VIT-AP UNIVERSITY

SOFTWARE ENGINEERING

LAB - L55 + L56

CSE 1005

HOUSE-HUB

BATCH-13

Submitted by

Bharani Kumar	22BCE7554
MD Afnan	22BCE9906
Chakradhar	22BCE9849
Atharva	22BCE9298
Yashas	22BCE8387

Faculty

Prof. Shaik Kareemulla

Problem Statement

In today's fast-paced real estate market, both buyers and renters struggle to find suitable housing options that meet their specific needs, while sellers face challenges in efficiently reaching potential customers. The current online platforms often lack user-friendly interfaces, detailed property listings, and secure, streamlined processes for transactions. Additionally, the absence of tailored authentication systems for buyers and sellers leads to complications in managing accounts and property listings.

House-Hub aims to bridge this gap by creating a comprehensive web platform that provides a seamless interface between customers and sellers. It offers detailed listings of houses available for sale or rent, allowing customers to browse, compare, and complete transactions directly on the site. For sellers, **House-Hub** provides an easy-to-use account management system, enabling them to list their properties with all necessary details, ensuring they reach the right audience. The platform also incorporates distinct authentication mechanisms for users and sellers, ensuring that each group has access to the appropriate tools and features, with all login details securely stored in a database.

House-Hub addresses the need for a unified, secure, and efficient real estate platform that simplifies the buying, renting, and selling processes, ultimately enhancing the experience for all parties.

HOUSE-HUB

(WEB DEVELOPMENT)

SRS DOCUMENT

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

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 Assumptions and Dependencies

3. Specific Requirements

3.1 Functional Requirements

3.2 Non-Functional Requirements

4. Appendices

1. Introduction

1.1 Purpose

House-Hub simplifies real estate by connecting buyers, renters, and sellers on a single platform. It enables easy property browsing, secure transactions, and efficient property management, enhancing the overall experience for all users.

1.2 Scope

House-Hub will offer a platform where users can browse and secure properties, and sellers can list and manage their offerings. It will include separate authentication for users and sellers, along with secure transaction processing. The platform will be mobile-responsive and scalable for future growth.

1.3 Definitions, Acronyms, and Abbreviations

User: An individual interacting with the House-Hub platform, either as a buyer/renter or a seller.
UI: User Interface, the visual elements of the House-Hub platform that users interact with.
DB: Database, the system used to store and manage user, seller, and property information.
Authentication: The process of verifying a user's identity to grant access to a system or application.

1.4 References

- Zillow Real Estate Data on Kaggle: Link
- U.S. Real Estate Data from Redfin: Link

1.5 Overview

This SRS document details the overall description of the system, specific functional and non-functional requirements, and other essential components for the successful development and deployment of the House-Hub system.

2. Overall Description

2.1 Product Perspective

The platform simplifies the real estate process by offering a unified interface for both buyers/renters and sellers. It enables users to browse, compare, and secure properties, while sellers can efficiently list and manage their offerings. With secure transactions and intuitive navigation, it enhances the overall real estate experience for all parties involved.

2.2 Product Functions

- **Property Listings:** Sellers can create and manage property details.
- **User Browsing:** Buyers/renters can search and view property listings.
- **Transaction Processing:** Securely handles property purchases and rentals.
- **Account Management:** Separate authentication for users and sellers.
- **Communication Tools**: Enables direct messaging between users and sellers.

2.3 User Classes and Characteristics

- **Buyers/Renters:** Search for properties, requiring easy browsing and secure transactions.
- **Sellers**: List and manage properties, needing tools for listings and transaction tracking.
- **Administrators:** Manage the platform, monitor activity, and ensure security and functionality.

2.4 Operating Environment

- **Hardware:** Servers for hosting, client devices like desktops and smartphones.
- **Software:** Database (MySQL), and web technologies (HTML, CSS, JavaScript, NODE.JS).
- **Platforms:** Node.js for back-end, React for front-end, AWS, Azure.

2.5 Design and Implementation Constraints

- **Performance:** Ensure the platform loads quickly and handles multiple users efficiently.
- **Security:** Implement strong measures to protect user data and transactions.
- **Compatibility:** Design for compatibility across various devices, OS and browsers
- **Scalability:** Build a scalable architecture to accommodate growth in users and data.

2.6 Assumptions and Dependencies

- **Internet Access:** Reliable internet for platform use.
- **Browser Compatibility:** Modern, updated browsers for full functionality.
- **Cloud Services:** Dependence on cloud services for hosting and scalability.
- **Third-Party Integrations:** Reliable payment and communication services.

