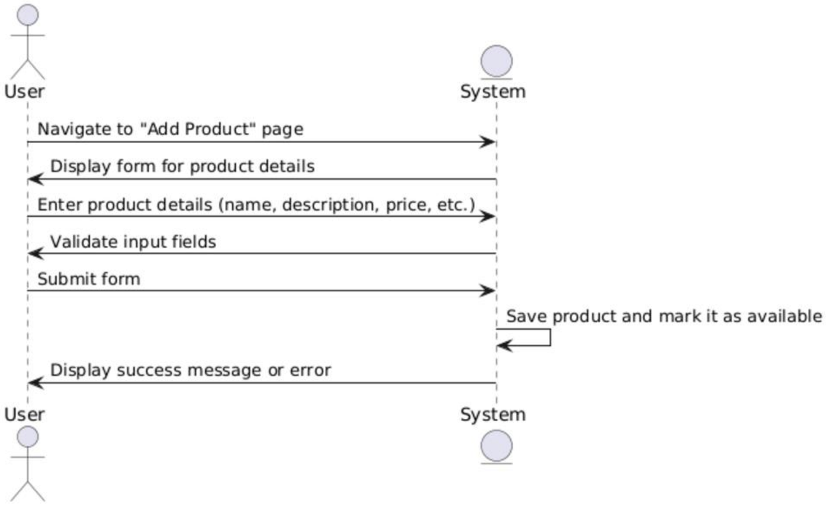
Register a user


User Registration Sequence Diagram
User
System
Navigate to Registration Page
Display Registration Form
Fill in Registration Details
Validate Input Data
Submit Registration Form
Process User Registration
Display Registration Result
User
System

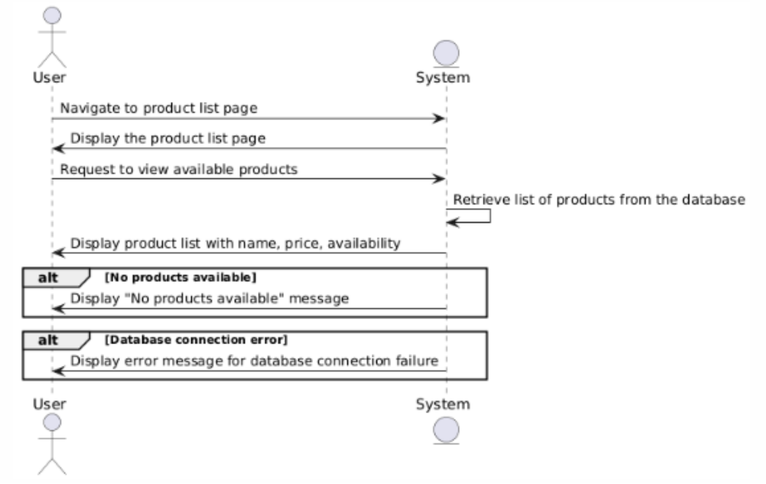
Login user


User
System
UserService
Navigate to "Login" page
Provide login fields (username, password)
Enter username and password
validateLogin(username, password)
return validation result (valid/invalid)
alt
[Credentials valid]
Redirect to ModeSelectionPage for USER
Redirect to AdminDashboard for ADMIN
[Credentials invalid]
Display error message (invalid credentials)
User
System
UserService

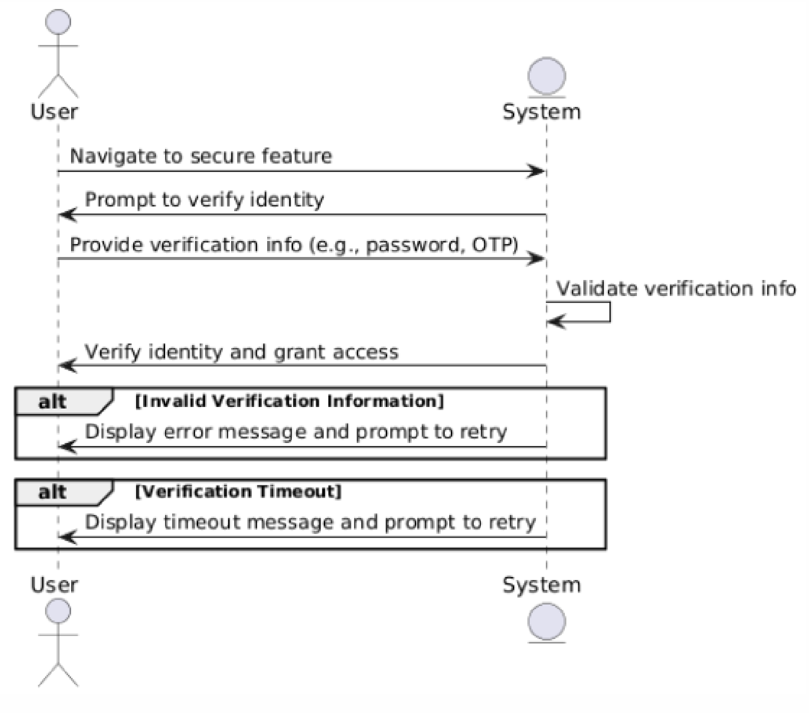
List a product for rent


User
System
Navigate to "Add Product" page
Display form for product details
Enter product details (name, description, price, etc.)
Validate input fields
Submit form
Save product and mark it as available
Display success message or error
User
System

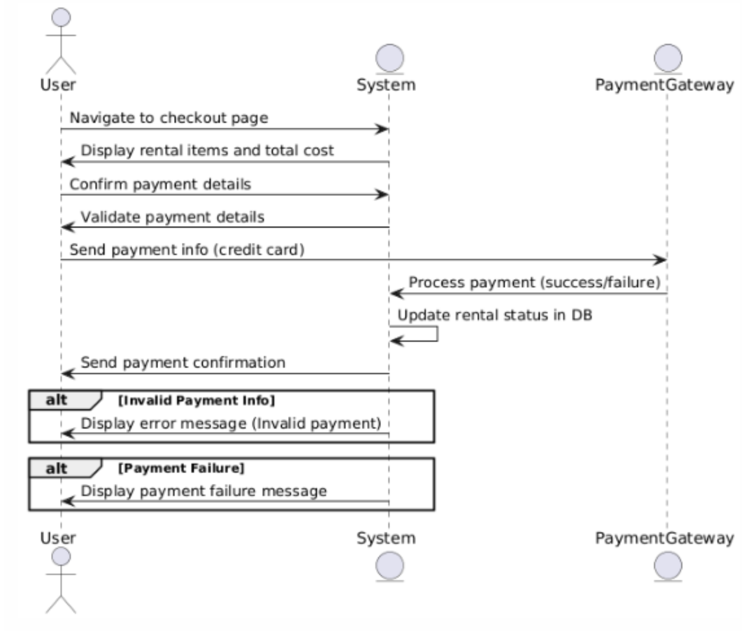
View product list


User
System
Navigate to product list page
Display the product list page
Request to view available products
Retrieve list of products from the database
Display product list with name, price, availability
alt
[No products available]
Display "No products available" message
alt
[Database connection error]
Display error message for database connection failure
User
System

