4.1 Elaboration Phase

The **Elaboration Phase** is the second phase in the Rational Unified Process (RUP) software development methodology. It is focused on defining the detailed architecture of the system and ensuring that the requirements, risks and high-level vision are fully understood and properly addressed. During the Elaboration Phase, the development team will create detailed models and diagrams to represent the architecture of the system, such as use case diagrams, class diagrams, object diagrams, collaboration or communication diagrams, sequence diagrams, activity diagrams, state chart diagrams, component diagrams, and deployment diagrams. These diagrams help to describe the relationships and interactions between objects, classes, and components, as well as the flow of data and processes within the system.

The use case diagram represents the system's functions and services from the user's point of view. The class diagram represents the objects and classes within the system and their relationships. The object diagram represents the instances of the classes at a specific moment in time. The collaboration or communication diagram represents the interactions between objects and classes. The sequence diagram represents the order of messages between objects over time. The activity diagram represents the flow of activities within the system. The state chart diagram represents the possible states of the system and how they transition between states. The component diagram represents the physical components and the relationships between them. The deployment diagram represents the hardware and software components on which the system will run.

By creating these detailed models and diagrams, the development team can identify potential problems, design solutions, and develop a detailed plan for implementing the system. This phase is crucial for ensuring that the system is designed correctly and meets the needs of the stakeholders.

The following are some **steps** that we should do in the Elaboration phase of the RUP methodology for our e-shop application:

1. **Use Case Diagram (v.2):** This diagram is used to represent the functional **requirements** of the system and the **interactions** between the system and its actors. It helps to identify the main functions of the system and the relationships between the actors and the system. In this **updated version**, we noticed that it was important to add **shop** and **shop category** classes and objects.

2. **Class Diagram (v.2):** This diagram is used to represent the **structure** of the system, including classes, their attributes, and their relationships. It provides a clear understanding of the objects and the relationships between them in the system. In this **updated version**, we noticed that it was important to add **shop** and **shop category** classes and objects.

3. **Object Diagram (v.1):** This diagram is used to represent the **instances** of classes and the relationships between them at a specific point in time. It helps to understand the behavior of the system at runtime.

4. **Collaboration or Communication Diagram (v.1):** This diagram is used to represent the flow of **messages** between objects or components in a system. It helps to understand how the objects interact and communicate with each other.

5. **Sequence Diagram (v.1):** This diagram is used to represent the **interaction between objects or components over time**, including the order of messages and the conditions under which they are sent. It helps to understand the flow of events in the system and the dependencies between the objects.

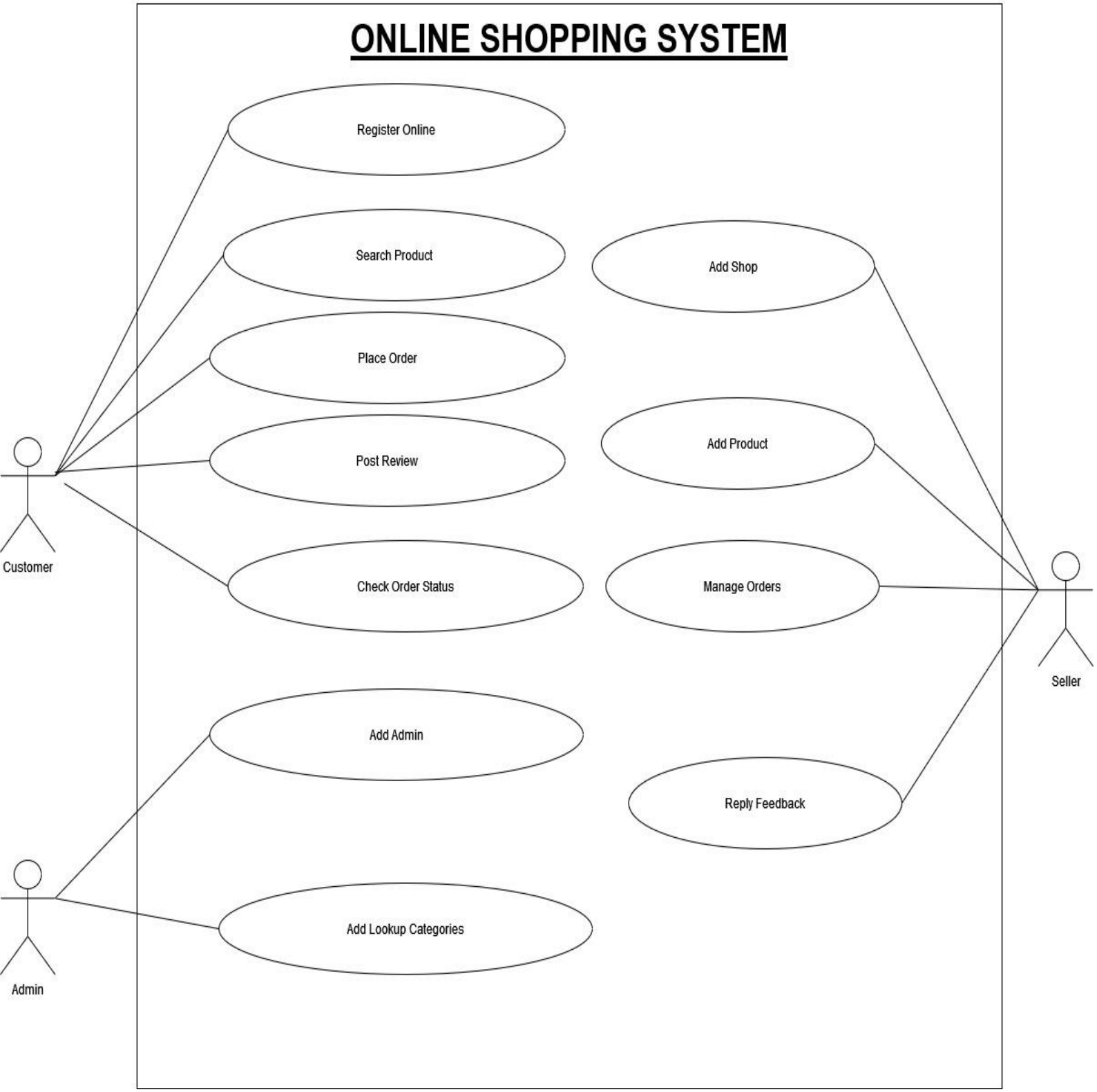
6. **Activity Diagram (v.1):** This diagram is used to represent the **flow of activities** within a system. It helps to understand the logic of a process and the relationships between the steps involved.