3. Specific Requirements

3.1 Functional Requirements

- **FR1:** The system shall allow users to browse and filter property listings based on criteria such as location, price, and features.
- **FR2:** The system shall enable sellers to create, edit, and manage property listings, including adding descriptions, photos, and pricing information.
- **FR3:** The system shall provide secure transaction processing for purchasing or renting properties, ensuring a completion rate of at least 95%.
- **FR4:** The system shall support real-time updates for property availability and transaction status, with updates reflected within 1 minute.
- **FR5:** The system shall log all user actions, property listings, and transactions for auditing and reporting purposes.

- **FR6:** The system shall offer an API for integrating with third party services such as payment gateways and communication tools, to enhance functionality.

1.2 Non-Functional Requirements

- **NFR1:** The system shall ensure data privacy and comply with relevant data protection regulations GDPR, CCPA
- **NFR2:** The system shall be scalable to handle a large number of concurrent users and property listings.
- **NFR3:** The system shall be available 99.9% of the time to ensure reliability and uptime.
- **NFR4:** The system shall have a mean time to recovery (MTTR) of less than 2 hours in case of outages or disruptions.
- **NFR5:** The system shall provide detailed logging and monitoring capabilities to track user activity, system performance, and errors.

2. Appendices

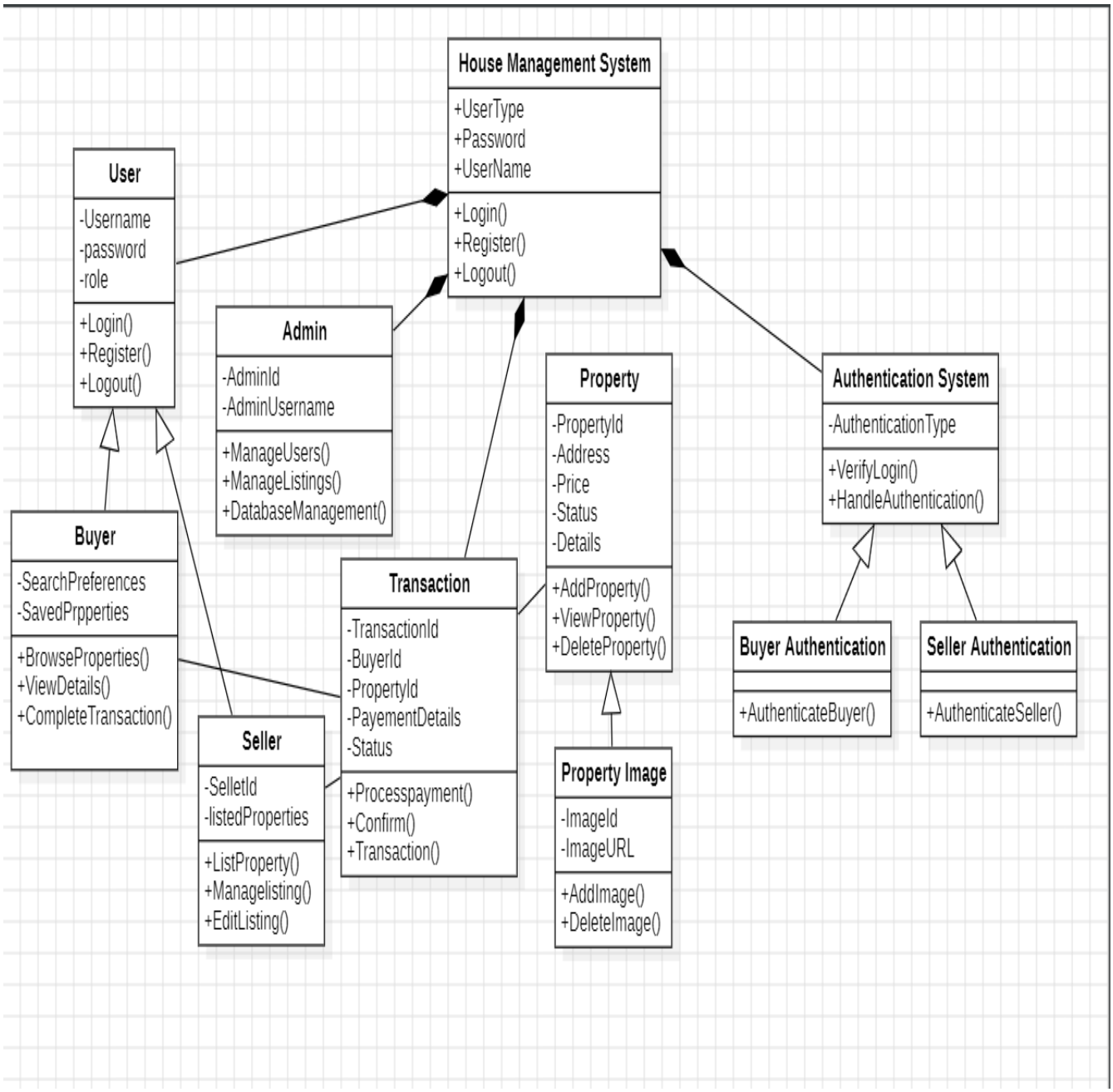
2.1 Glossary

- **Fraud Detection:** Identifying and preventing fraudulent activities.
- **Anomaly Detection:** Finding unusual patterns that may indicate fraud.
- **Data Integrity:** Ensuring data accuracy and consistency.
- **Transaction Monitoring:** Observing transactions to detect and prevent fraud.