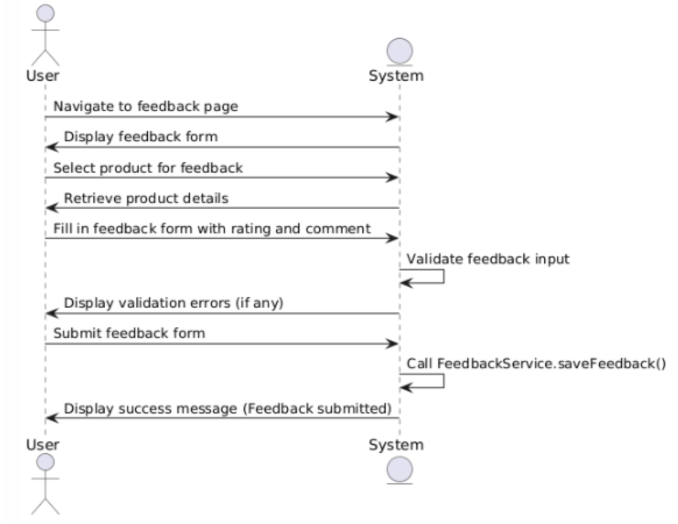
Verify User Identity


User
System
Navigate to secure feature
Prompt to verify identity
Provide verification info (e.g., password, OTP)
Validate verification info
Verify identity and grant access
alt
[Invalid Verification Information]
Display error message and prompt to retry
alt
[Verification Timeout]
Display timeout message and prompt to retry
User
System

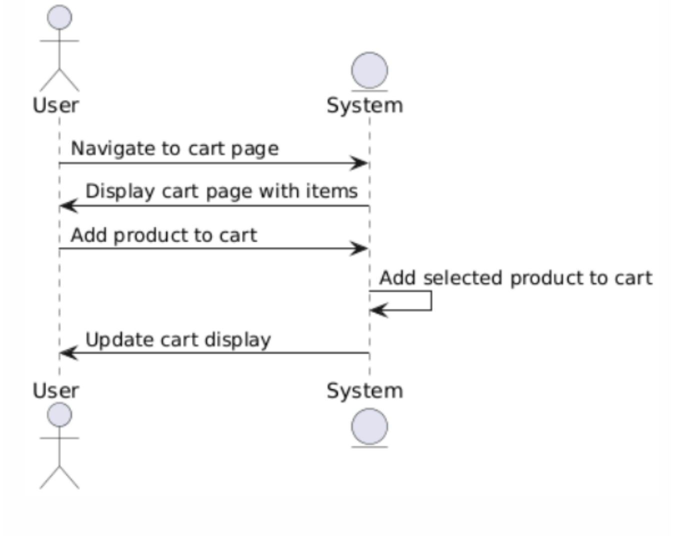
. Make Payment for Rentals


User
System
PaymentGateway
Navigate to checkout page
Display rental items and total cost
Confirm payment details
Validate payment details
Send payment info (credit card)
Process payment (success/failure)
Update rental status in DB
Send payment confirmation
alt
[Invalid Payment Info]
Display error message (Invalid payment)
alt
[Payment Failure]
Display payment failure message
User
System
PaymentGateway

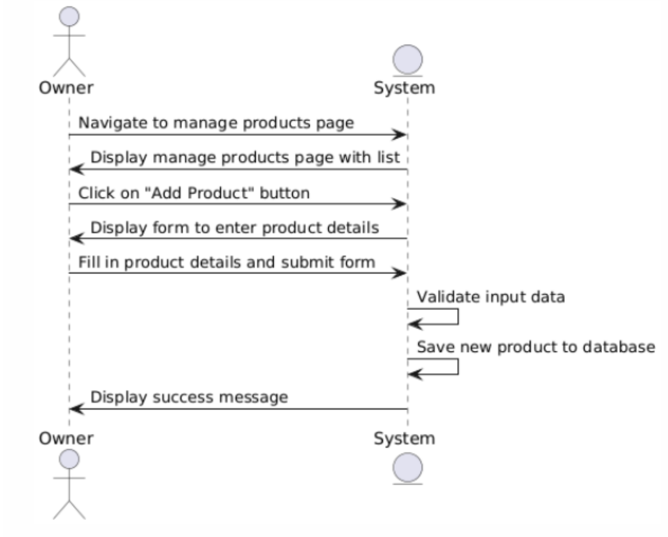
Leave Feedback for Products


User
System
Navigate to feedback page
Display feedback form
Select product for feedback
Retrieve product details
Fill in feedback form with rating and comment
Validate feedback input
Display validation errors (if any)
Submit feedback form
Call FeedbackService.saveFeedback()
Display success message (Feedback submitted)
User
System

