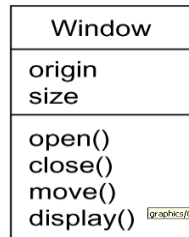


The UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system.

Class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces. Graphically, a class is rendered as a rectangle, usually including its name, attributes, and operations.



Interface

Interface is a collection of operations that specify a service of a class or component.

An interface therefore describes the externally visible behavior of that element.

An interface might represent the complete behavior of a class or component or only a part of that behavior.

An interface is rendered as a circle together with its name. An interface rarely stands alone. Rather, it is typically attached to the class or component that realizes the interface

I Spelling



Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger than the sum of all the elements. Therefore, collaborations have structural, as well as behavioral, dimensions. A given class might participate in several collaborations.

Graphically, a collaboration is rendered as an ellipse with dashed lines, usually including only its name



Usecase

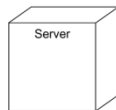
- Use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor
- Use case is used to structure the behavioral things in a model.
- A use case is realized by a collaboration. Graphically, a use case is rendered as an ellipse with solid lines, usually including only its name



Component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs



Node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability. Graphically, a node is rendered as a cube, usually including only its name



Interaction

Interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose

An interaction involves a number of other elements, including messages, action sequences and links

Graphically a message is rendered as a directed line, almost always including the name of its operation

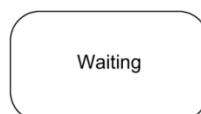


State Machine

State machine is a behavior that specifies the sequences of states an object or an interaction goes through during its lifetime in response to events, together with its responses to those events

State machine involves a number of other elements, including states, transitions, events and activities

Graphically, a state is rendered as a rounded rectangle, usually including its name and its substates



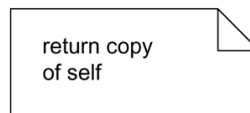
Package:-

- A package is a general-purpose mechanism for organizing elements into groups. Structural things, behavioral things, and even other grouping things may be placed in a package
- Graphically, a package is rendered as a tabbed folder, usually including only its name and, sometimes, its contents



A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements.

Graphically, a note is rendered as a rectangle with a dog-eared corner, together with a textual or graphical comment



Relationships in the UML: There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

Dependency:-

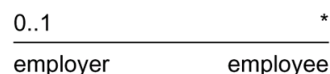
Dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing

Graphically a dependency is rendered as a dashed line, possibly directed, and occasionally including a label



Association is a structural relationship that describes a set of links, a link being a connection among objects.

Graphically an association is rendered as a solid line, possibly directed, occasionally including a label, and often containing other adornments, such as multiplicity and role names



Aggregation is a special kind of association, representing a structural relationship between a whole and its parts. Graphically, a generalization relationship is rendered as a solid line with a hollow arrowhead pointing to the parent



Realization is a semantic relationship between classifiers, wherein one classifier specifies a contract that another classifier guarantees to carry out. Graphically a realization relationship is rendered as a cross between a generalization and a dependency relationship



Diagrams in the UML

- **Diagram** is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships).
- In theory, a diagram may contain any combination of things and relationships.
- For this reason, the UML includes nine such diagrams:
 - Class diagram
 - Object diagram
 - Use case diagram
 - Sequence diagram
 - Collaboration diagram
 - Statechart diagram
 - Activity diagram
 - Component diagram
 - Deployment diagram

Class diagram

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams that include active classes address the static process view of a system.

Object diagram

- Object diagrams represent static snapshots of instances of the things found in class diagrams
- These diagrams address the static design view or static process view of a system
- An object diagram shows a set of objects and their relationships

Use case diagram

- A use case diagram shows a set of use cases and actors and their relationships
- Use case diagrams address the static use case view of a system.
- These diagrams are especially important in organizing and modeling the behaviors of a system.

Interaction Diagrams

Both sequence diagrams and collaboration diagrams are kinds of interaction diagrams

Interaction diagrams address the dynamic view of a system

A sequence diagram is an interaction diagram that emphasizes the time-ordering of messages

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages

Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other

Statechart diagram

- A statechart diagram shows a state machine, consisting of states, transitions, events, and activities
- Statechart diagrams address the dynamic view of a system
- They are especially important in modeling the behavior of an interface, class, or collaboration and emphasize the event-ordered behavior of an object

Activity diagram

An activity diagram is a special kind of a statechart diagram that shows the flow from activity to activity within a system

Activity diagrams address the dynamic view of a system

They are especially important in modeling the function of a system and emphasize the flow of control among objects.

Component diagram

- A component diagram shows the organizations and dependencies among a set of components.
- Component diagrams address the static implementation view of a system
- They are related to class diagrams in that a component typically maps to one or more classes, interfaces, or collaborations.

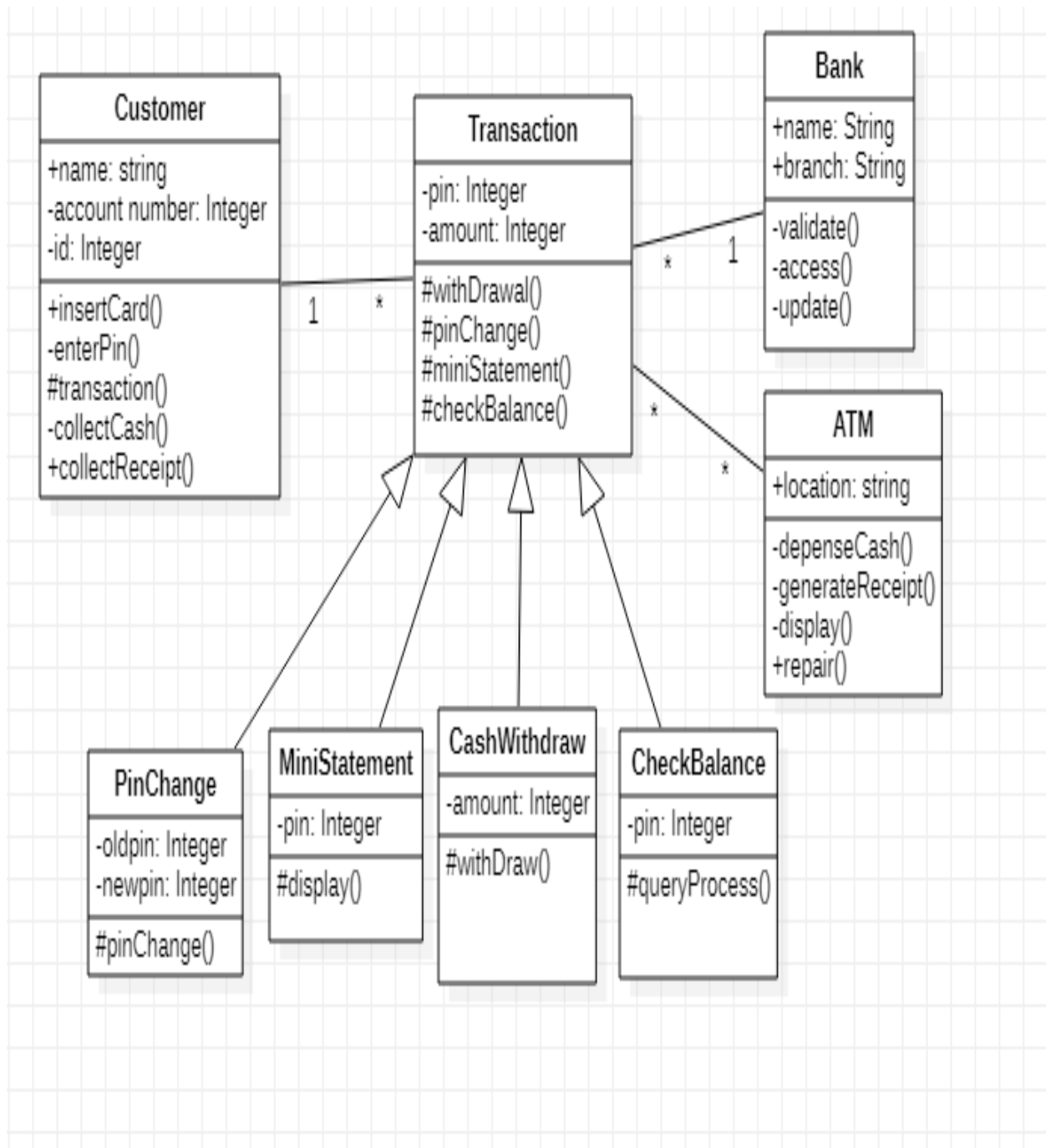
Deployment diagram

- A deployment diagram shows the configuration of run-time processing nodes and the components that live on them
- Deployment diagrams address the static deployment view of an architecture

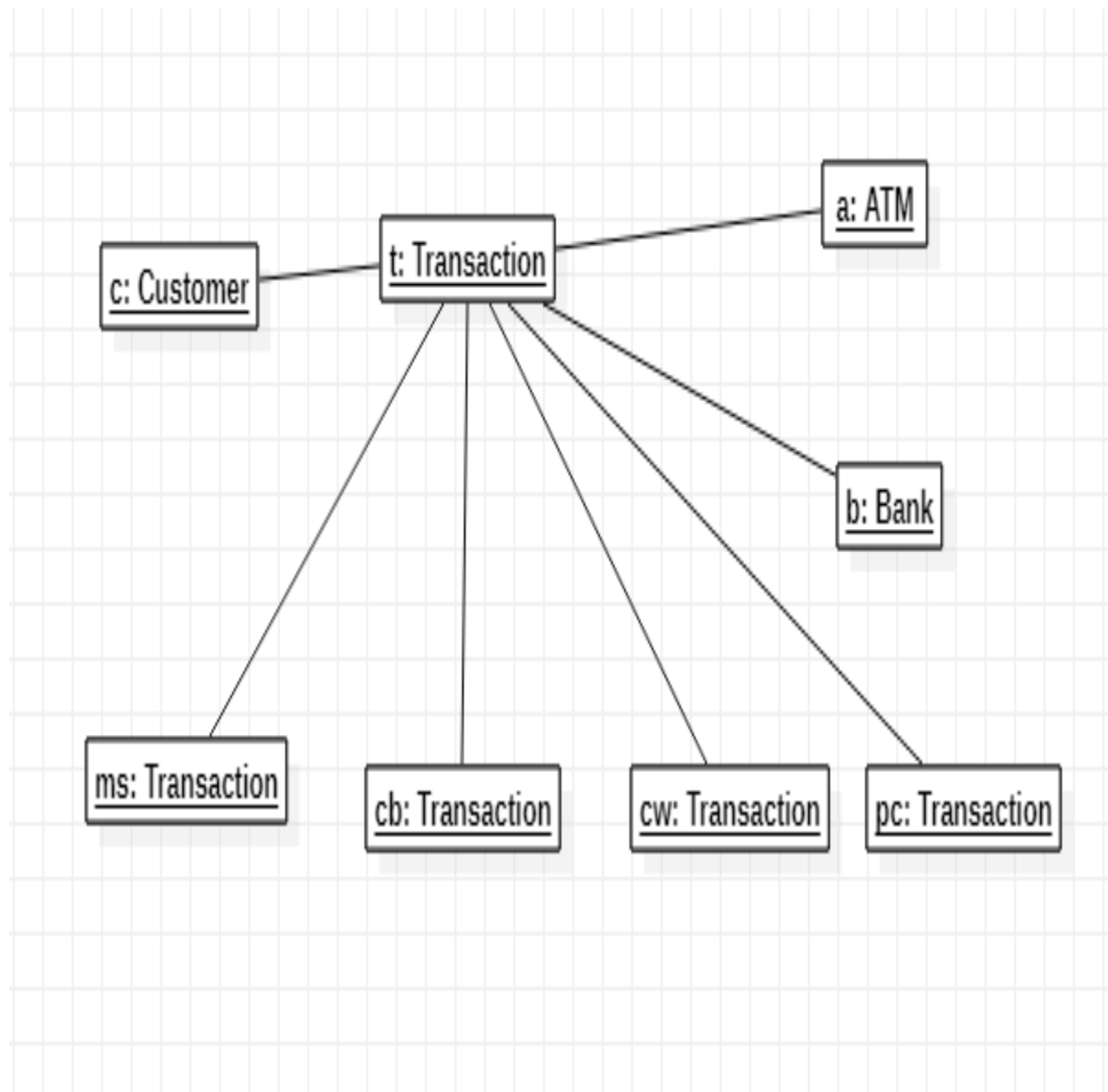
Why we need UML Lab?