7. **State Chart Diagram (v.1):** This diagram is used to represent the **states and transitions of objects or components in a system**. It helps to understand the behavior of the system over time, including the possible states and events that trigger transitions between them.

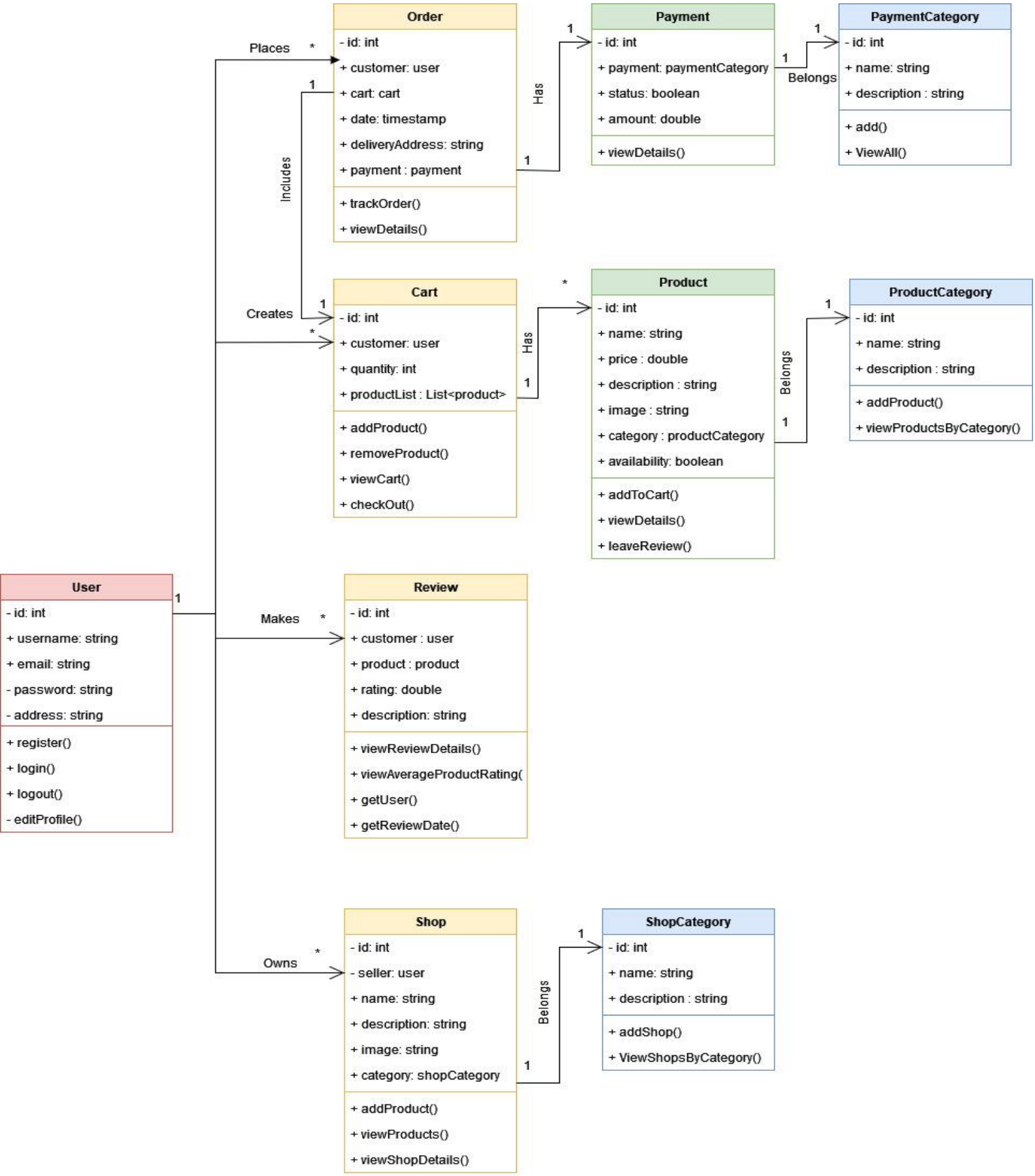
8. **Component Diagram (v.1):** This diagram is used to represent the components and the **relationships** between them in a system. It helps to understand the structure of the system and the dependencies between the components.