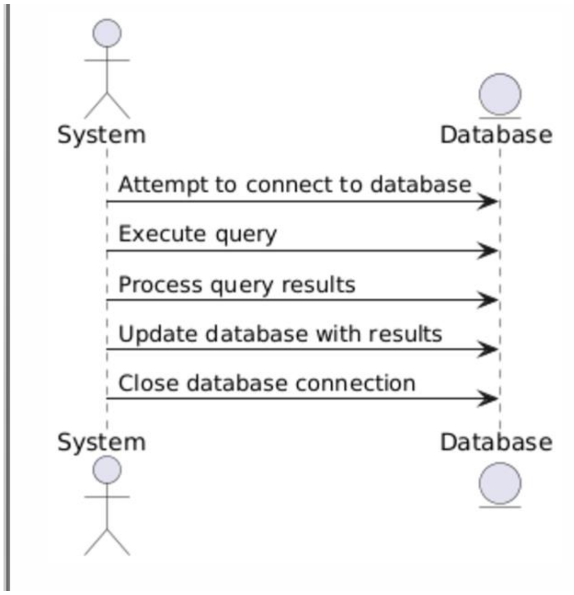
. Cart Management



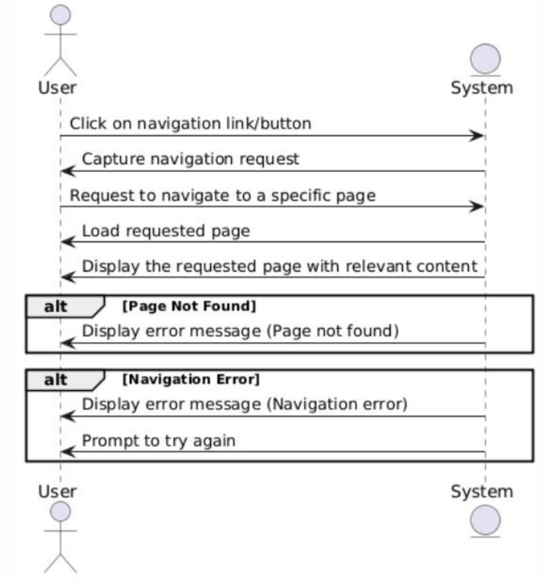
Manage Products


Owner
System
Navigate to manage products page
Display manage products page with list
Click on "Add Product" button
Display form to enter product details
Fill in product details and submit form
Validate input data
Save new product to database
Display success message
Owner
System

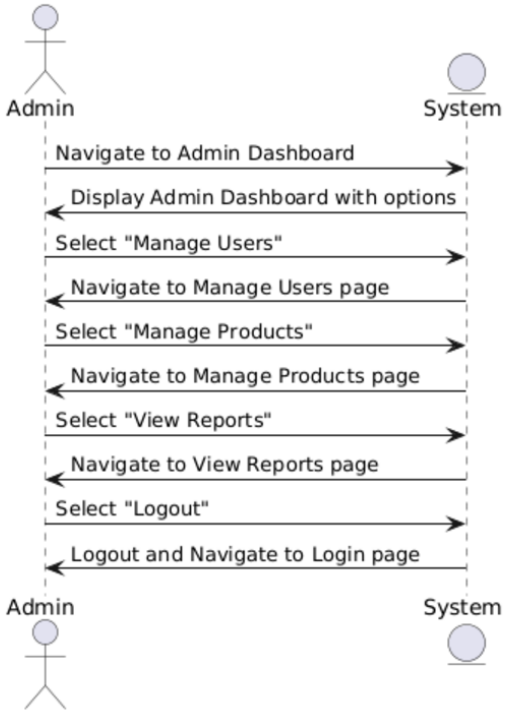
Database Management


System
Database
Attempt to connect to database
Execute query
Process query results
Update database with results
Close database connection
System
Database

UI Navigation


User
System
Click on navigation link/button
Capture navigation request
Request to navigate to a specific page
Load requested page
Display the requested page with relevant content
alt
[Page Not Found]
Display error message (Page not found)
alt
[Navigation Error]
Display error message (Navigation error)
Prompt to try again
User
System

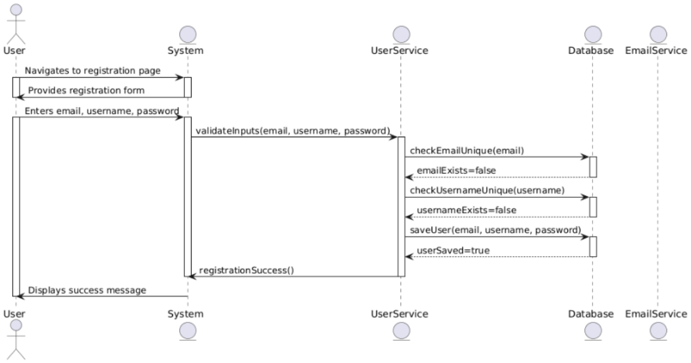
Admin management


Admin
System
Navigate to Admin Dashboard
Display Admin Dashboard with options
Select "Manage Users"
Navigate to Manage Users page
Select "Manage Products"
Navigate to Manage Products page
Select "View Reports"
Navigate to View Reports page
Select "Logout"
Logout and Navigate to Login page
Admin
System

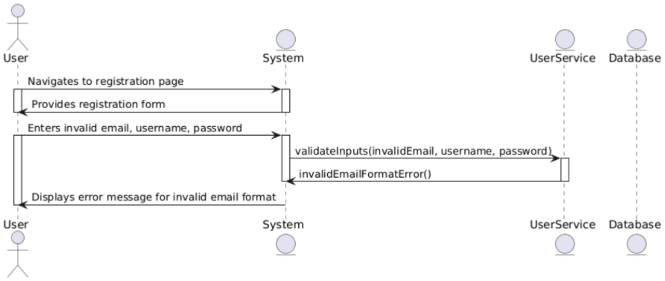
6. Sequence Diagram

Register a user

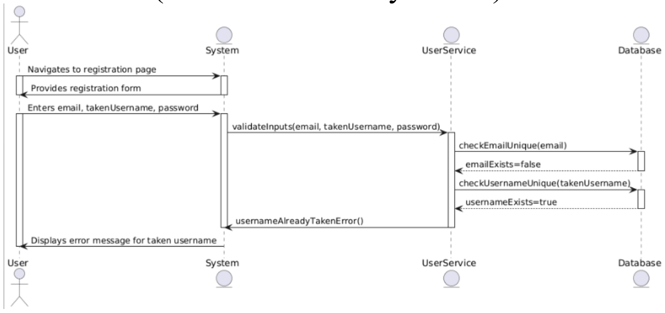
Main success scenario:



Error Flow (Invalid Email Format)

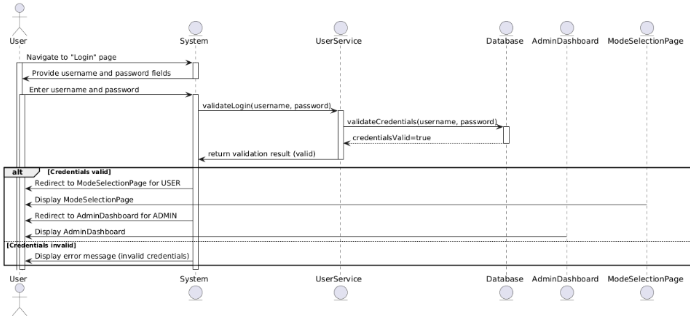

User
System
UserService
Database
Navigates to registration page
Provides registration form
Enters invalid email, username, password
validateInputs(invalidEmail, username, password)
invalidEmailFormatError()
Displays error message for invalid email format
User
System
UserService
Database