The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. The overall goal of UML diagrams is to allow teams to visualize how a project is or will be working, and they can be used in any field, not just software engineering.

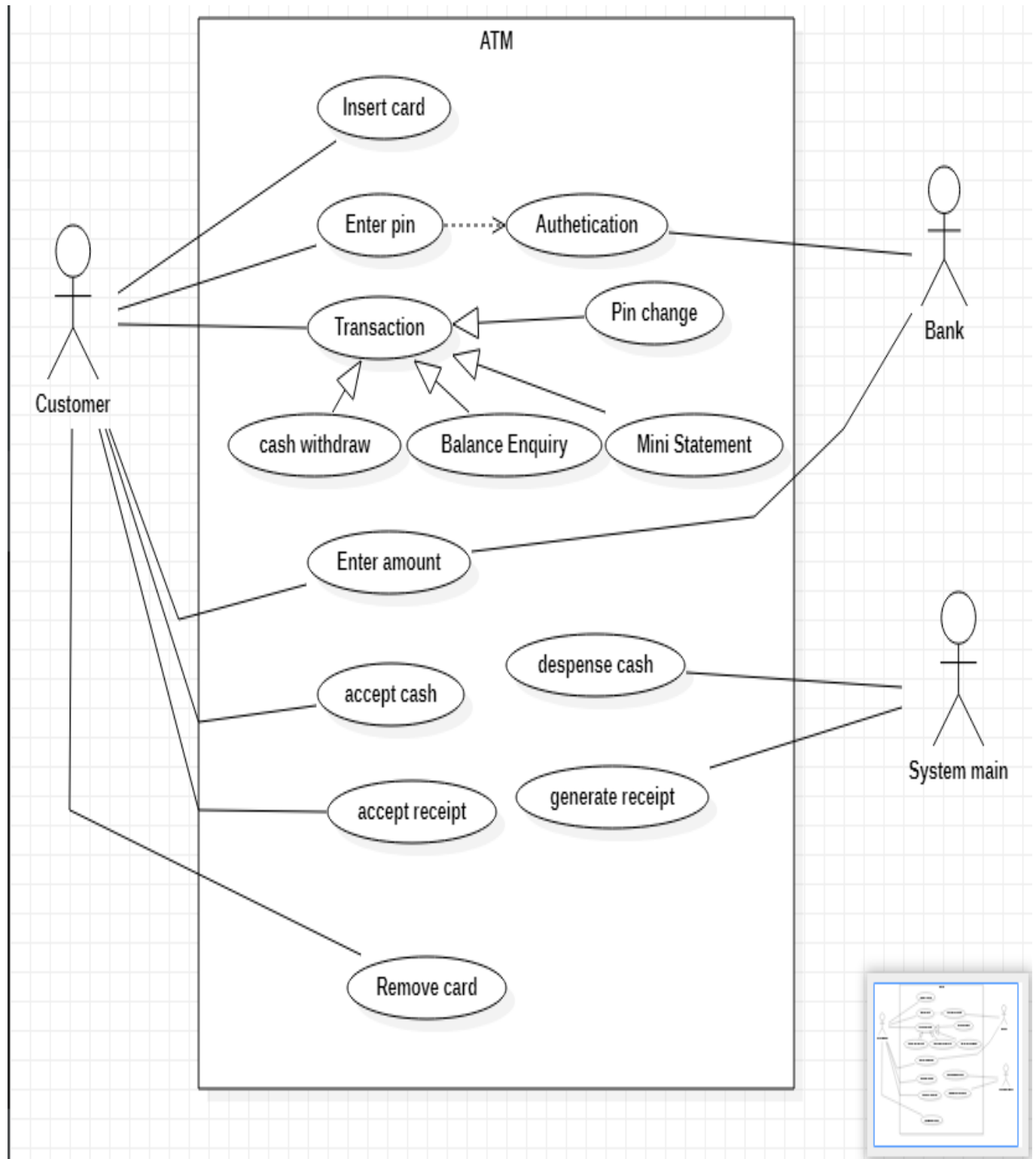
Task-1 Draw basic Class Diagram And Object Diagram to identify and describe key concepts like classes,and their relations for ATM?



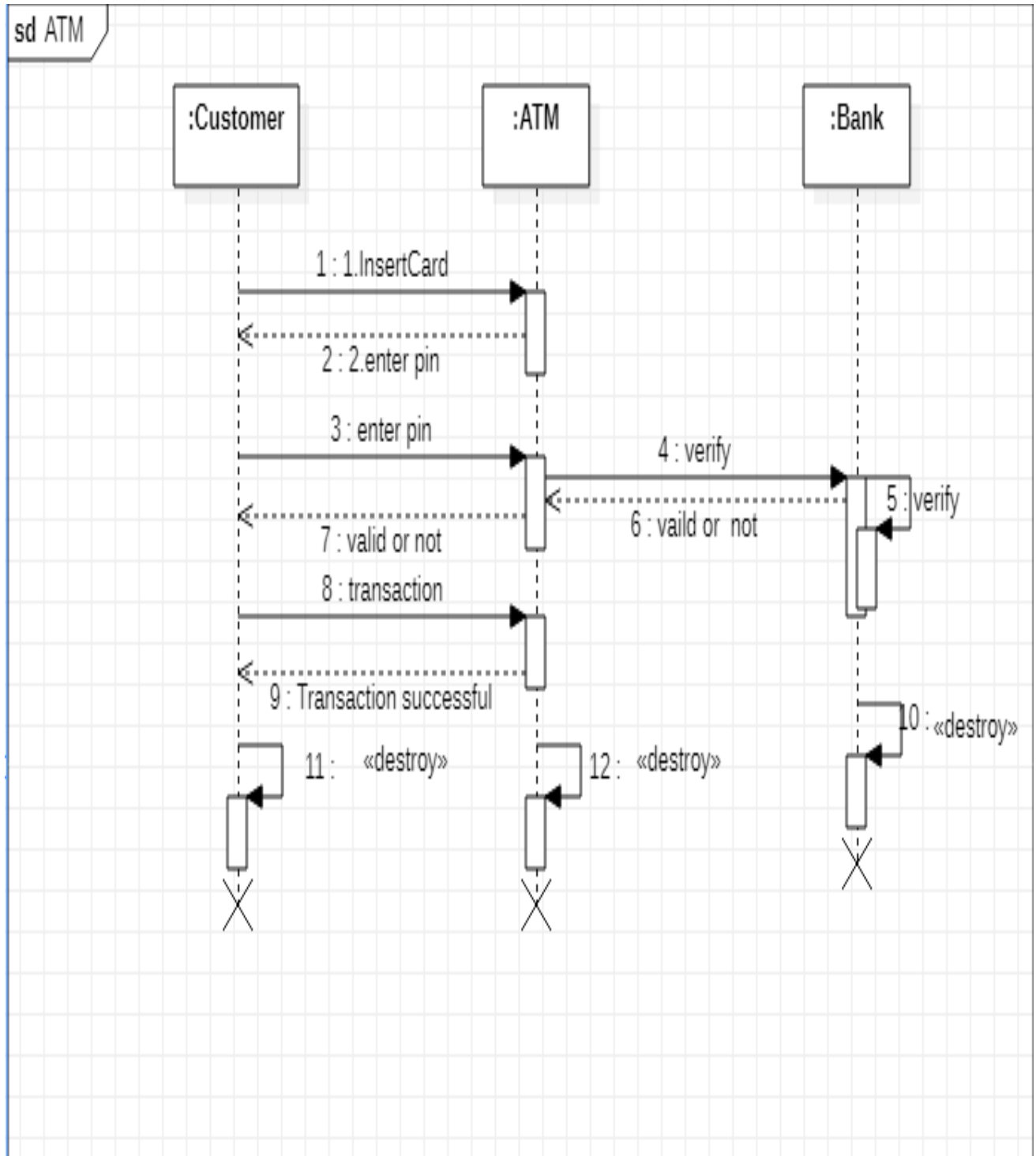
Object Diagram



Task-2 Draw one or more Use Case diagrams for capturing and representing requirements of the system. Use case diagrams must include template showing description and steps of the Use Case for various scenarios for ATM?

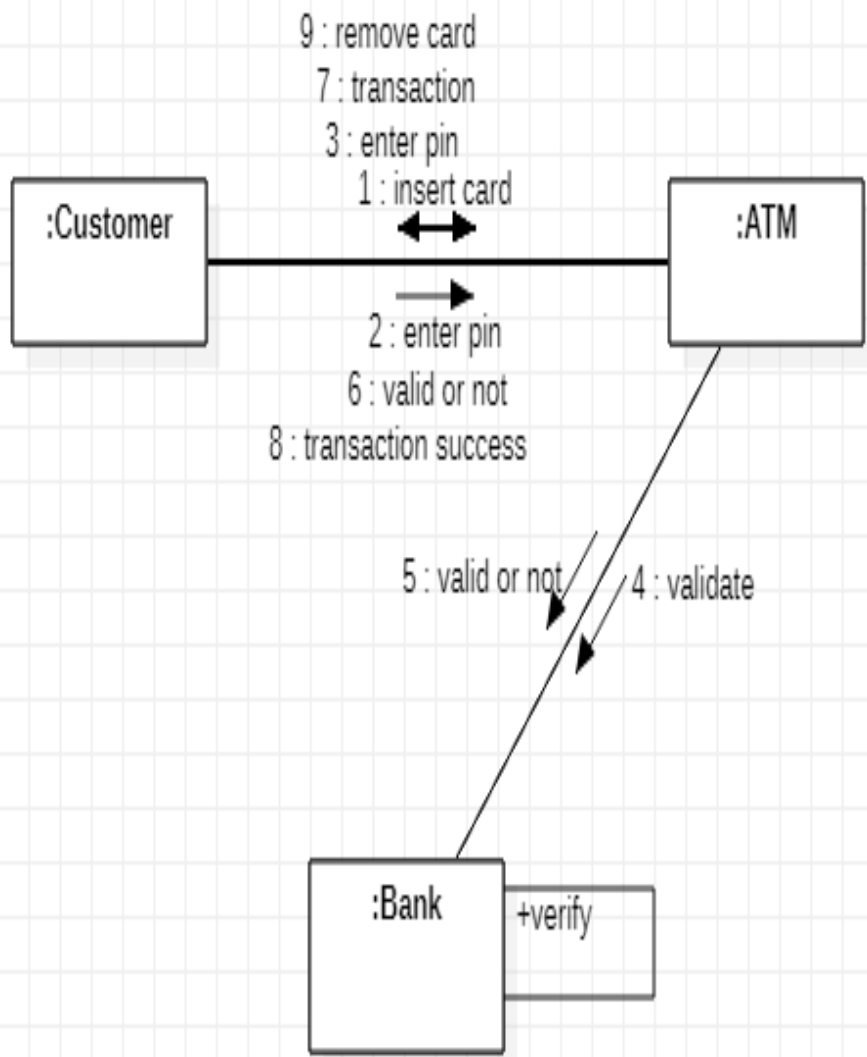


Task- 3 Draw sequence diagrams OR communication diagrams with advanced notation for system to show objects and their message exchanges.for ATM?