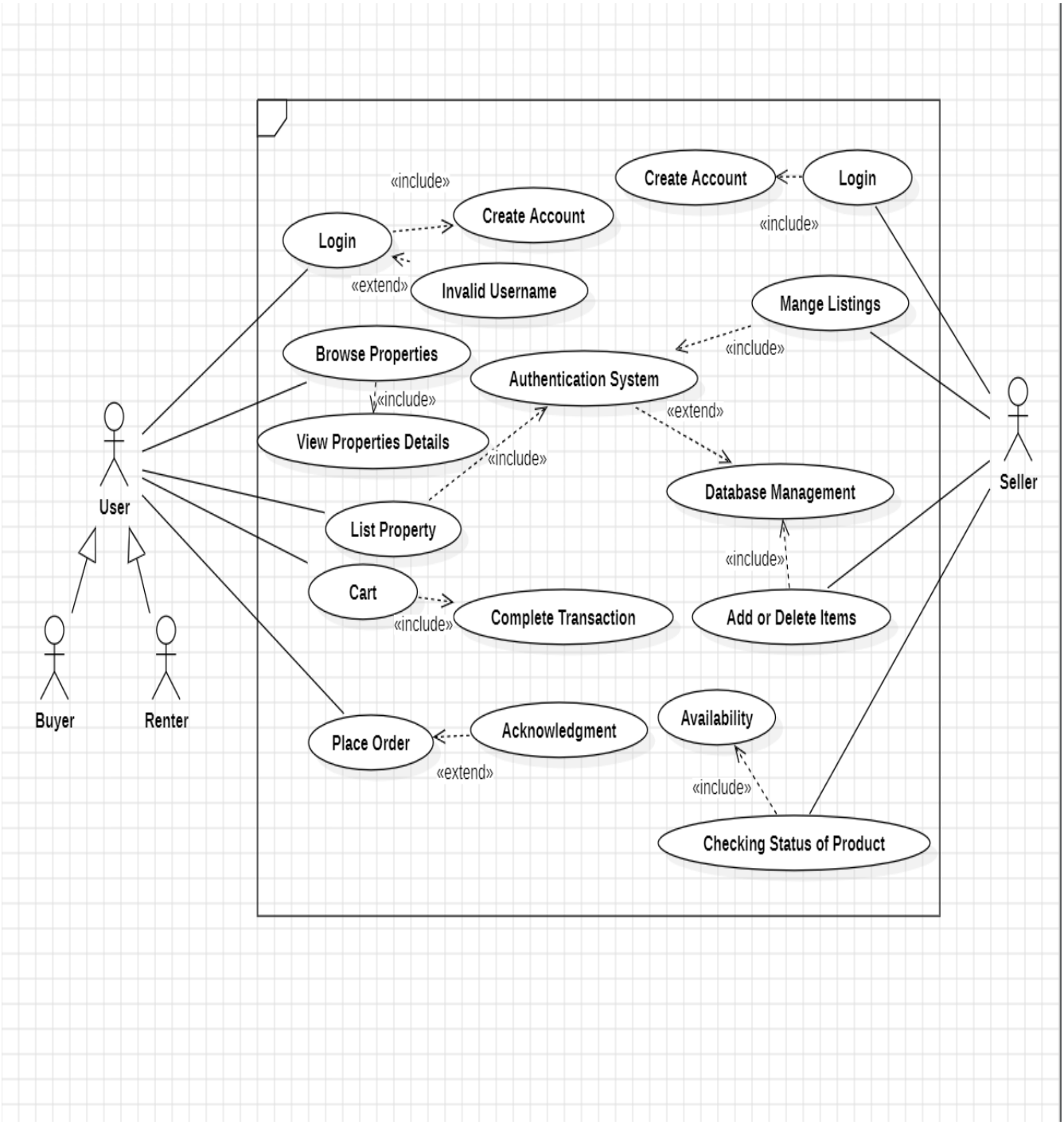
2.2 References

- Mozilla Developer Network Web Docs: https://developer.mozilla.org/
- Stack Overflow: https://stackoverflow.com/
- W3Schools: https://www.w3schools.com/