Error Flow (Username Already Taken)

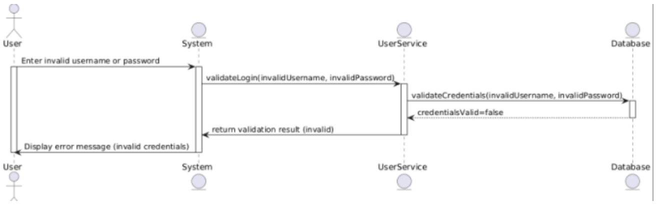

User
System
UserService
Database
Navigates to registration page
Provides registration form
Enters email, takenUsername, password
validateInputs(email, takenUsername, password)
checkEmailUnique(email)
emailExists=false
checkUsernameUnique(takenUsername)
usernameExists=true
usernameAlreadyTakenError()
Displays error message for taken username
User
System
UserService
Database

Login

Main success scenario

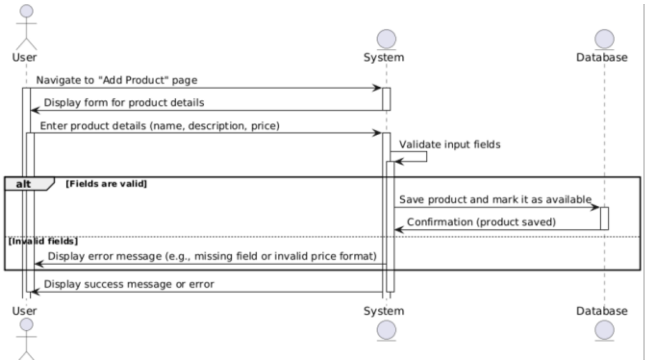

User
System
UserService
Database
AdminDashboard
ModeSelectionPage
Navigate to "Login" page
Provide username and password fields
Enter username and password
validateLogin(username, password)
validateCredentials(username, password)
credentialsValid=true
return validation result (valid)
alt
[Credentials valid]
Redirect to ModeSelectionPage for USER
Display ModeSelectionPage
Redirect to AdminDashboard for ADMIN
Display AdminDashboard
[Credentials invalid]
Display error message (invalid credentials)
User
System
UserService
Database
AdminDashboard
ModeSelectionPage

Invalid credentials

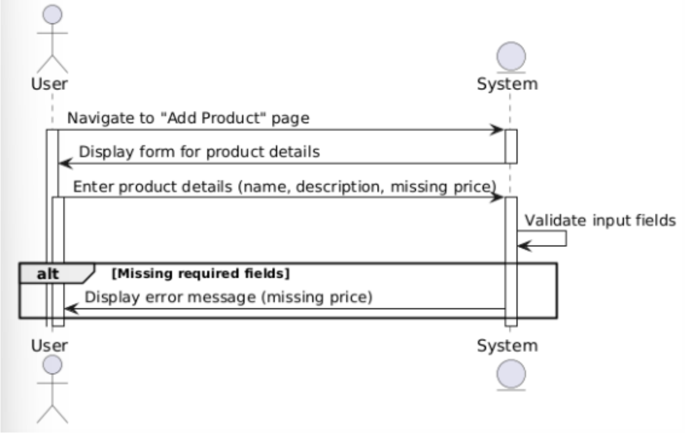

User
System
UserService
Database
Enter invalid username or password
validateLogin(invalidUsername, invalidPassword)
validateCredentials(invalidUsername, invalidPassword)
credentialsValid=false
return validation result (invalid)
Display error message (invalid credentials)
User
System
UserService
Database

List a product for rent

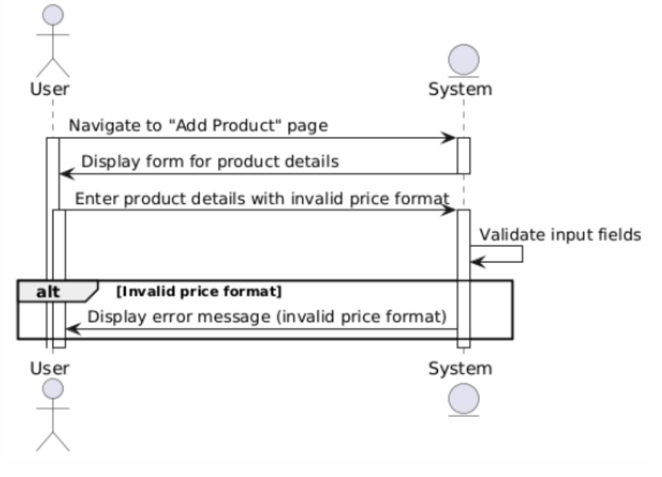
Main scenario


User
System
Database
Navigate to "Add Product" page
Display form for product details
Enter product details (name, description, price)
Validate input fields
alt
[Fields are valid]
Save product and mark it as available
Confirmation (product saved)
[Invalid fields]
Display error message (e.g., missing field or invalid price format)
Display success message or error
User
System
Database

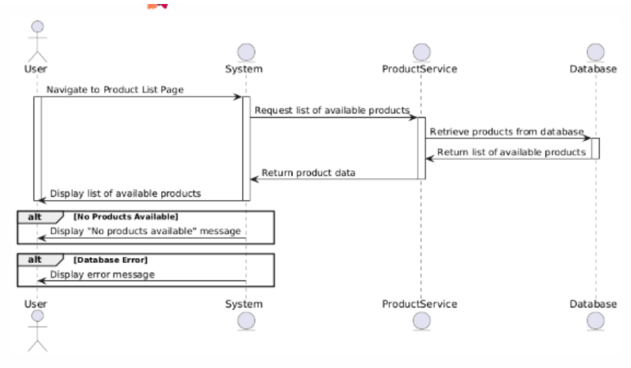
Missing required fields


User
System
Navigate to "Add Product" page
Display form for product details
Enter product details (name, description, missing price)
Validate input fields
alt
[Missing required fields]
Display error message (missing price)
User
System