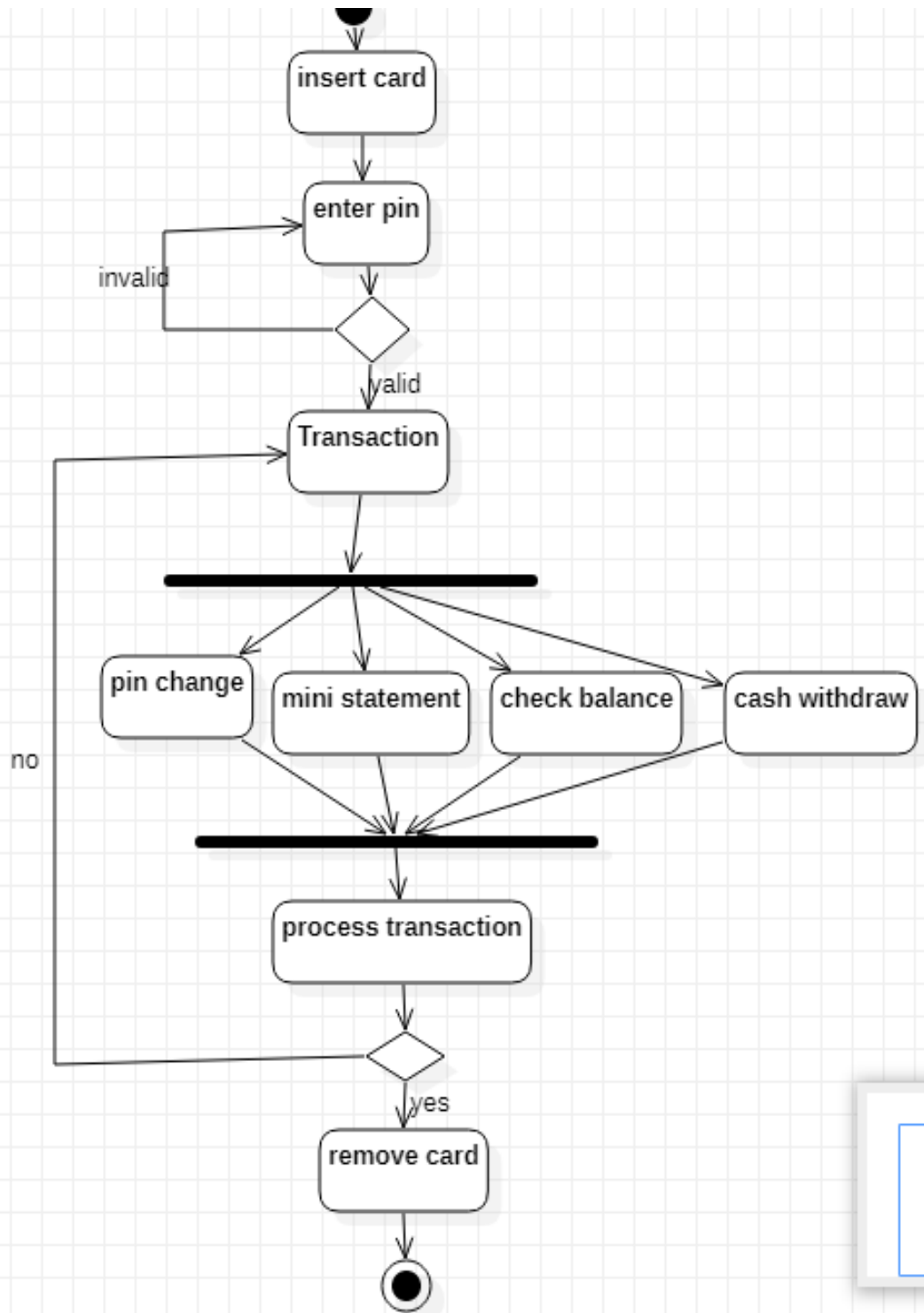


Communication diagram

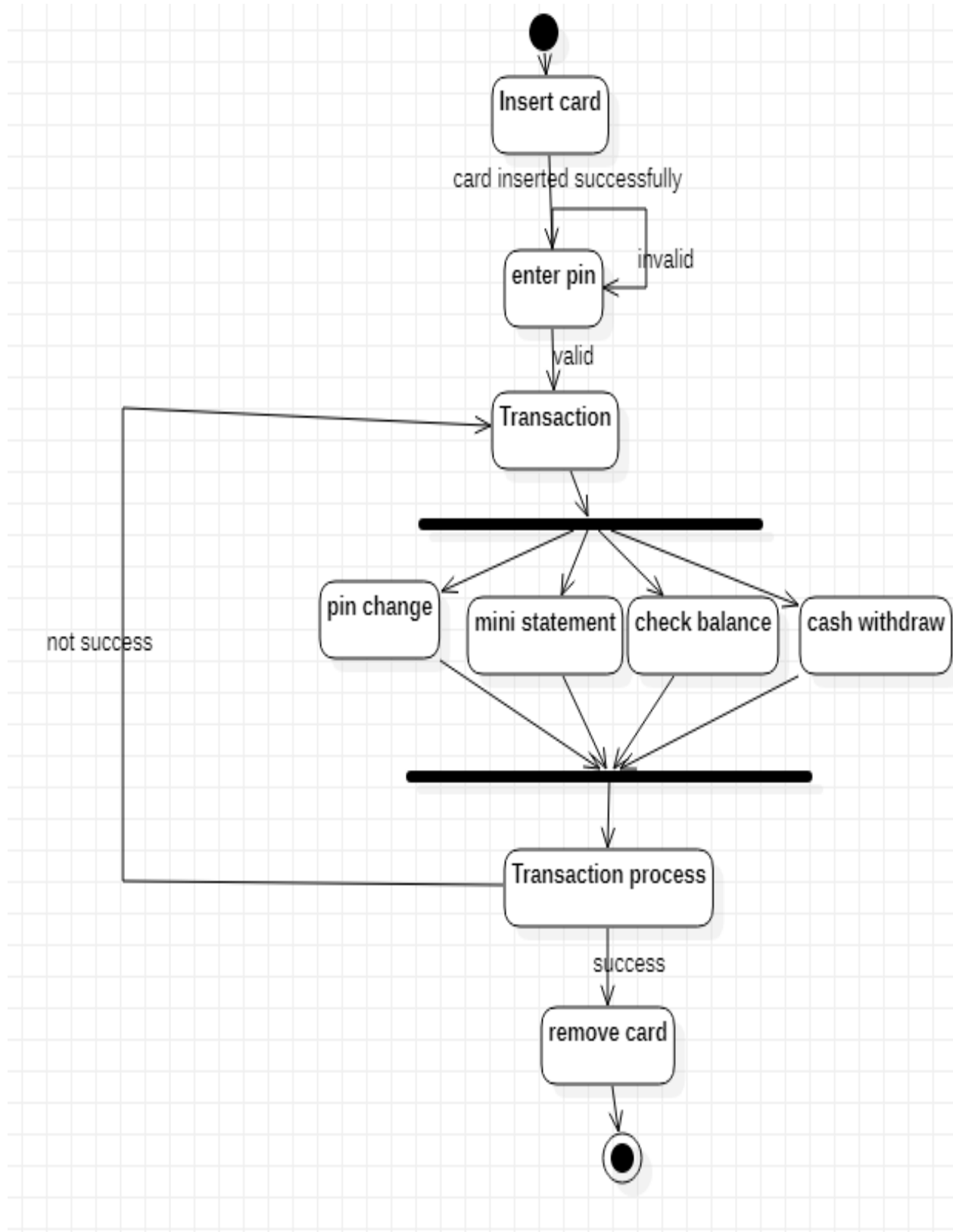
sd ATM



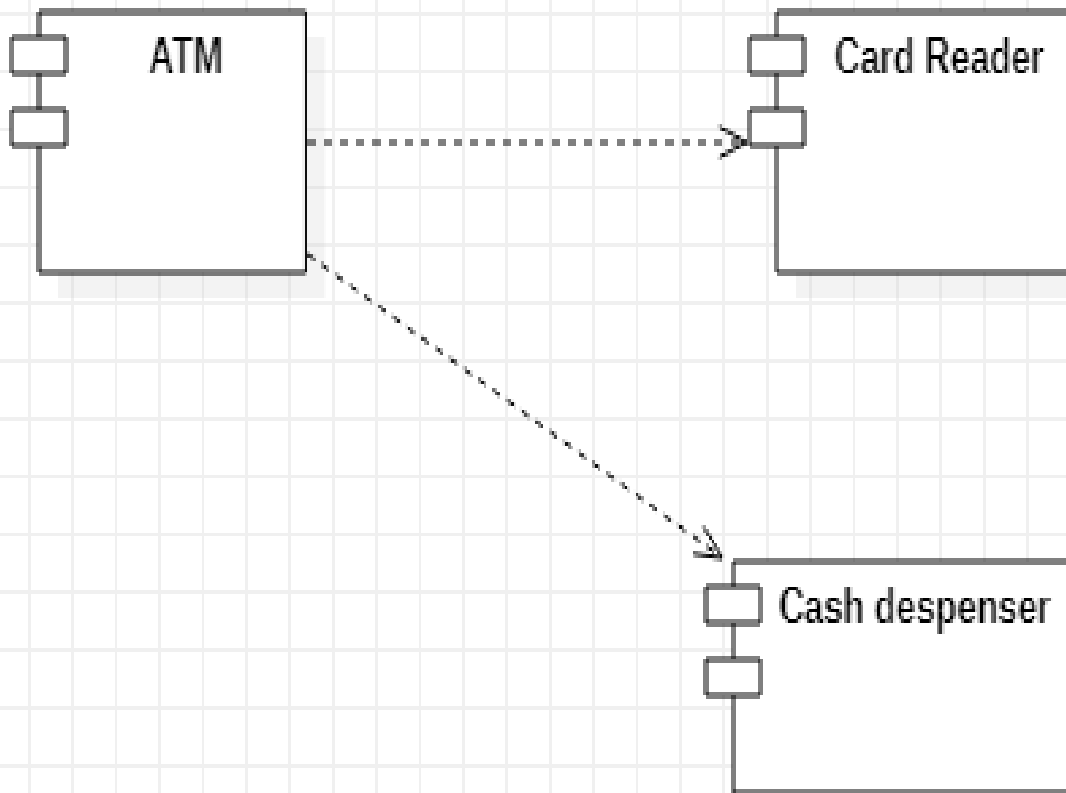
Task- 4 Draw activity diagrams to display either business flows or like flow charts for ATM?



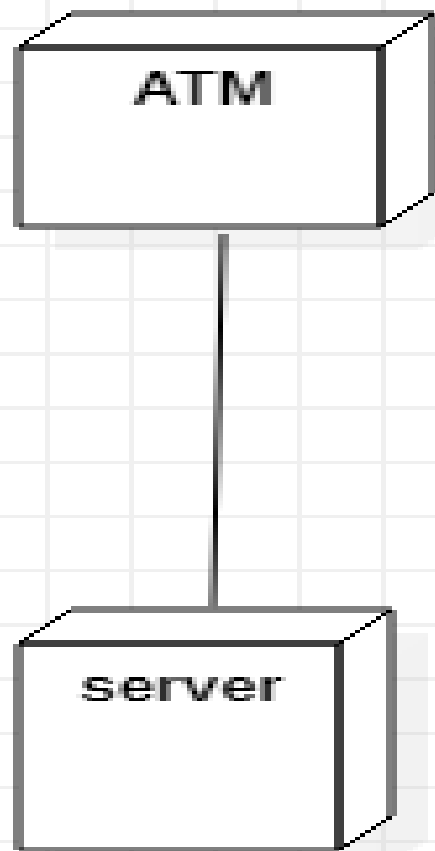
Task-5 Develop State chart diagrams for ATM?



Task-6 Draw component diagrams assuming that build the system reusing existing components along with a few new ones for ATM?