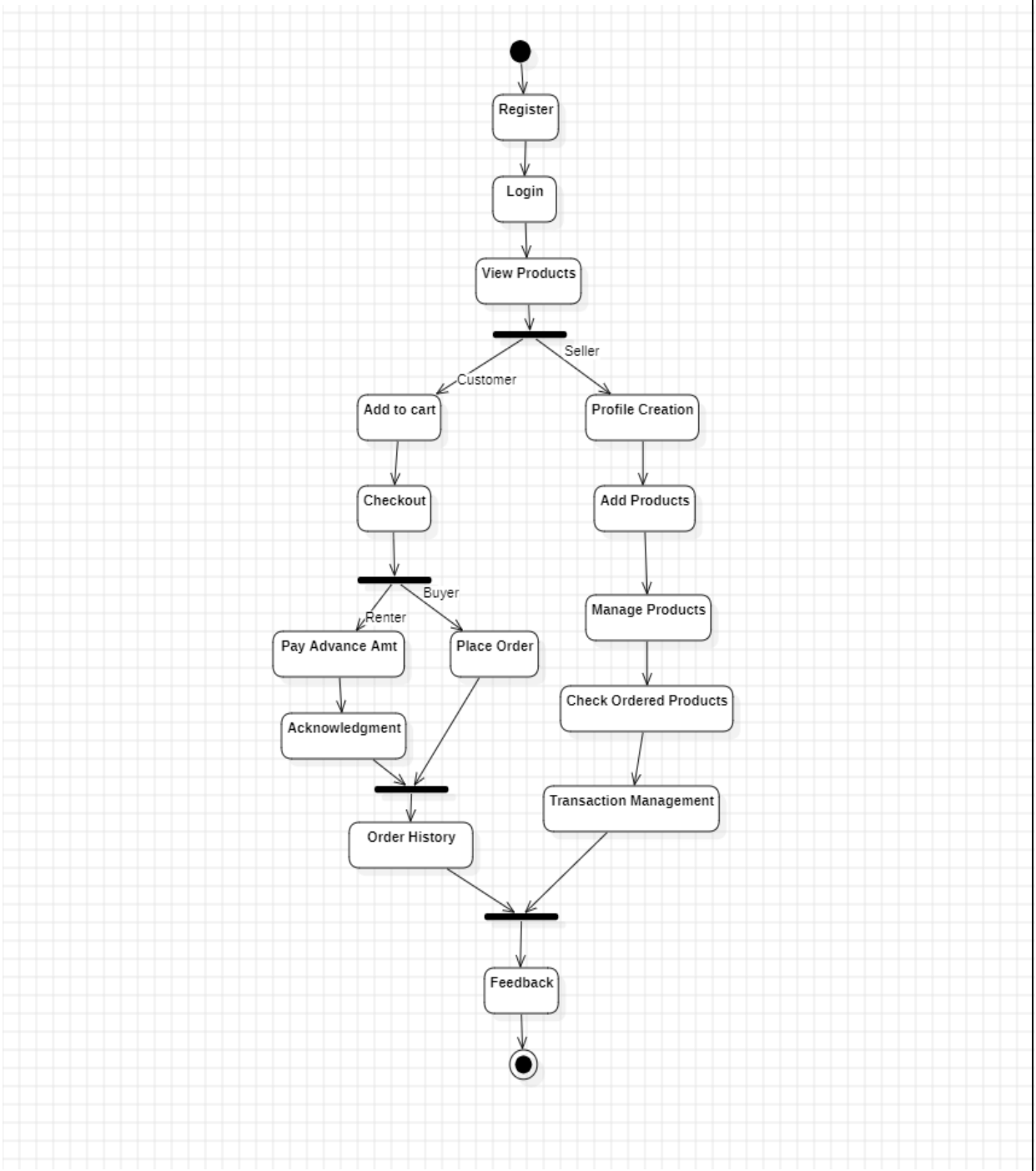
CLASS DIAGRAM



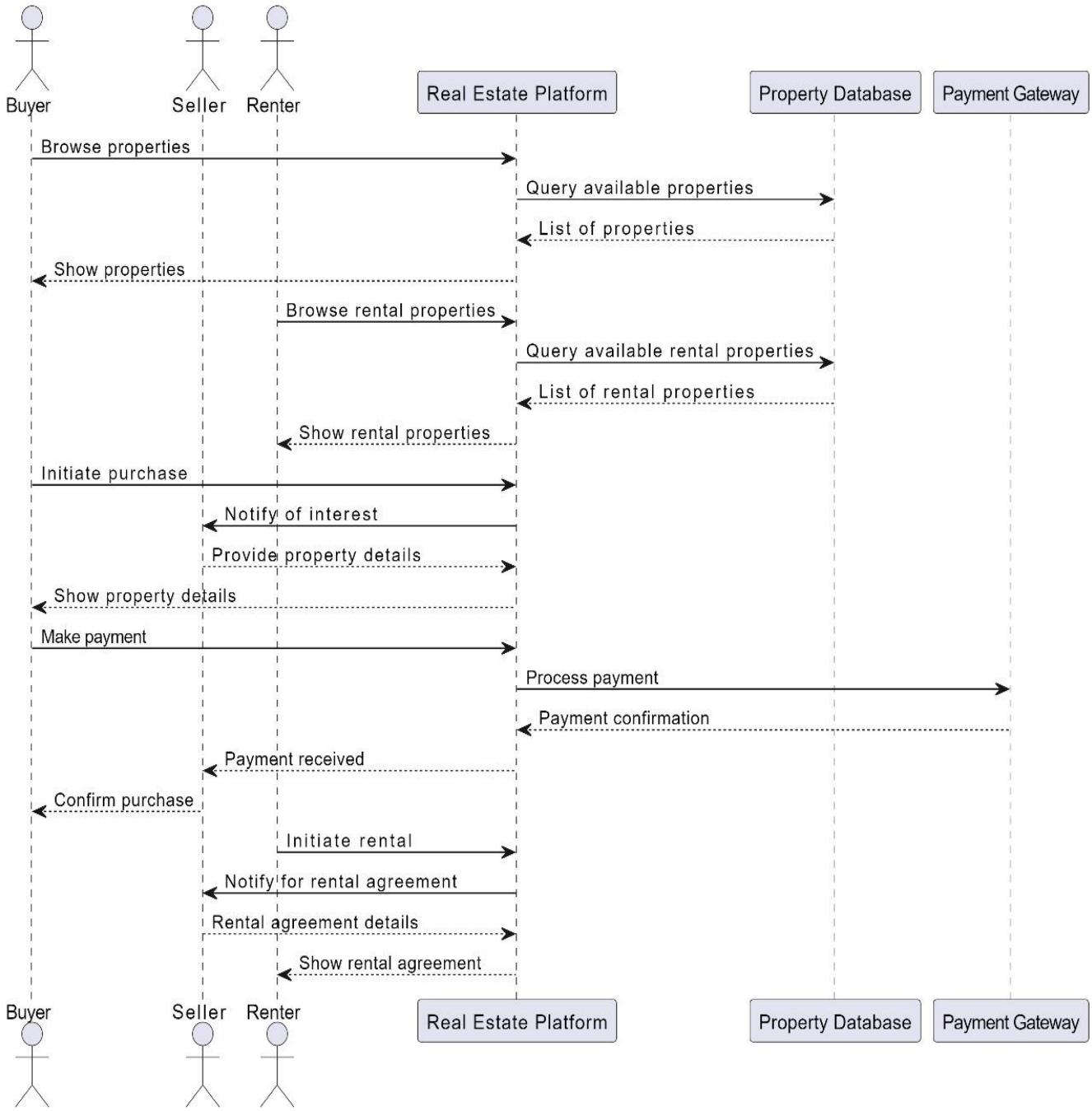