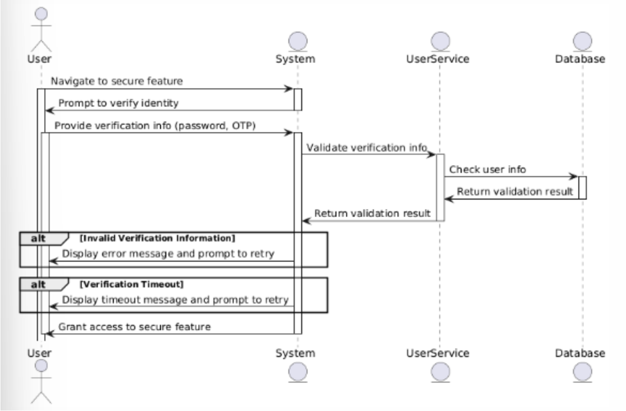
Invalid Price Format


User
System
Navigate to "Add Product" page
Display form for product details
Enter product details with invalid price format
Validate input fields
alt
[Invalid price format]
Display error message (invalid price format)
User
System

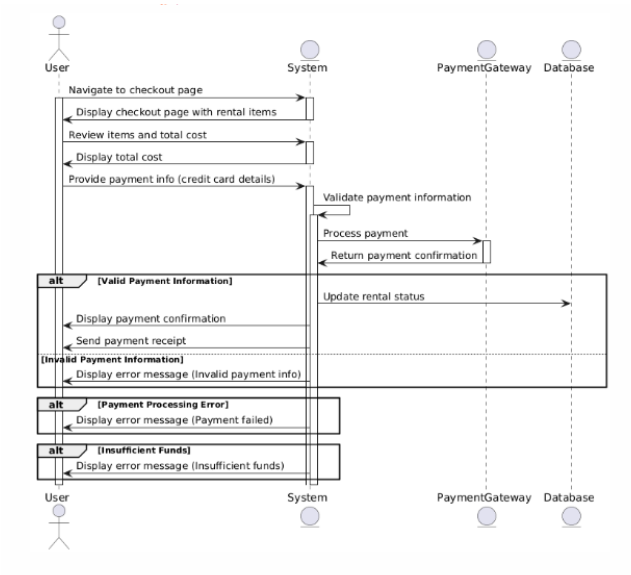
View product list


User
System
ProductService
Database
Navigate to Product List Page
Request list of available products
Retrieve products from database
Return list of available products
Return product data
Display list of available products
alt [No Products Available]
Display "No products available" message
alt [Database Error]
Display error message
User
System
ProductService
Database

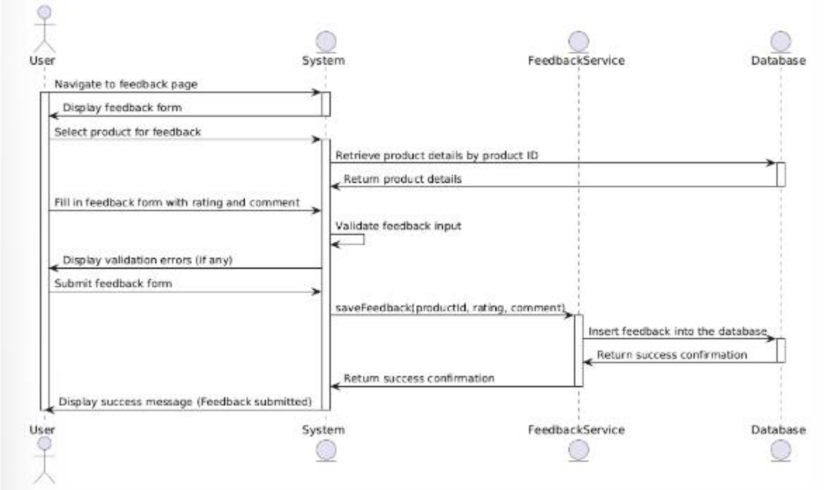
Verify user identity


User
System
UserService
Database
Navigate to secure feature
Prompt to verify identity
Provide verification info (password, OTP)
Validate verification info
Check user info
Return validation result
Return validation result
alt [Invalid Verification Information]
Display error message and prompt to retry
alt [Verification Timeout]
Display timeout message and prompt to retry
Grant access to secure feature
User
System
UserService
Database

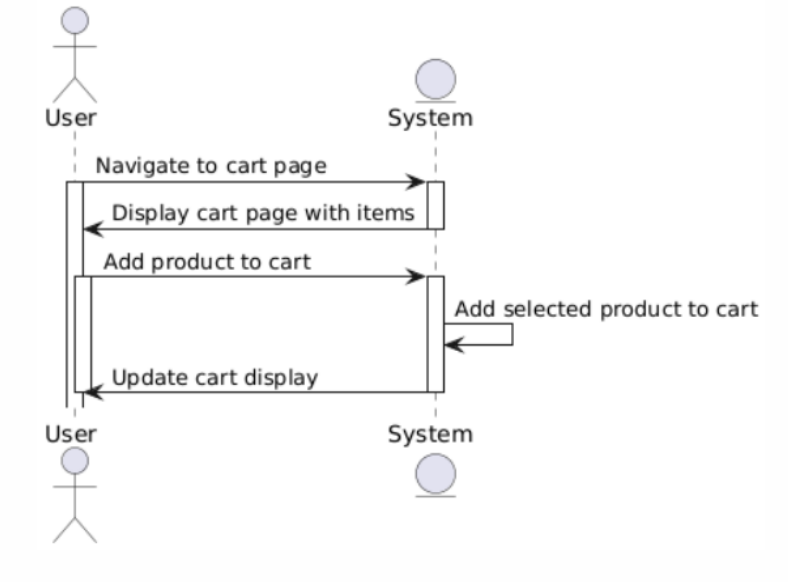
. Make Payment for Rentals


User
System
PaymentGateway
Database
Navigate to checkout page
Display checkout page with rental items
Review items and total cost
Display total cost
Provide payment info (credit card details)
Validate payment information
Process payment
Return payment confirmation
alt [Valid Payment Information]
Update rental status
Display payment confirmation
Send payment receipt
[Invalid Payment Information]
Display error message (Invalid payment info)
alt [Payment Processing Error]
Display error message (Payment failed)
alt [Insufficient Funds]
Display error message (Insufficient funds)
User
System
PaymentGateway
Database