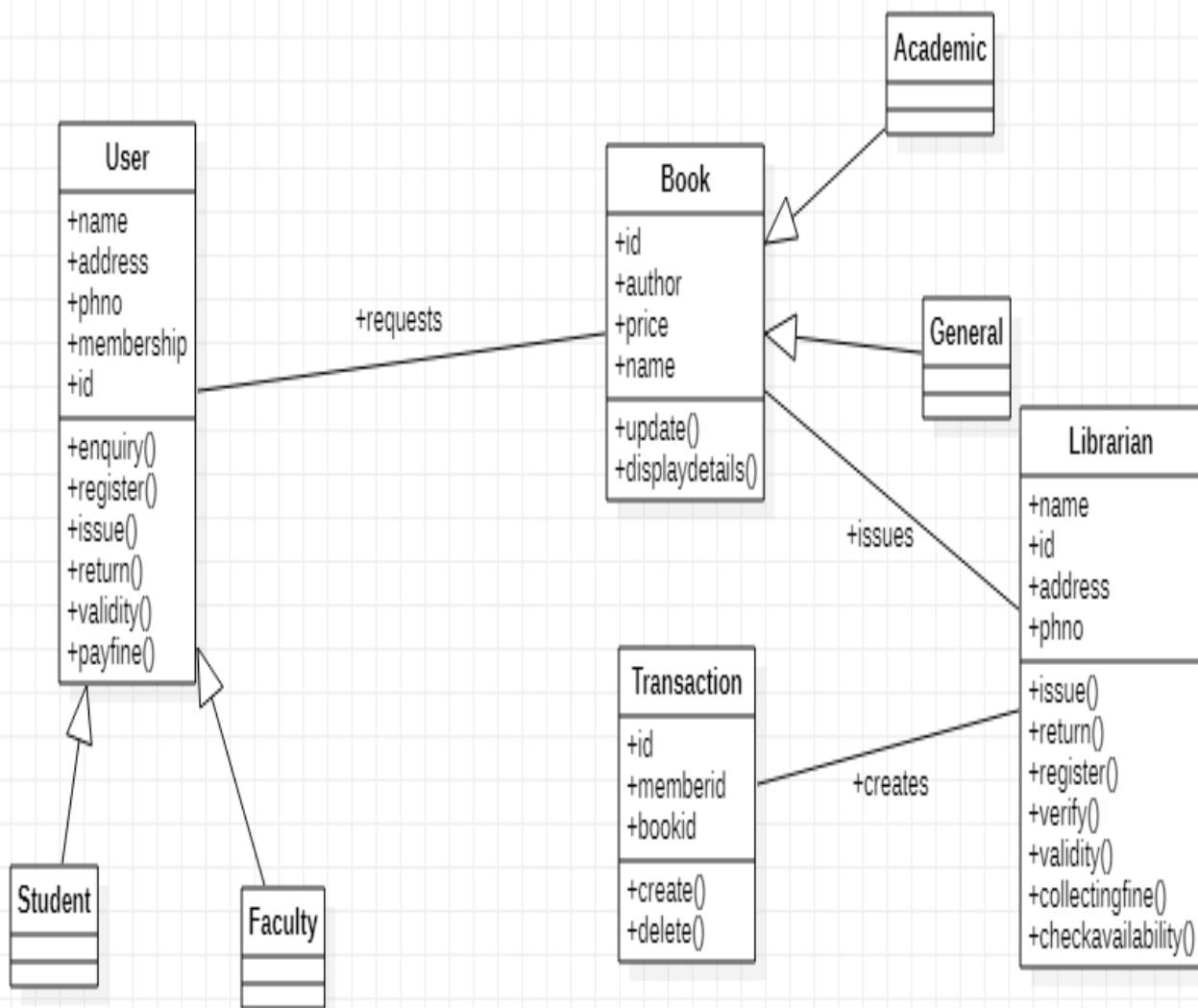


Task-7 Draw deployment diagrams to model the runtime architecture of system for ATM?

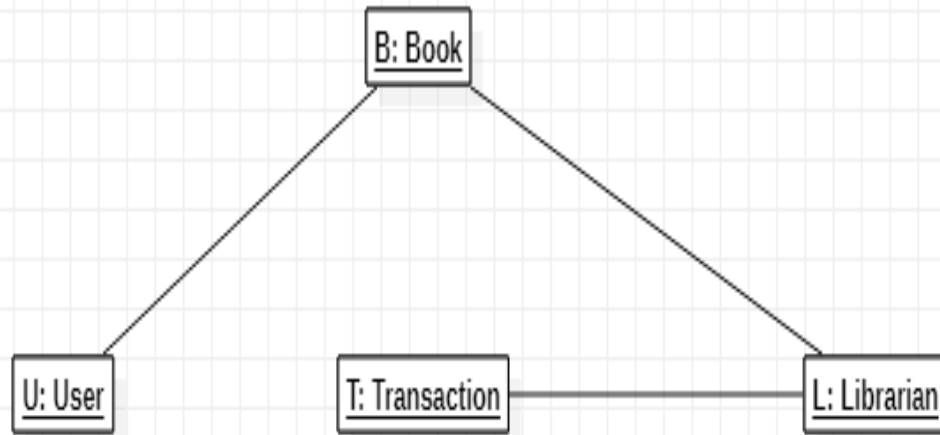


Task-8 Case study for Library Management System.