9. **Deployment Diagram (v.1):** This diagram is used to represent the **physical deployment** of the system, including the hardware and software components and their relationships. It helps to understand the environment in which the system will run and the dependencies between the components.

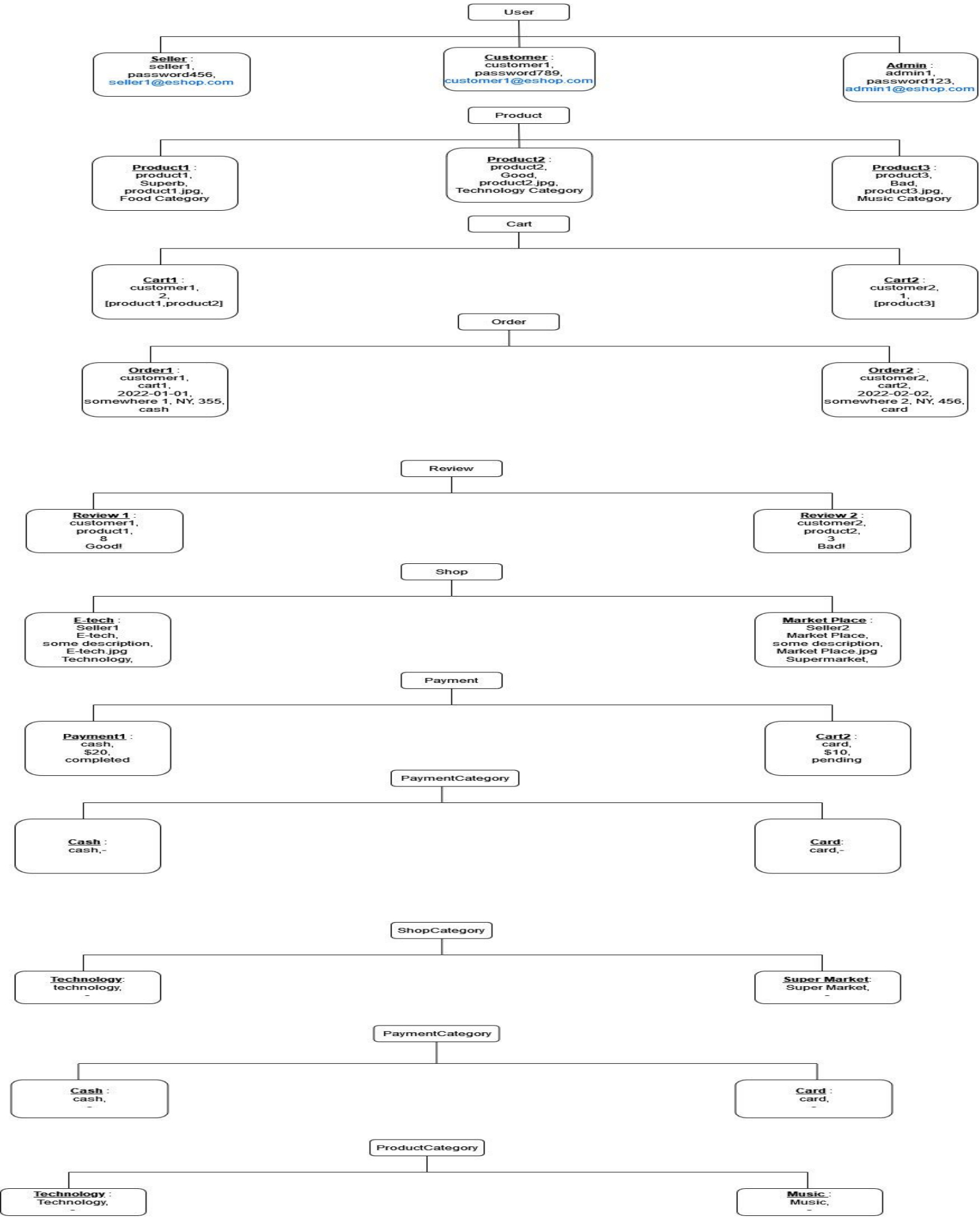
4.1.1 Use Case Diagram (2nd Version)