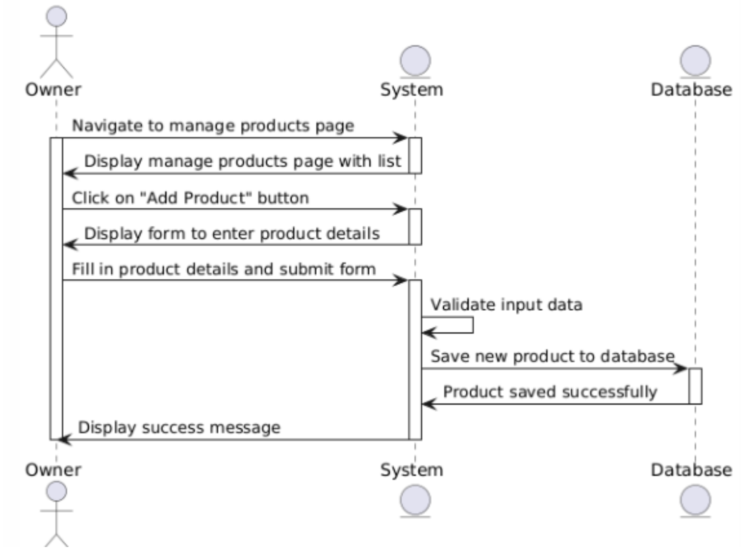
Leave Feedback for Products


User
System
FeedbackService
Database
Navigate to feedback page
Display feedback form
Select product for feedback
Retrieve product details by product ID
Return product details
Fill in feedback form with rating and comment
Validate feedback input
Display validation errors (if any)
Submit feedback form
saveFeedback(productId, rating, comment)
Insert feedback into the database
Return success confirmation
Return success confirmation
Display success message (Feedback submitted)
User
System
FeedbackService
Database

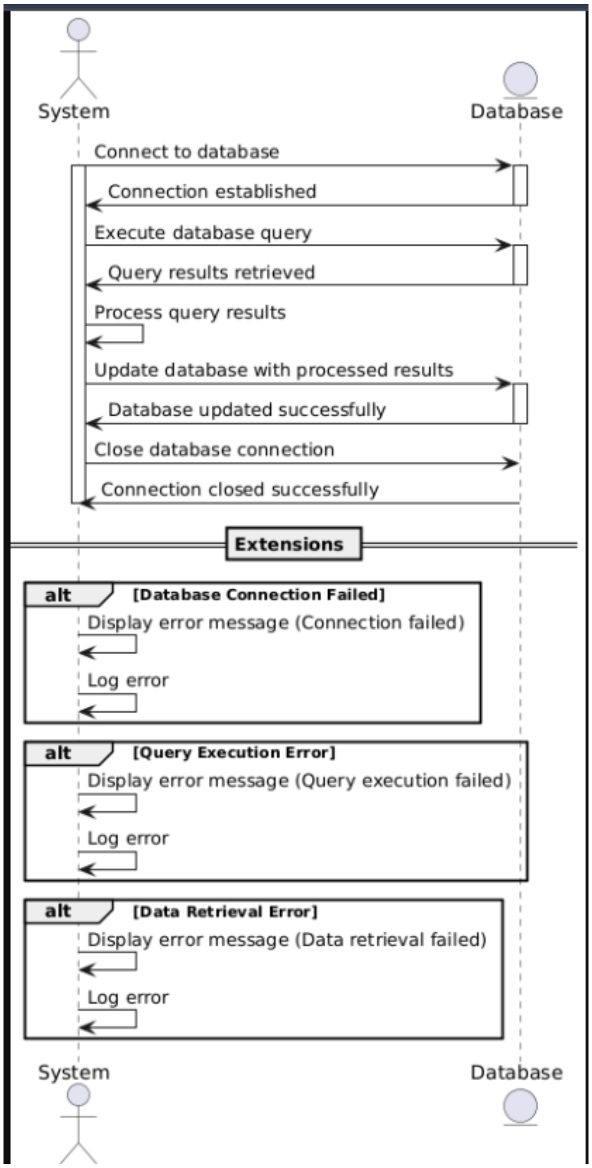
Cart Management



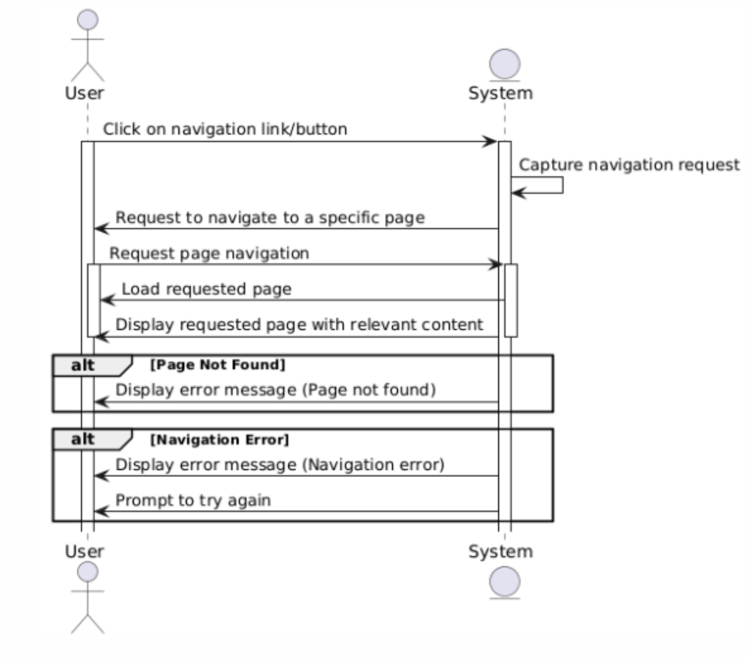
Manage product


Owner
System
Database
Navigate to manage products page
Display manage products page with list
Click on "Add Product" button
Display form to enter product details
Fill in product details and submit form
Validate input data
Save new product to database
Product saved successfully
Display success message
Owner
System
Database