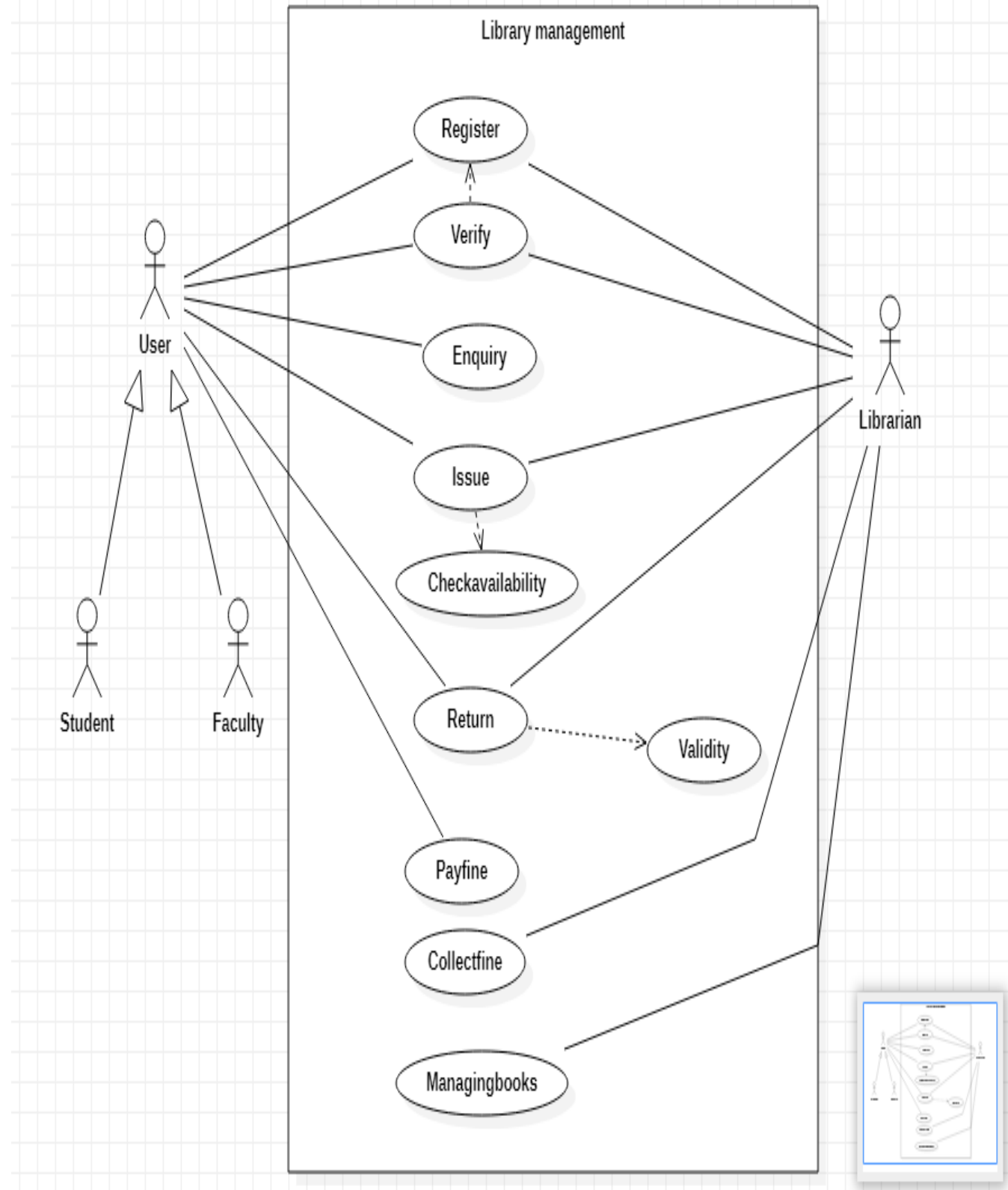
8.1 Class Diagram Library Management



Object Diagram Library Management

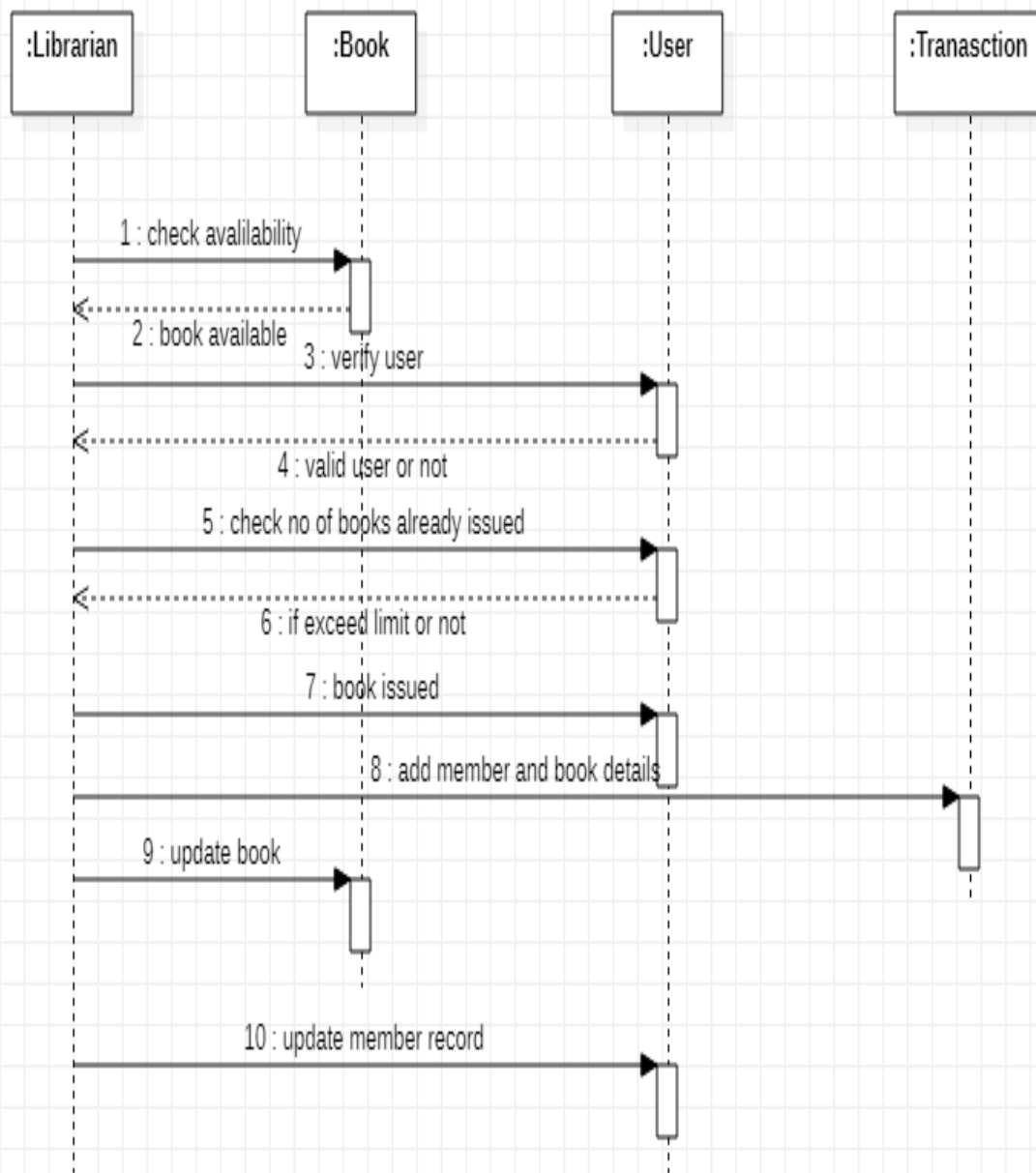


8.2 Use Case For Library Management

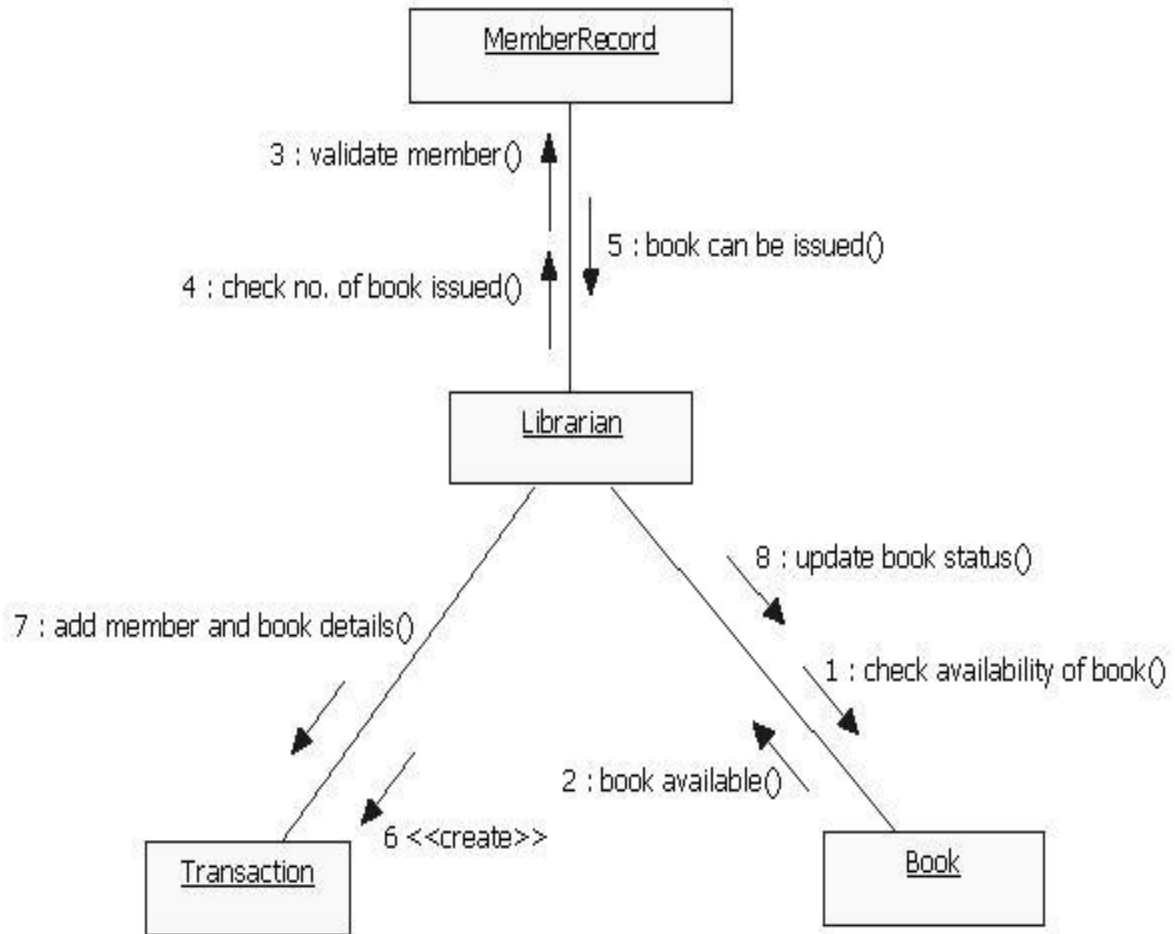


8.3 Sequence Diagram For Library Management

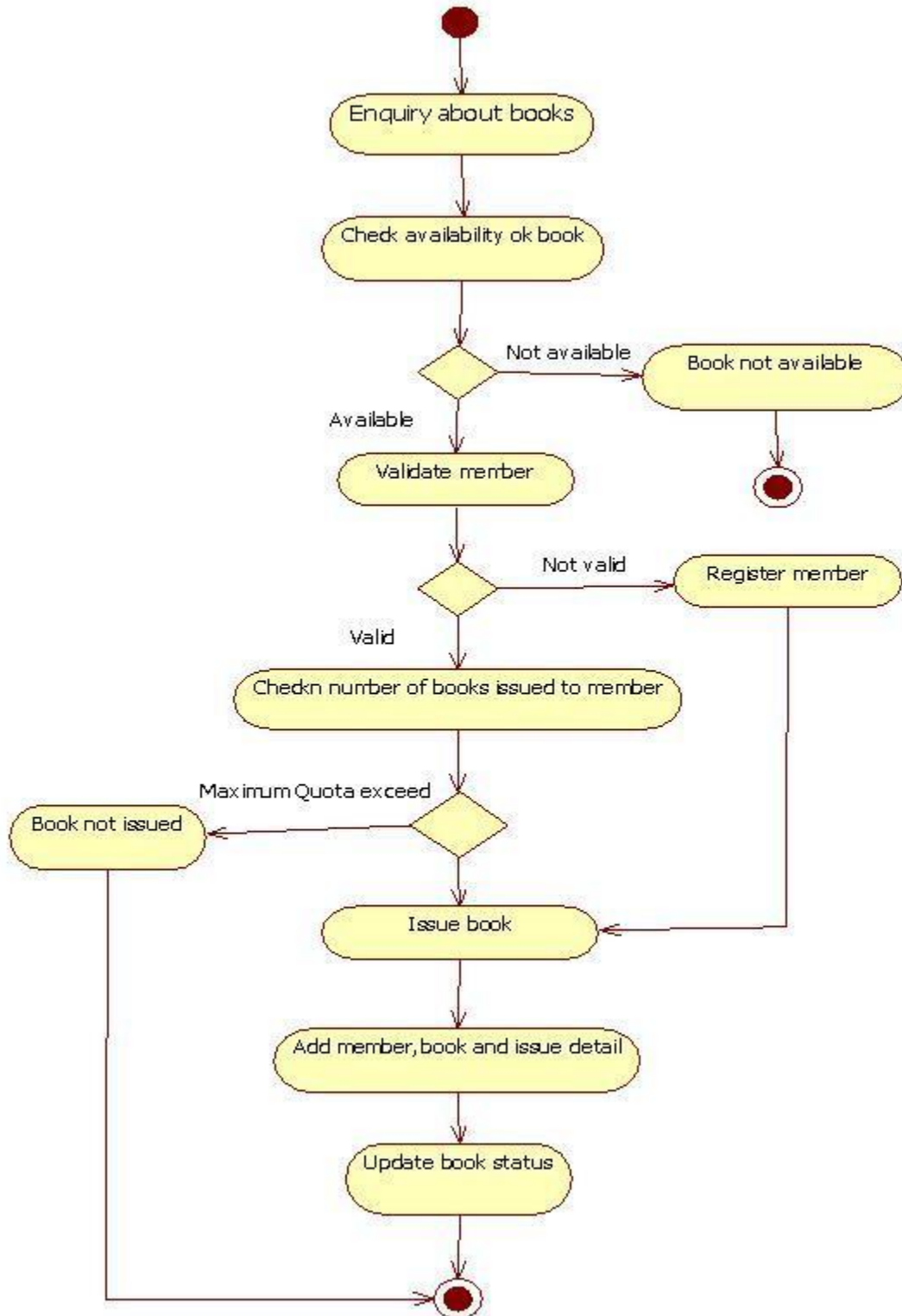
sd SequenceDiagram1



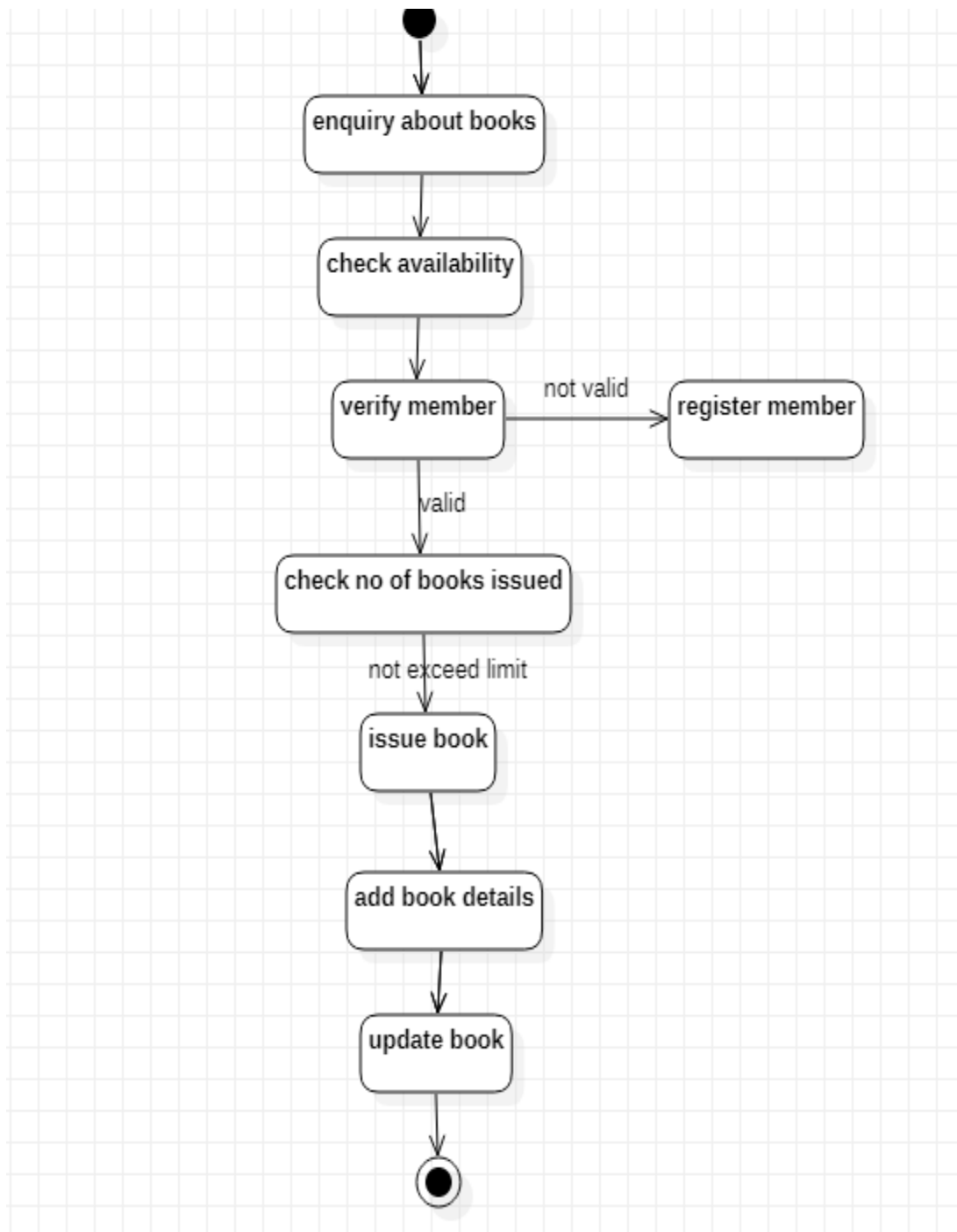