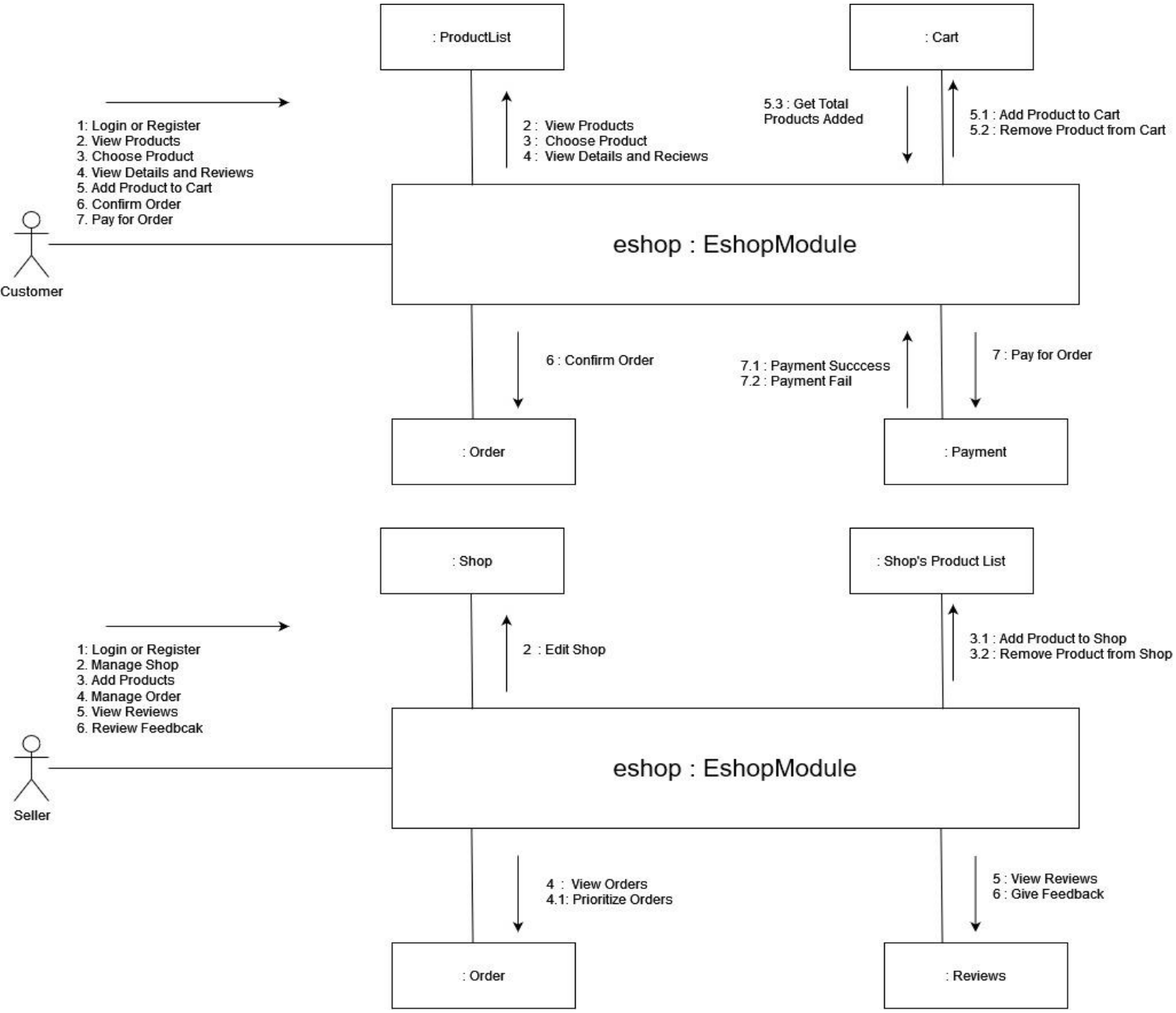
4.1.2 Class Diagram (2nd Version)