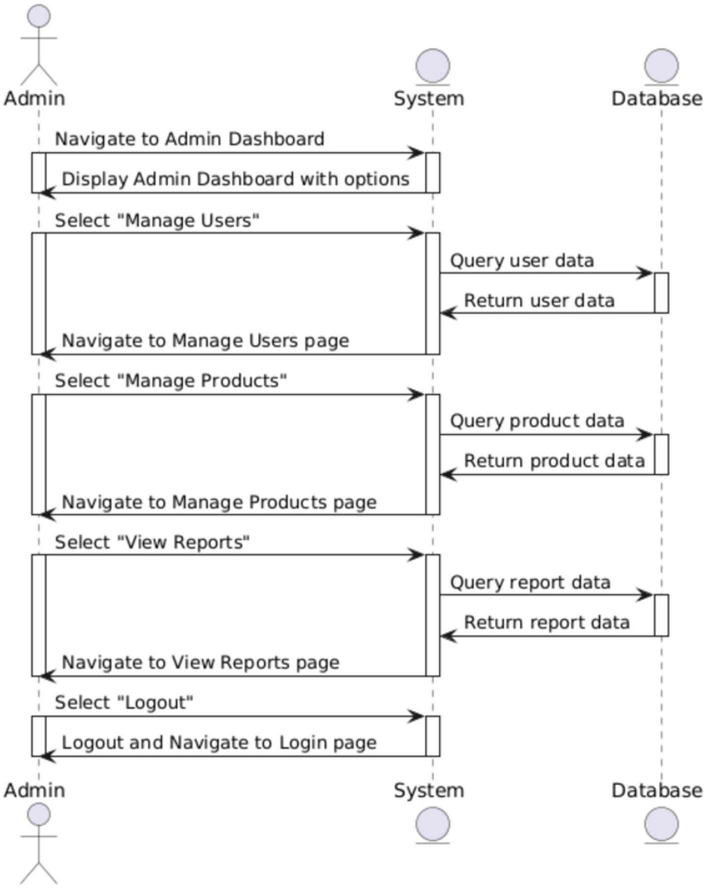
Database Management


System
Database
Connect to database
Connection established
Execute database query
Query results retrieved
Process query results
Update database with processed results
Database updated successfully
Close database connection
Connection closed successfully
Extensions
alt [Database Connection Failed]
Display error message (Connection failed)
Log error
alt [Query Execution Error]
Display error message (Query execution failed)
Log error
alt [Data Retrieval Error]
Display error message (Data retrieval failed)
Log error
System
Database

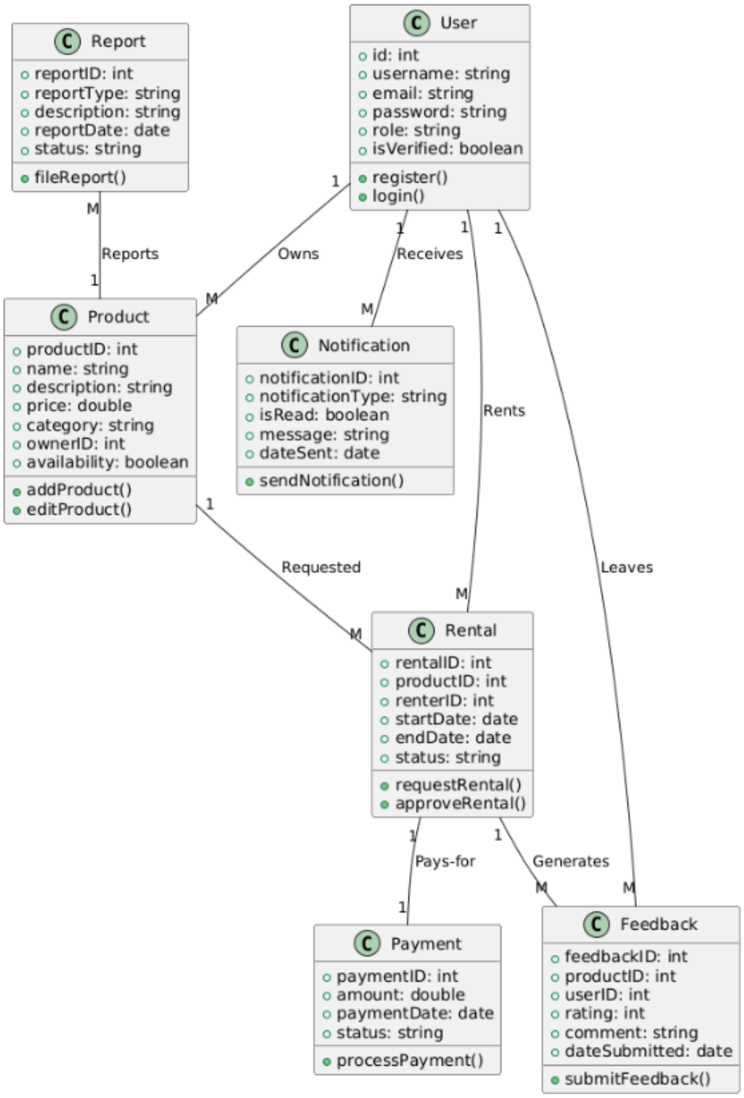
UI Navigation


User
System
Click on navigation link/button
Capture navigation request
Request to navigate to a specific page
Request page navigation
Load requested page
Display requested page with relevant content
alt
[Page Not Found]
Display error message (Page not found)
alt
[Navigation Error]
Display error message (Navigation error)
Prompt to try again
User
System