Collaboration Diagram For Library



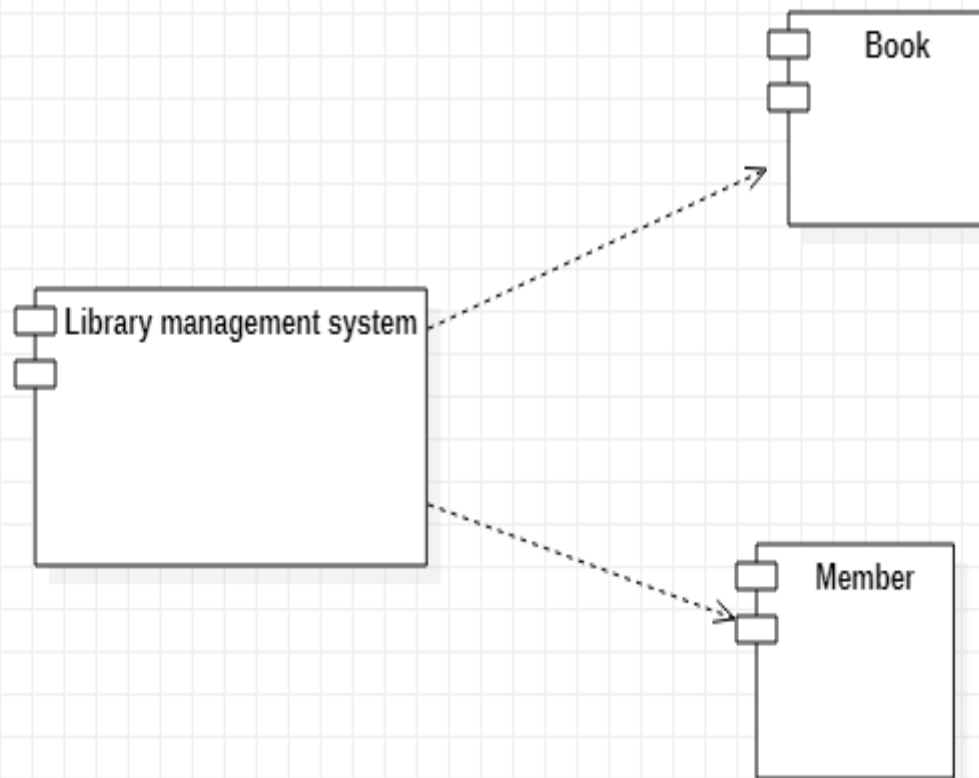
8.4 Activity Diagram Library Management



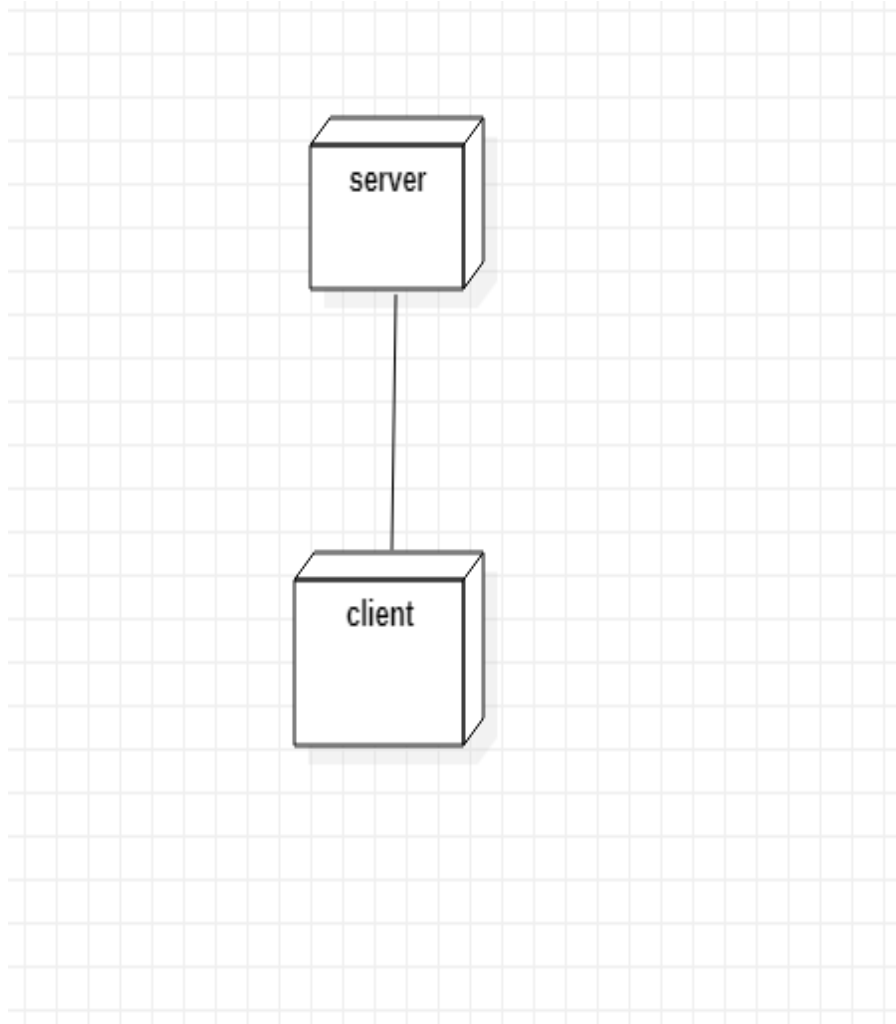
8.5 State chart Diagram For Library



8.6 Component Diagram for Library

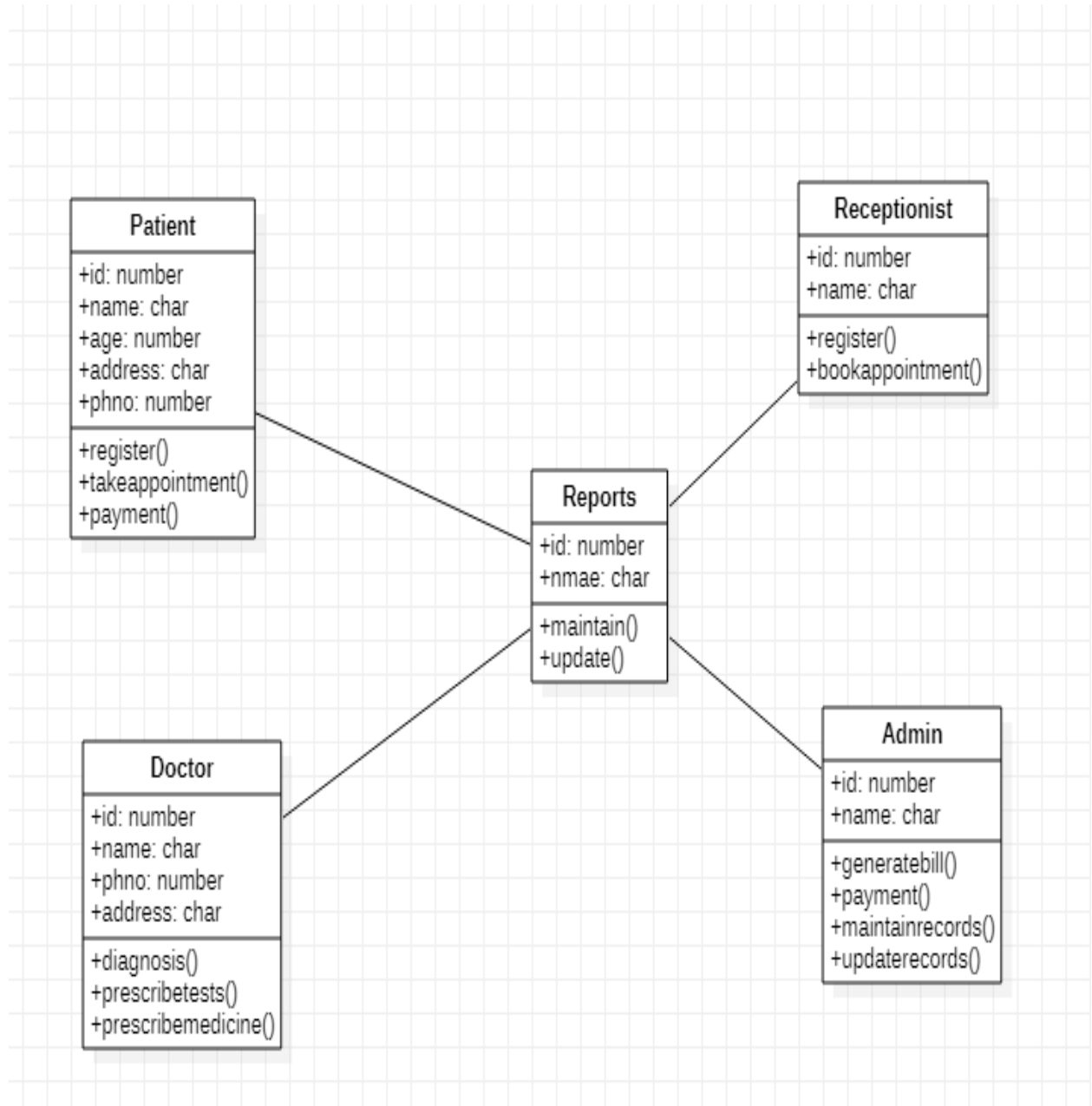


8.7 Deployment Diagram For Library

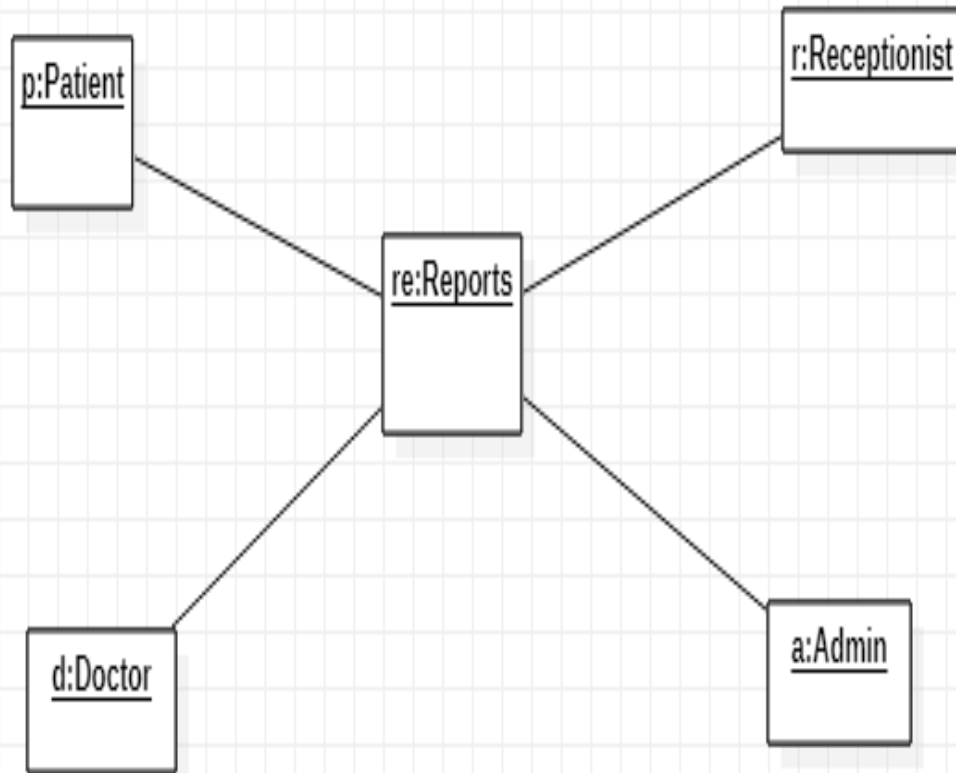


Task-9 Case study for Hospital Management System.