USE CASE DIAGRAM



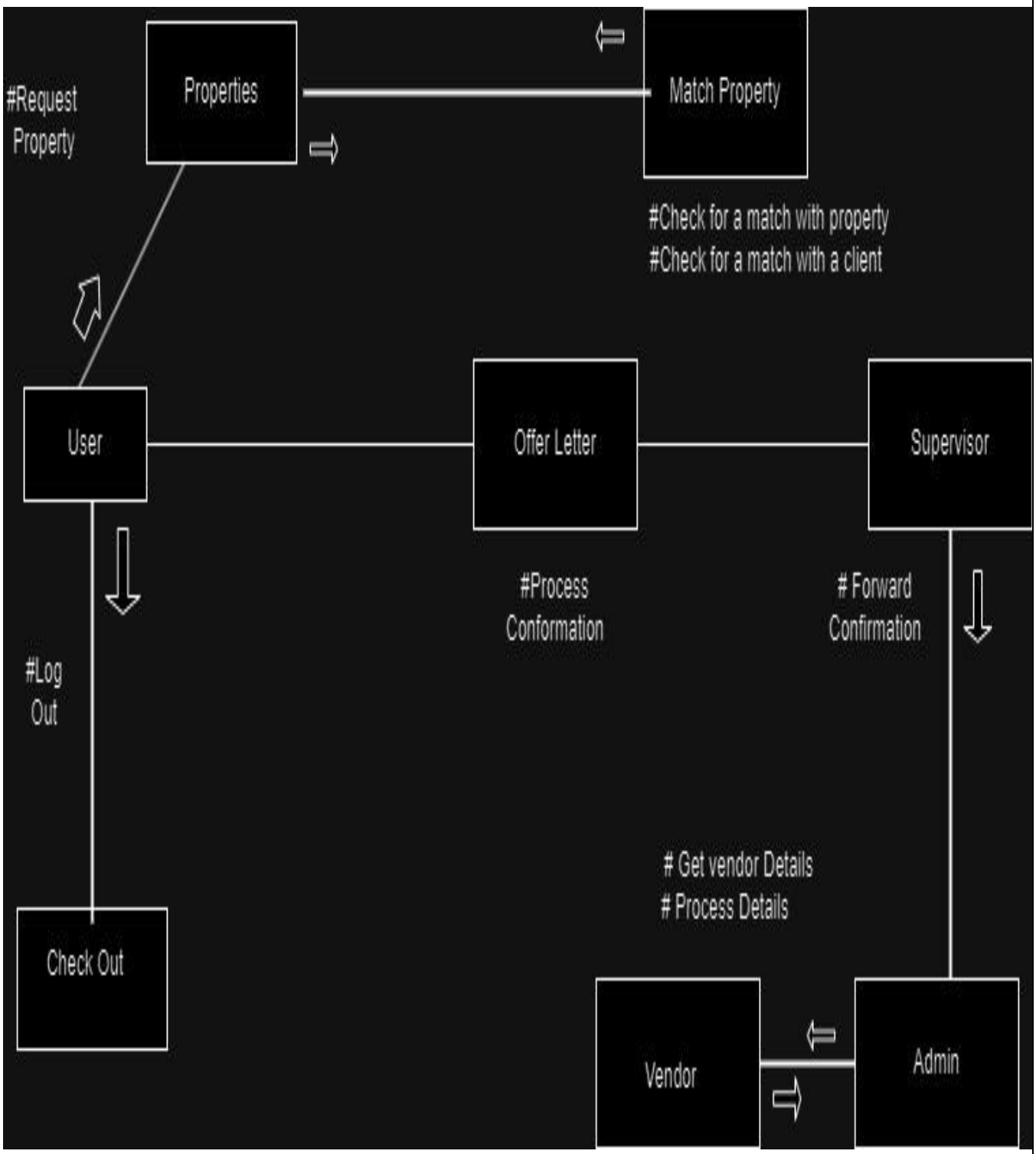