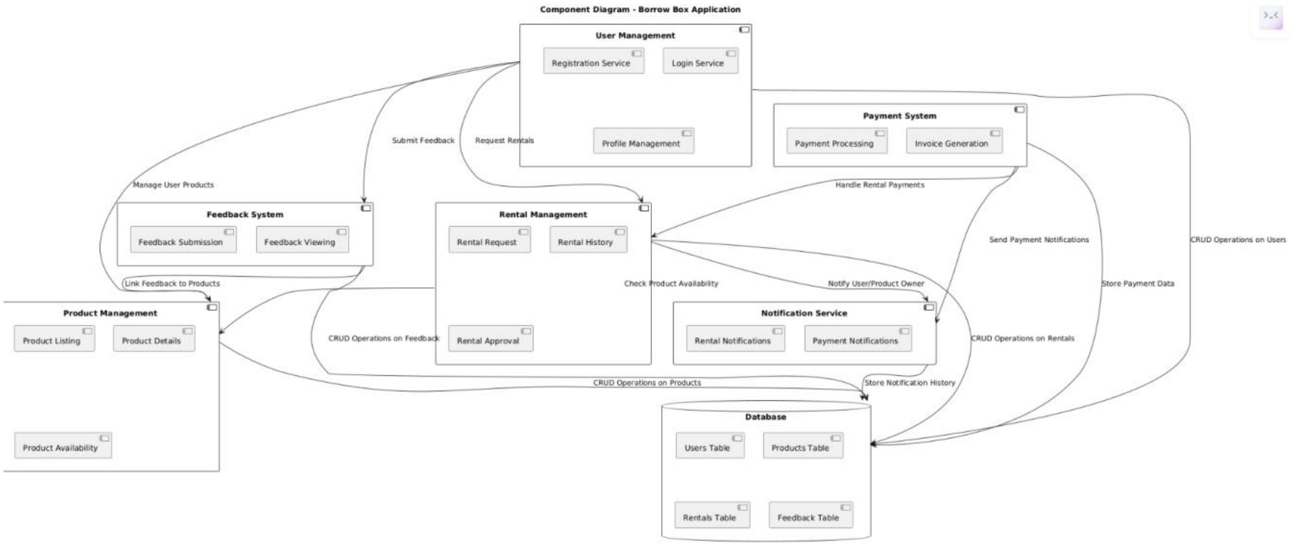
Admin Management



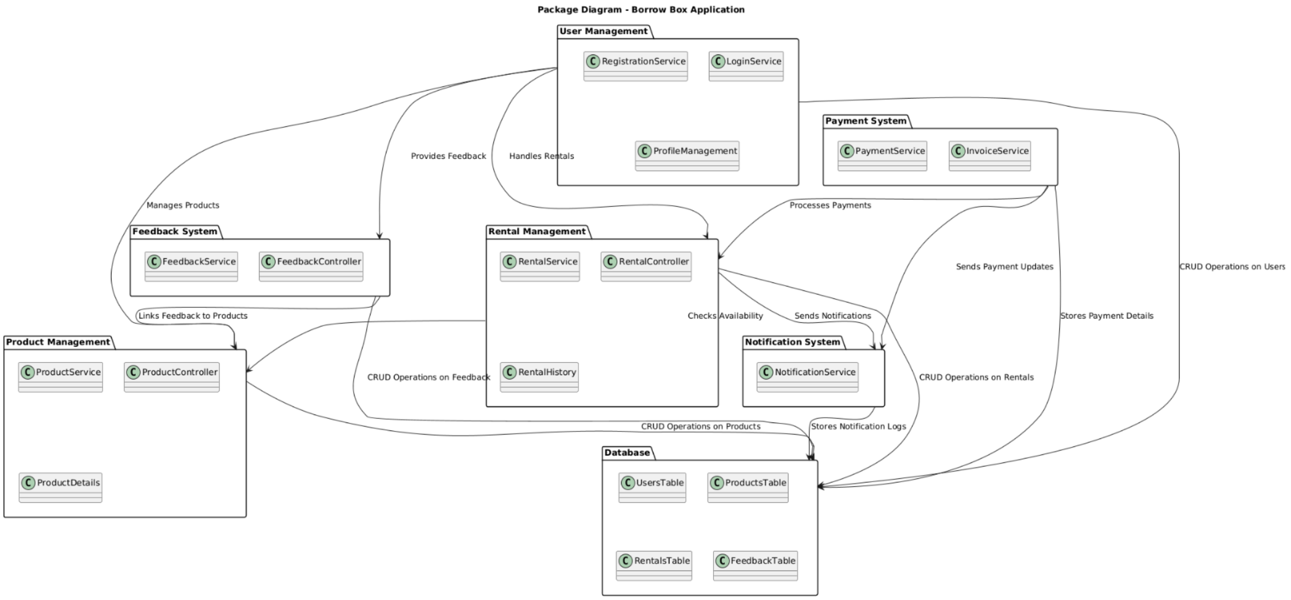
7. Class Diagram



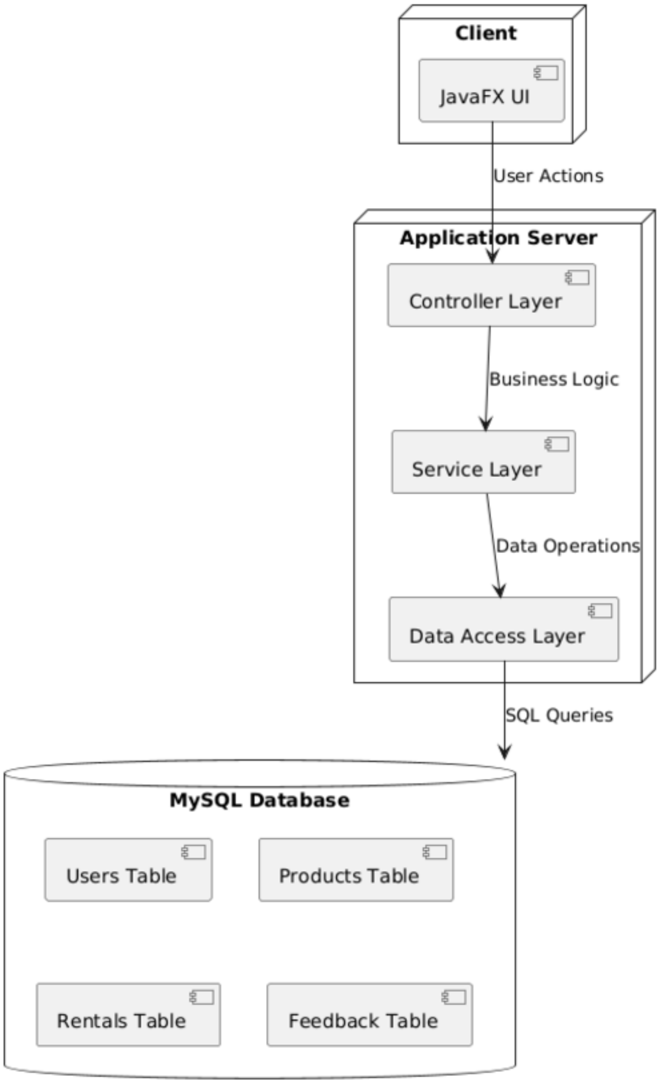
8. Component Diagram



9. Package Diagram



10. Deployment Diagram


Client
JavaFX UI
User Actions
Application Server
Controller Layer
Business Logic
Service Layer
Data Operations
Data Access Layer
SQL Queries
MySQL Database
Users Table
Products Table
Rentals Table
Feedback Table