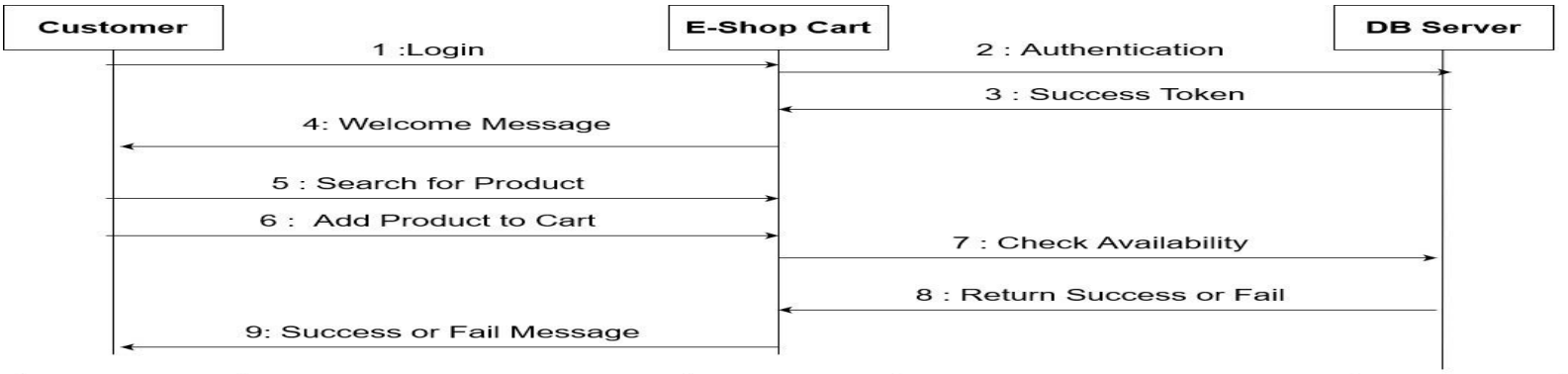
4.1.3 Object Diagram (1st Version)

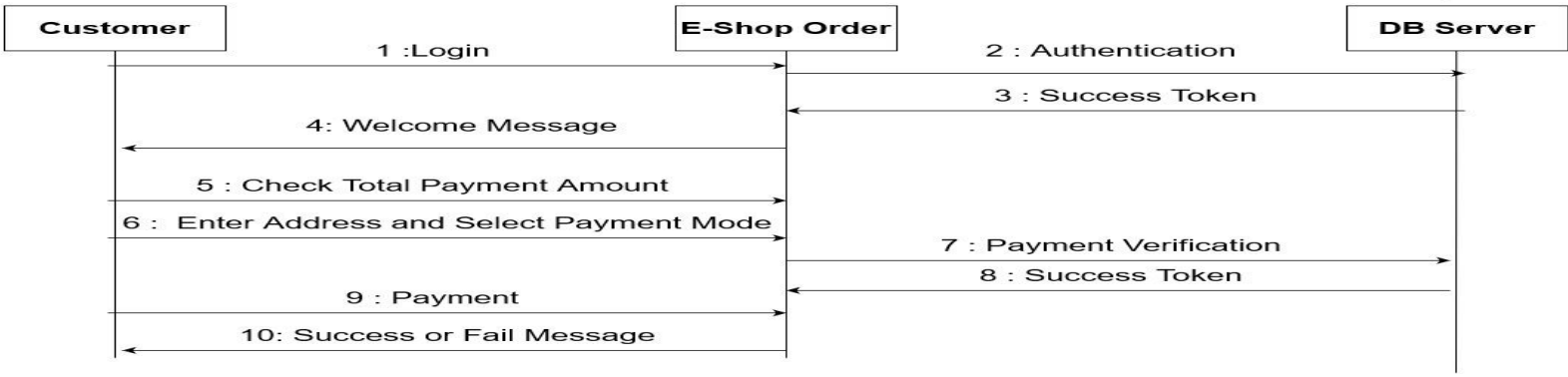


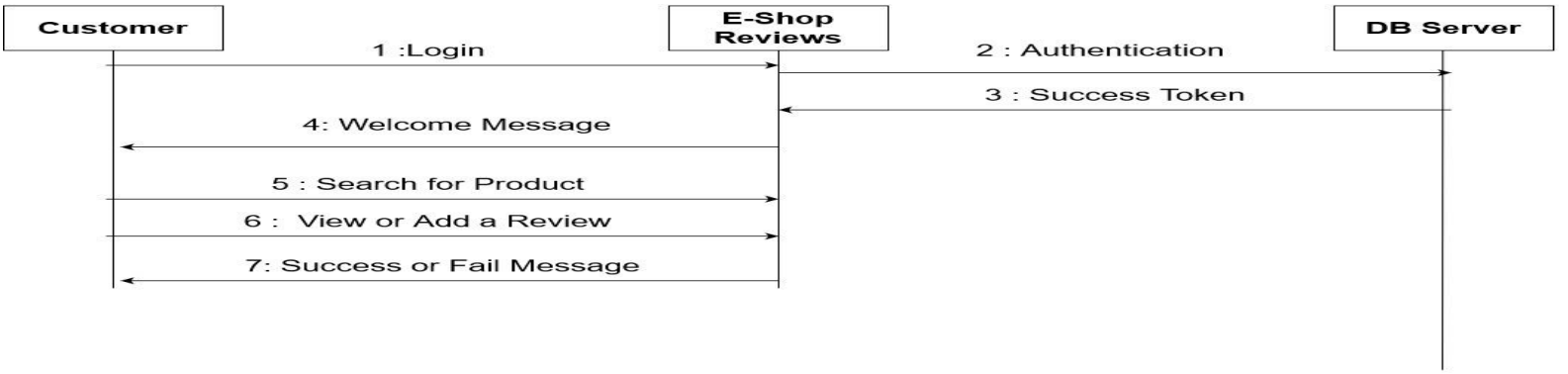
4.1.4 Collaboration or Communication Diagram (1st Version)