STATE CHART DIAGRAM



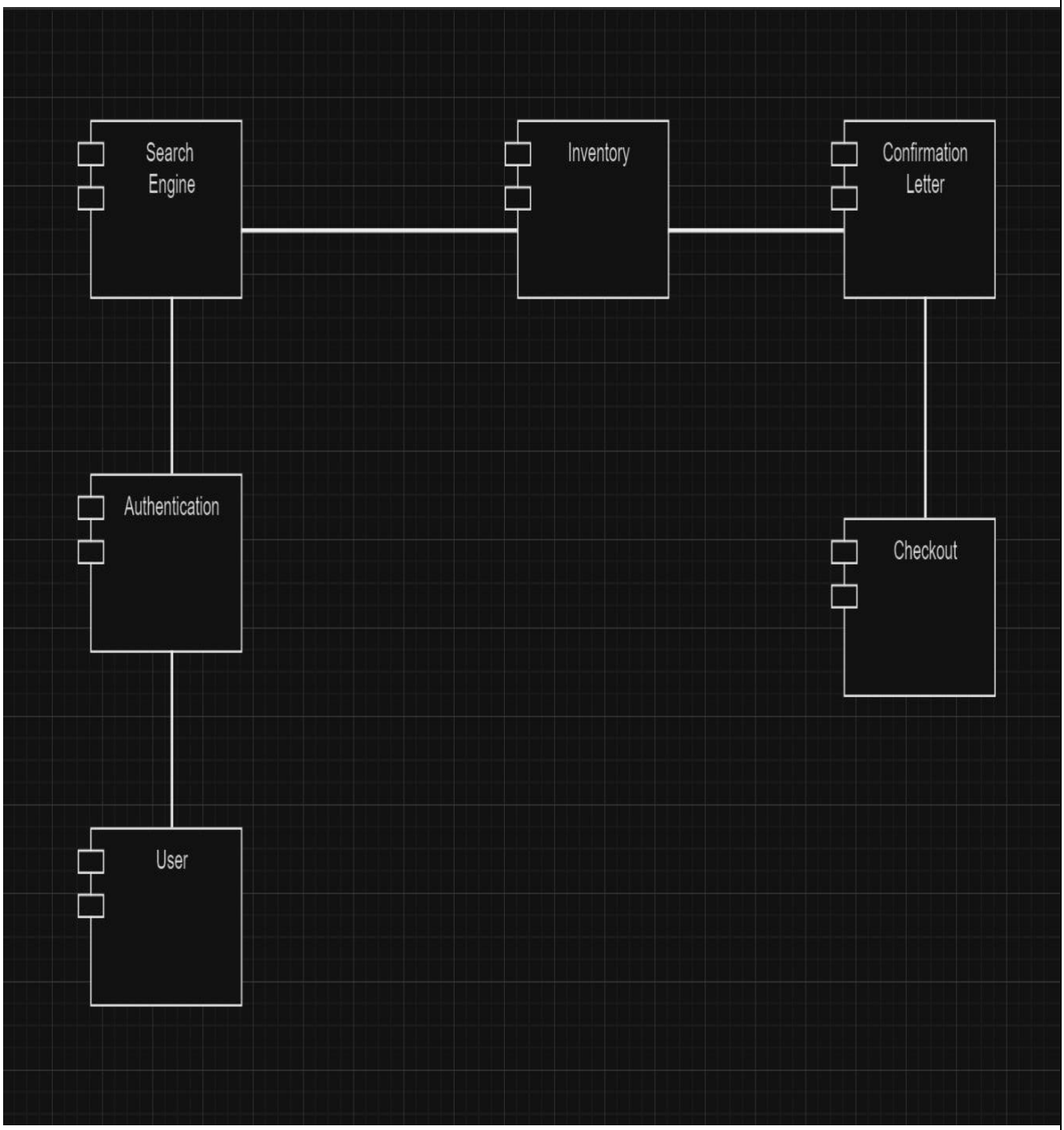
SEQUENCE DIAGRAM