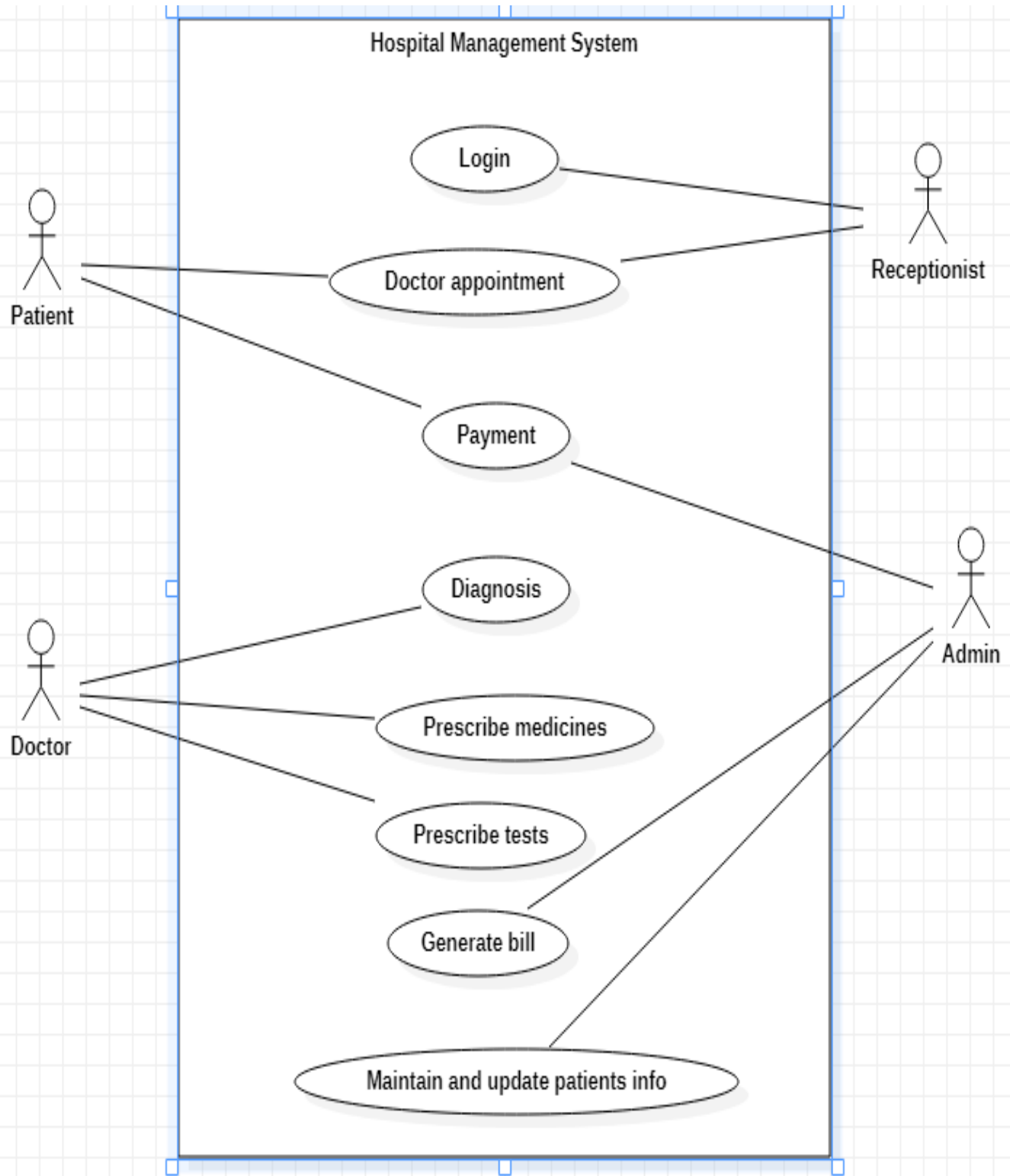
9.1.Class Diagram for Hospital Mgmt



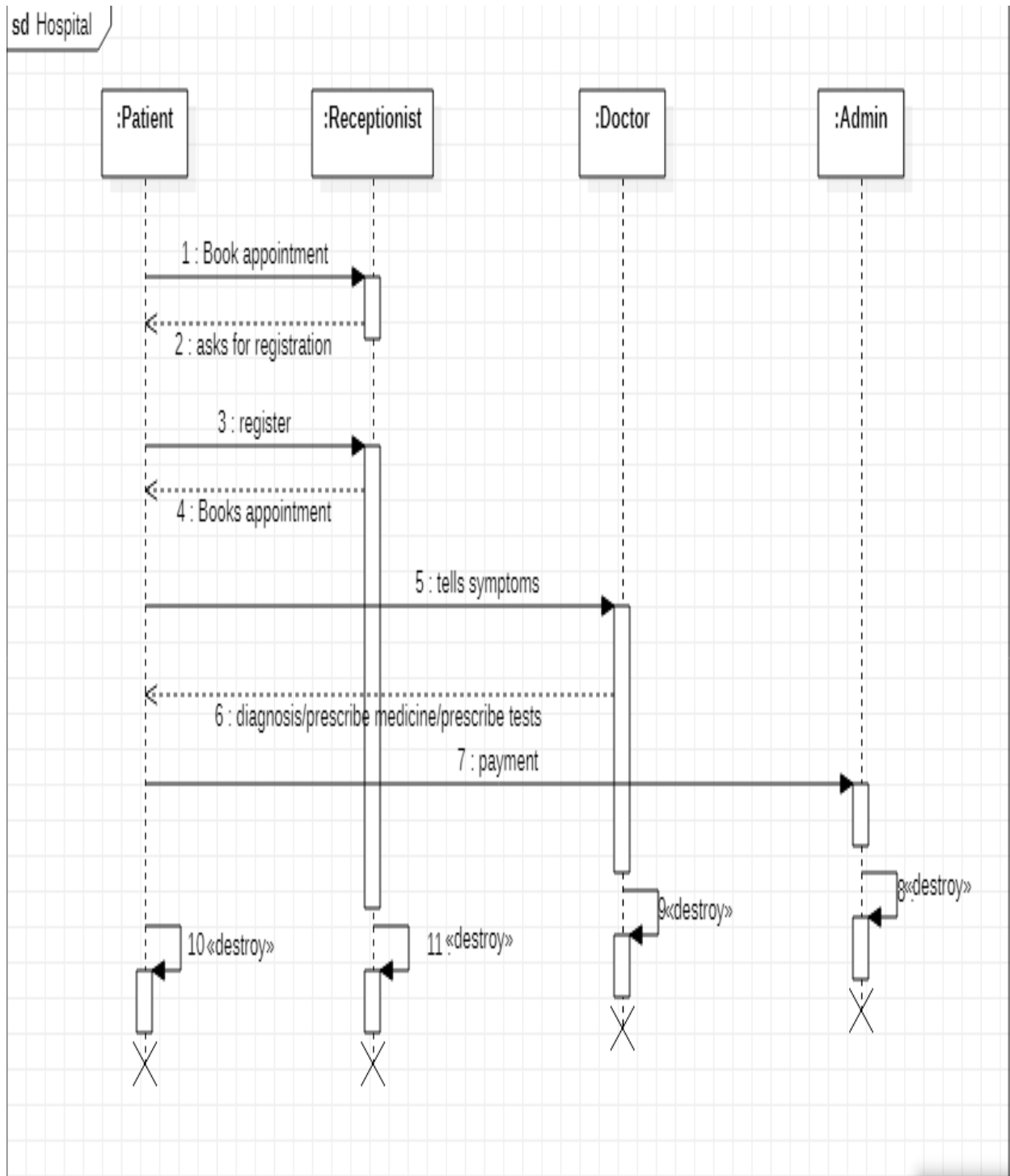
Object Diagram for Hospital Mgmt



9.2. Use Case Diagram for Hospital Mgmt

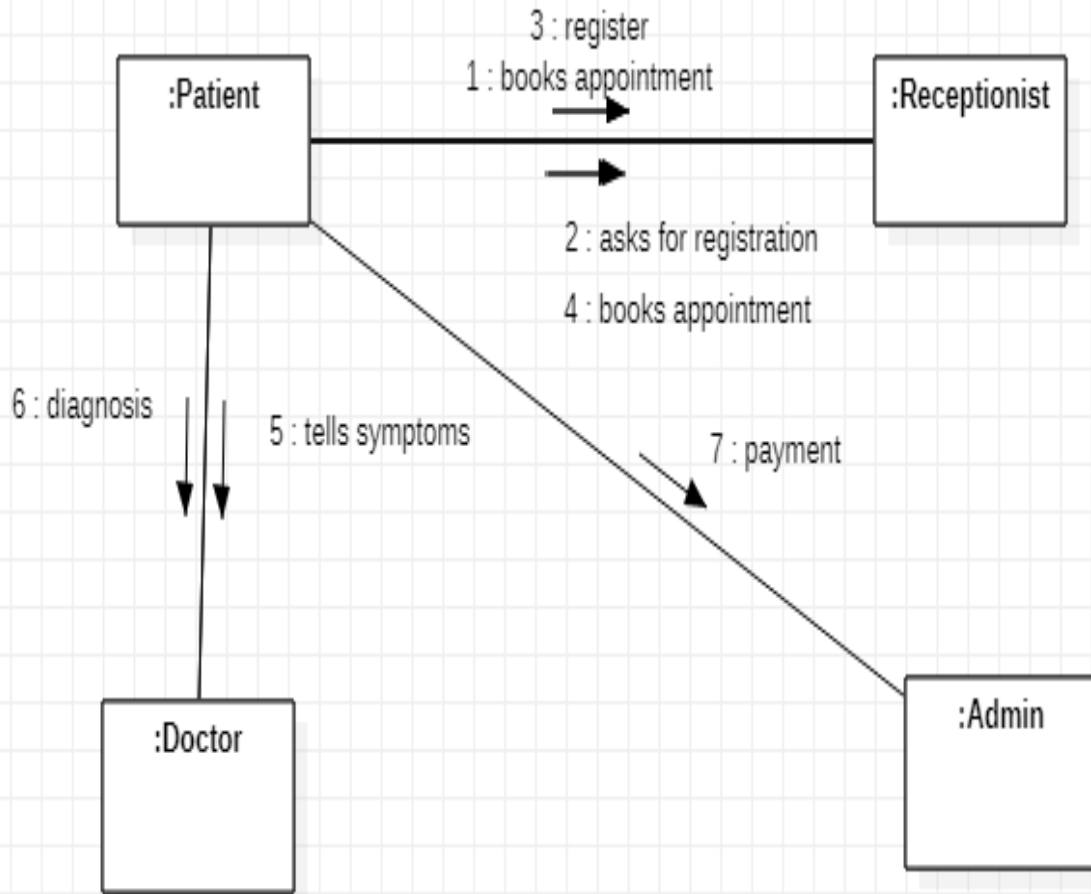


9.3.sequence Diagram for Hospital Mgmt

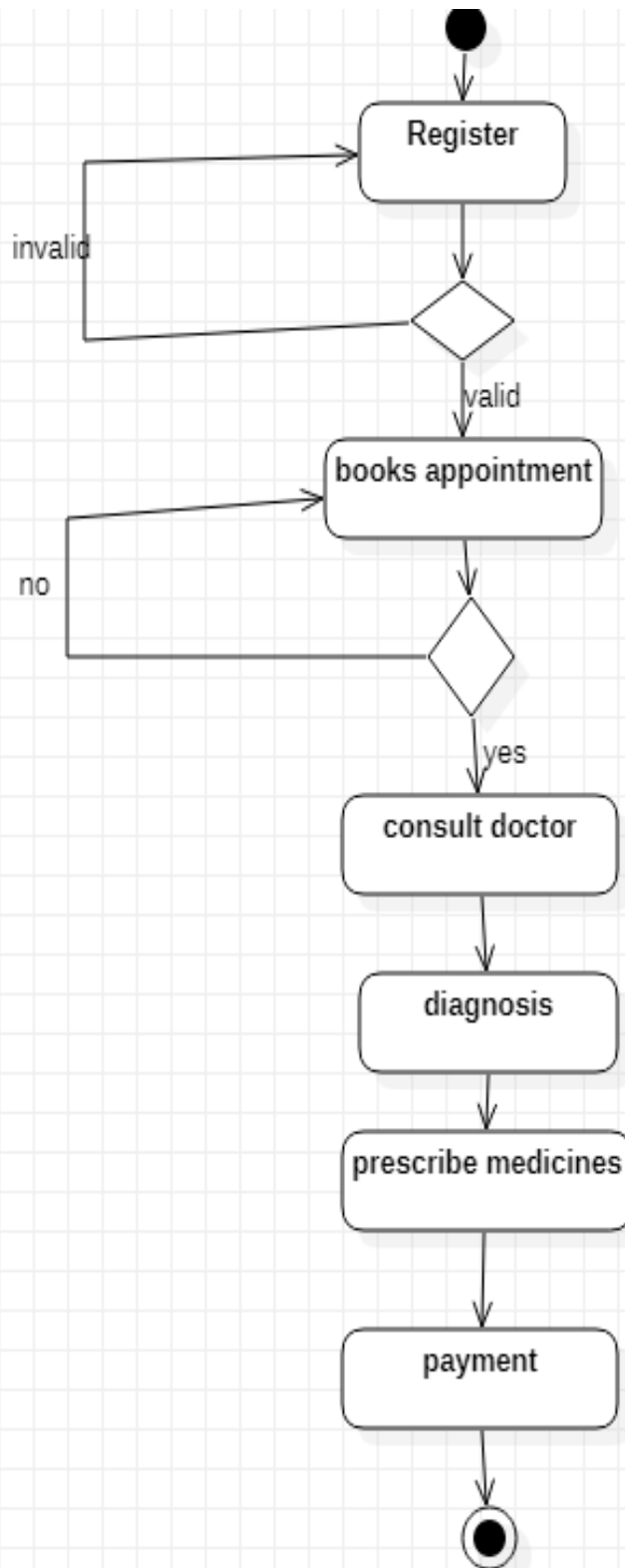


Collaboration Diagram For Hospital Mgmt.