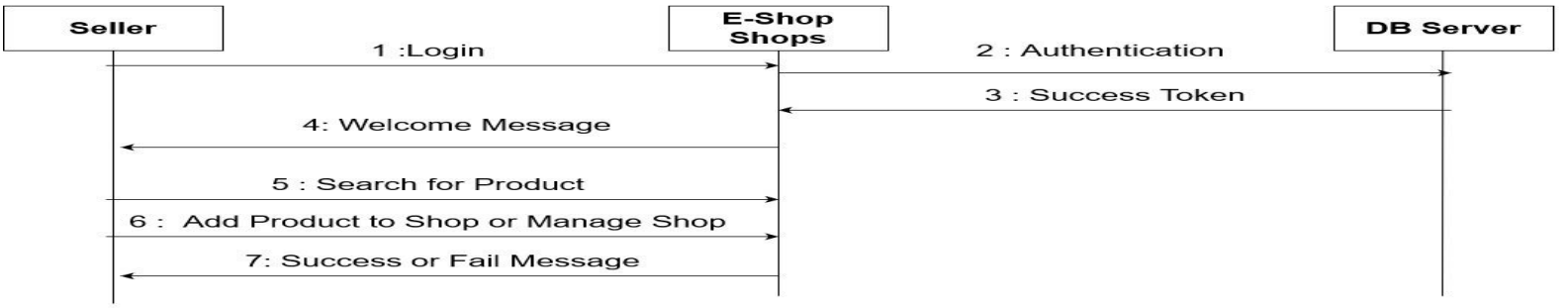


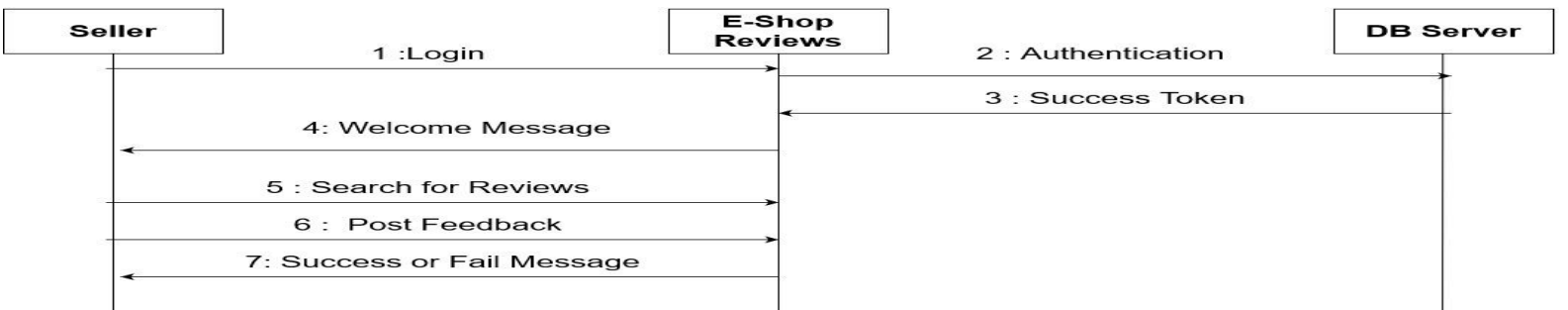
4.1.5 Sequence Diagram (1st Version)