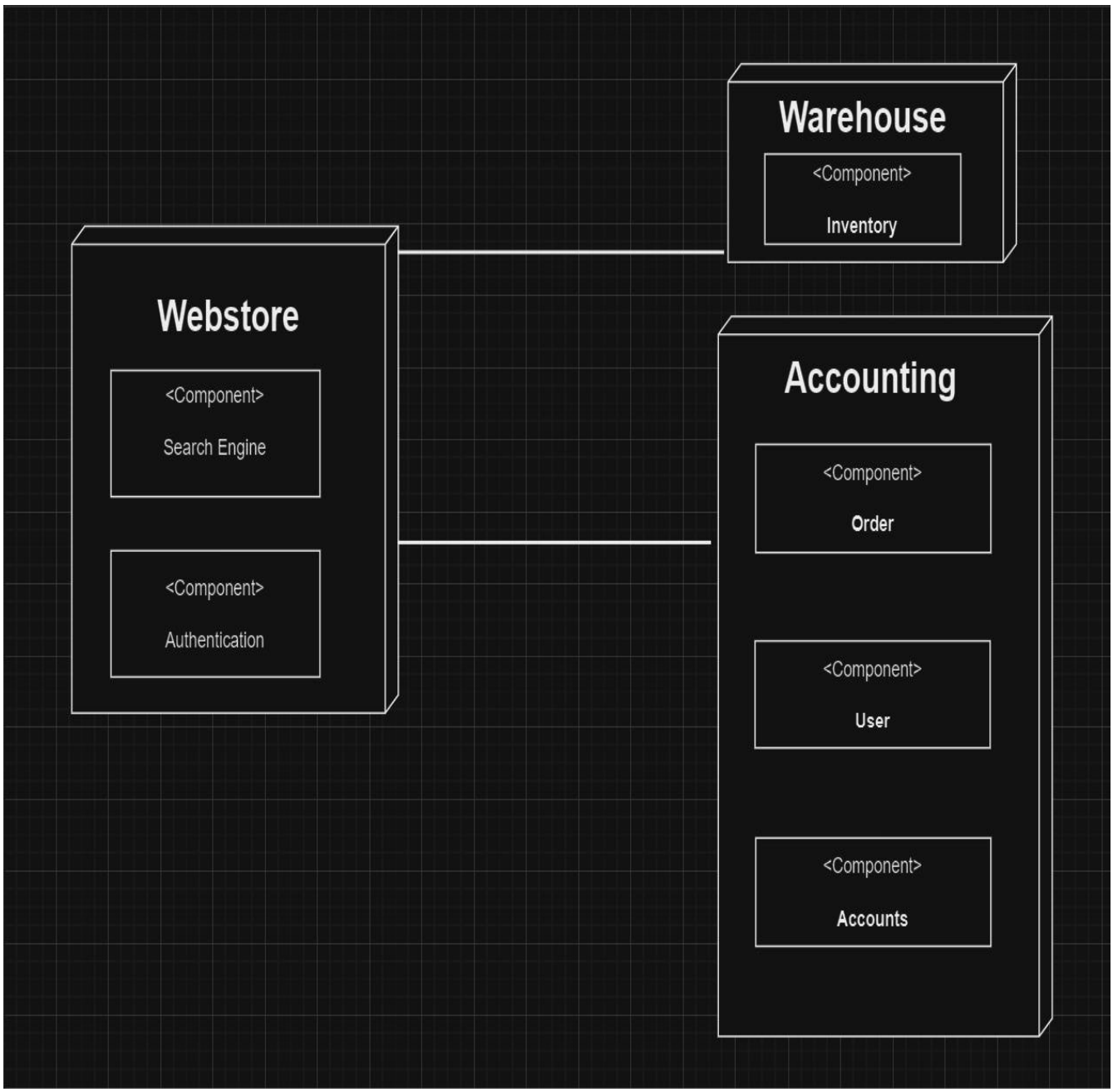
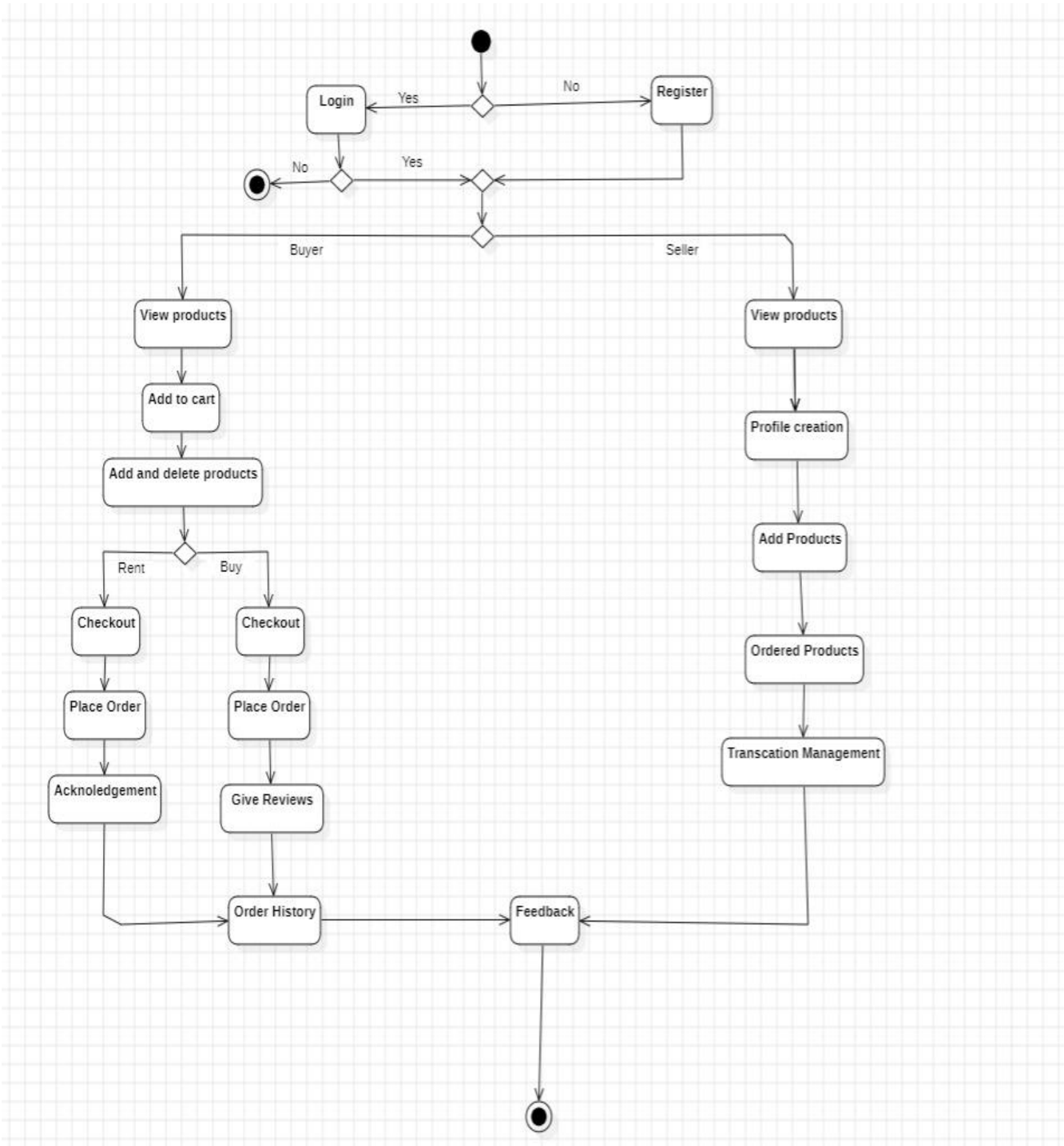
COLLABORATION DIAGRAM



COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM



ACTIVITY DIAGRAM