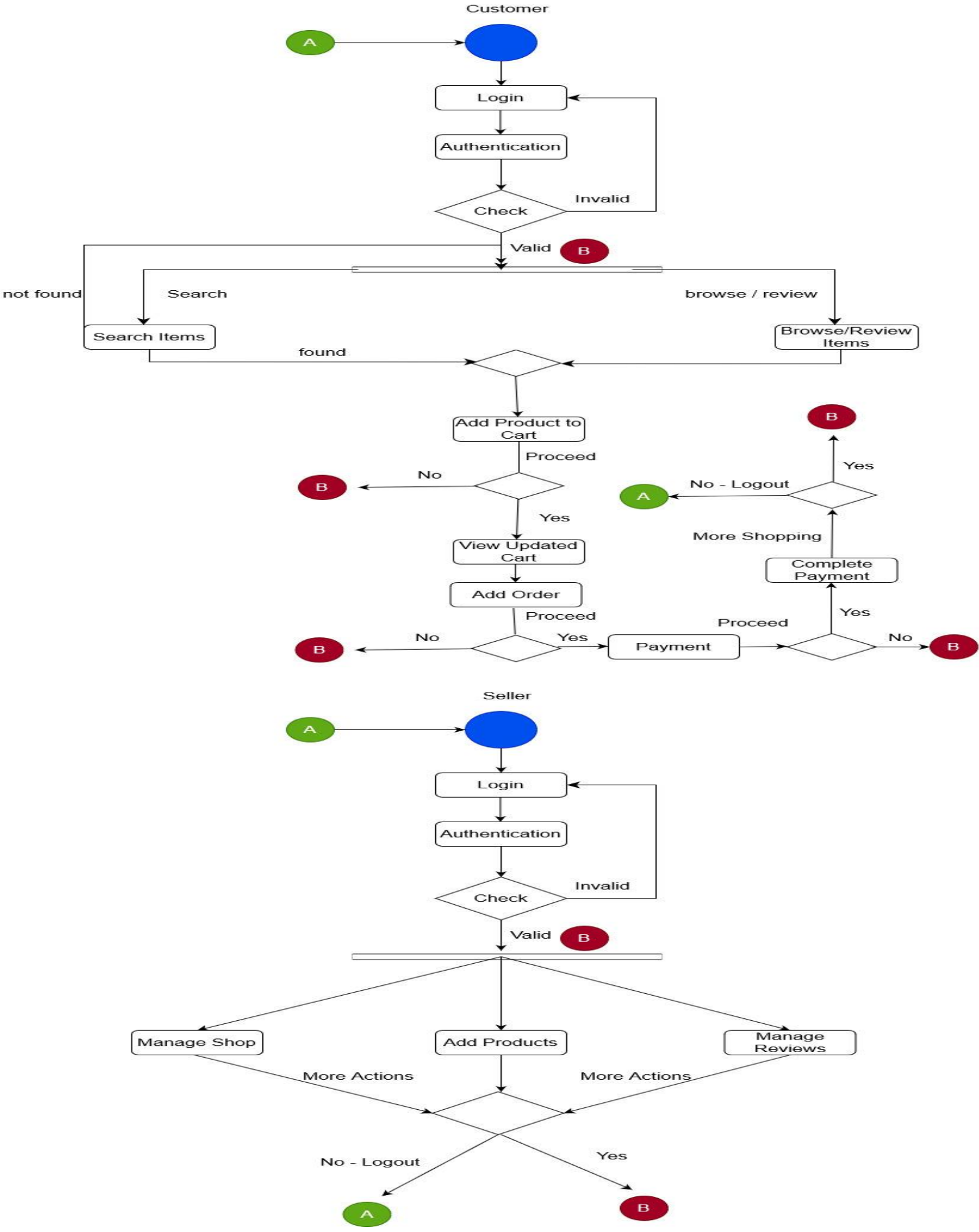




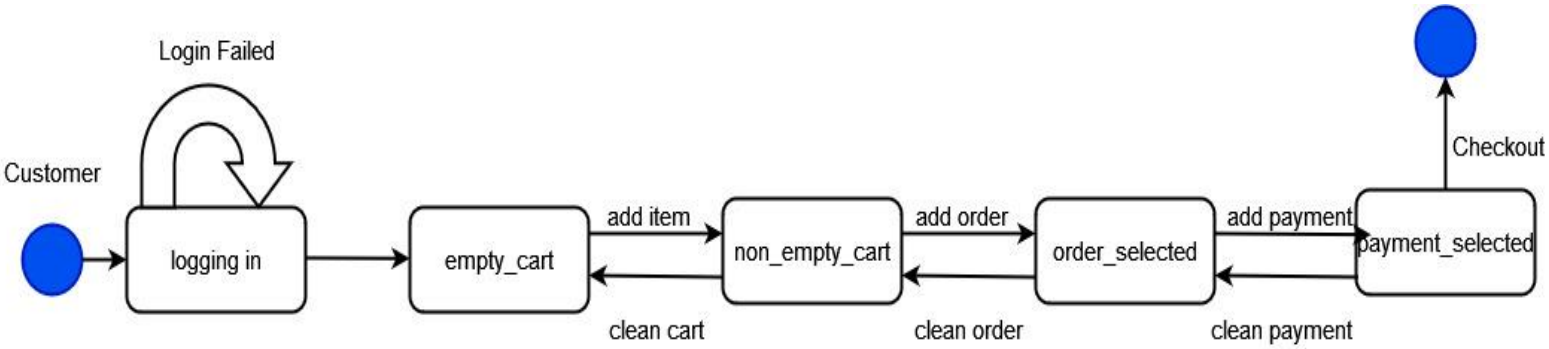


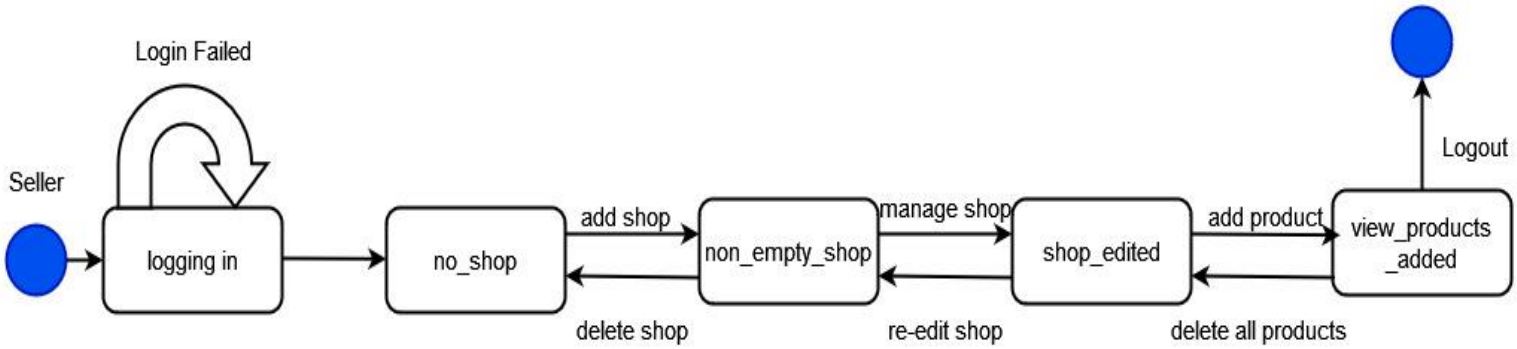


4.1.6 Activity Diagram (1st Version)

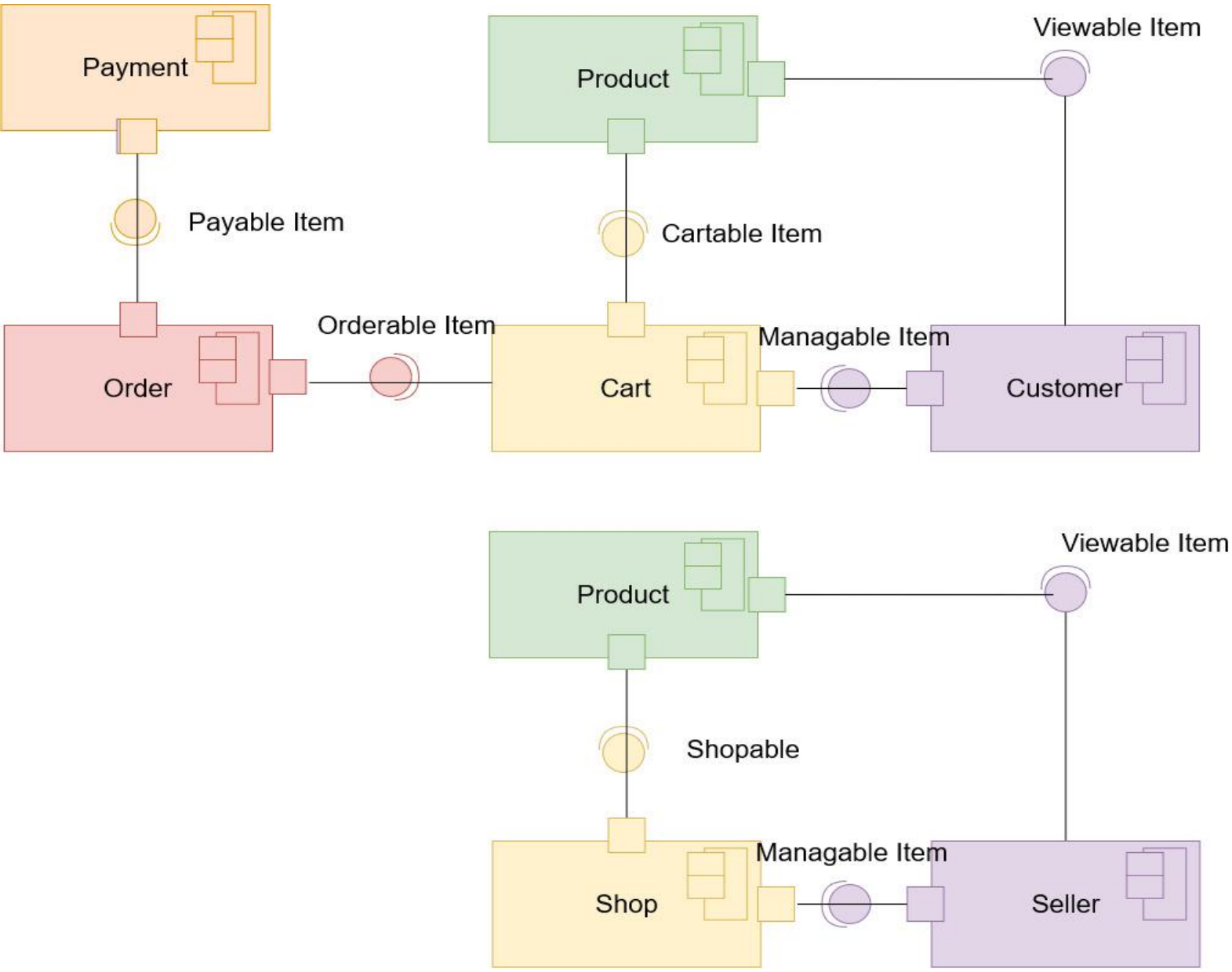


4.1.7 State Chart Diagram (1st Version)





4.1.8 Component Diagram (1st Version)



4.1.9 Deployment Diagram (1st Version)