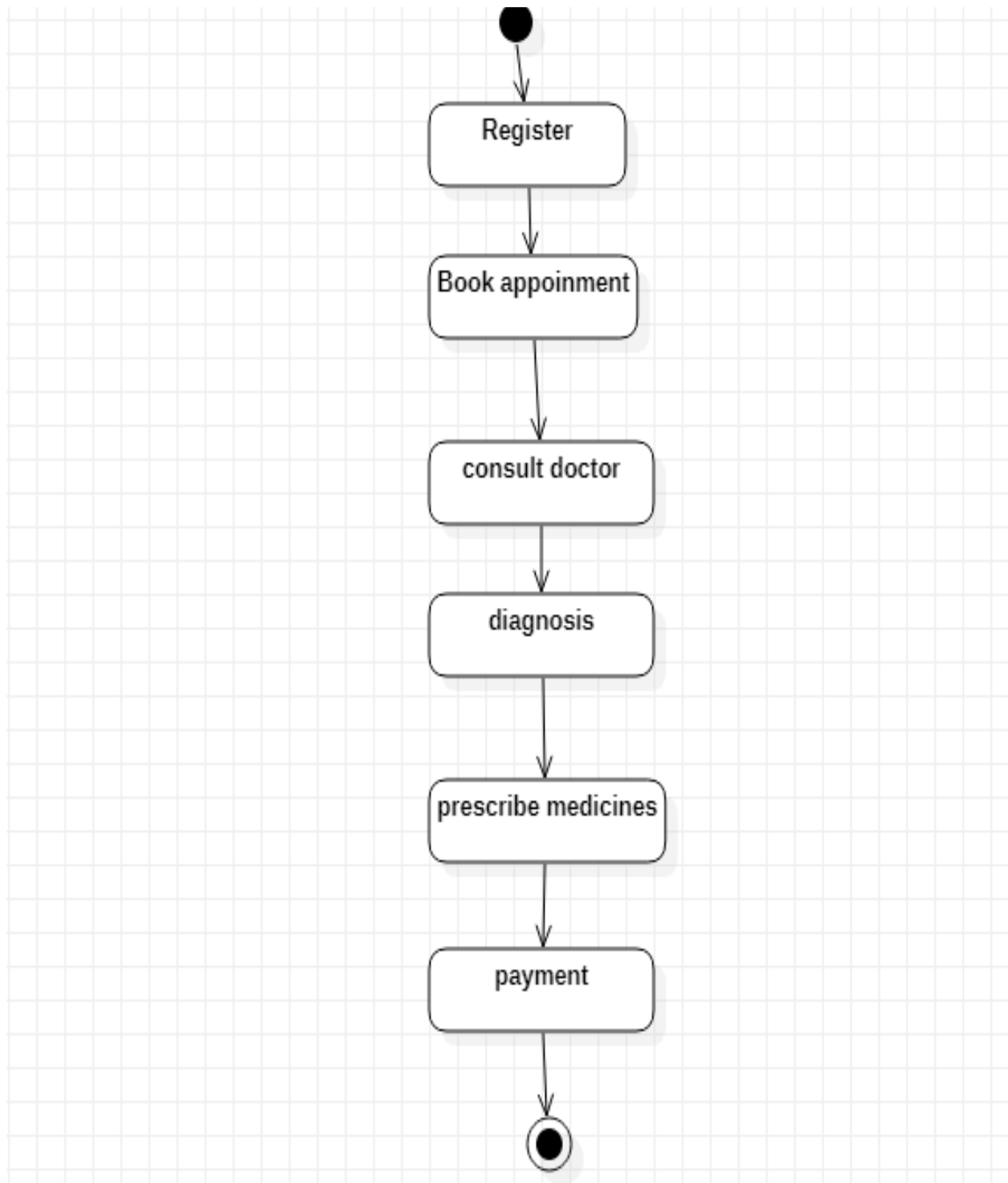
sd Hospital



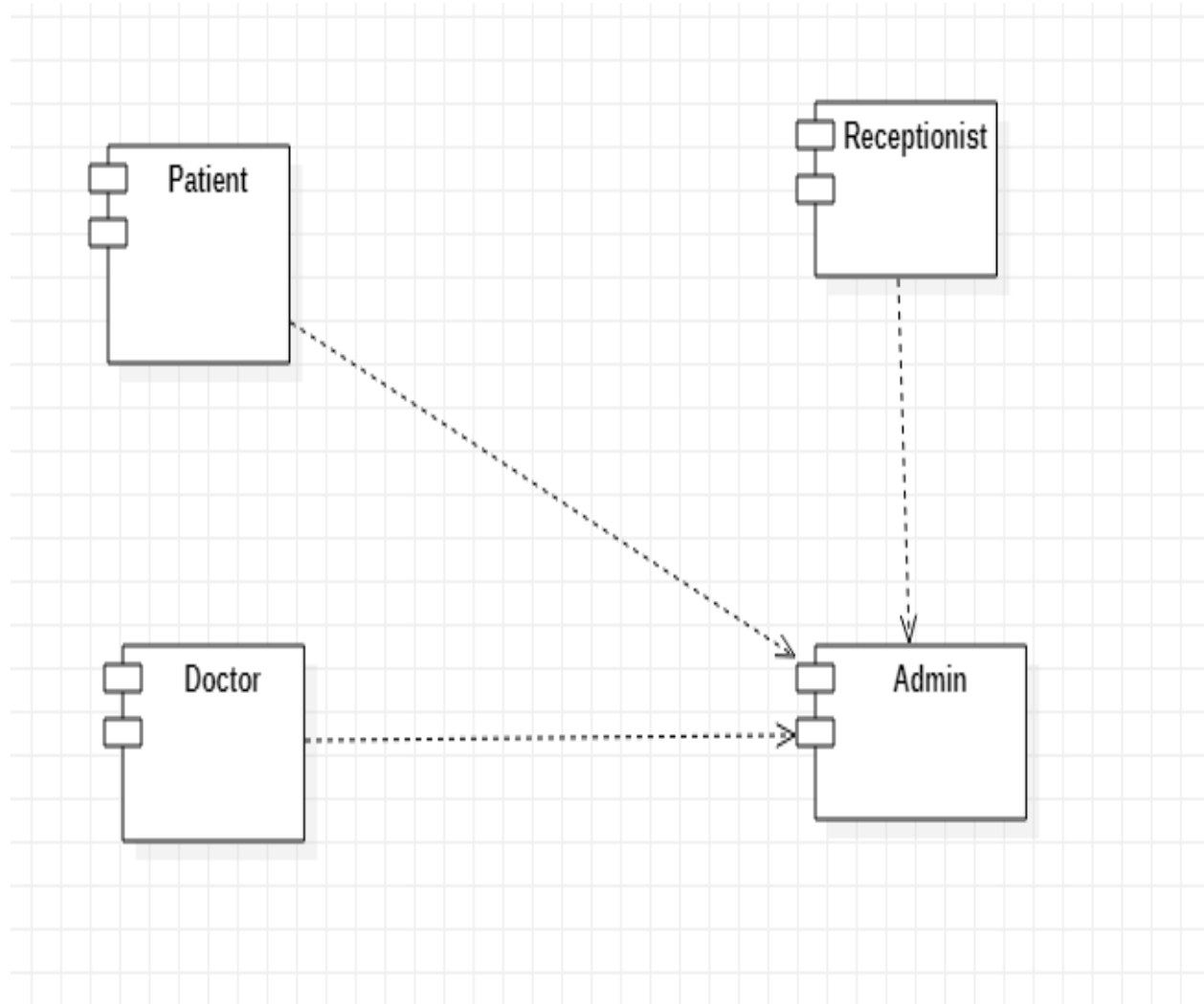
9.4. Activity Diagram For Hospital Mgmt



9.5. State Chart Diagram For Hospital Mgmt



9.6.Component Diagram For Hospital Mgmt

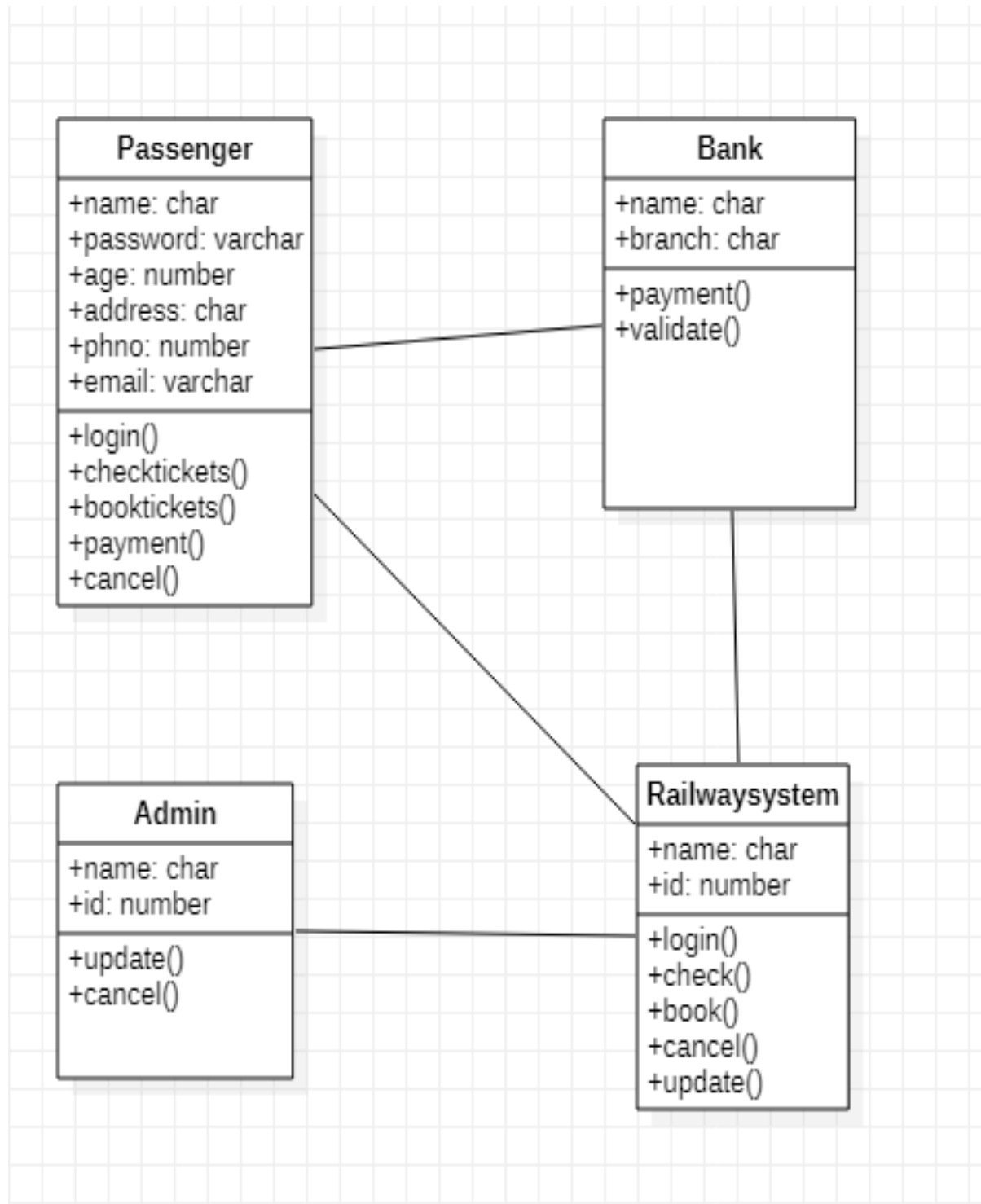


9.7. Deployment Diagram For Hospital Mgmt

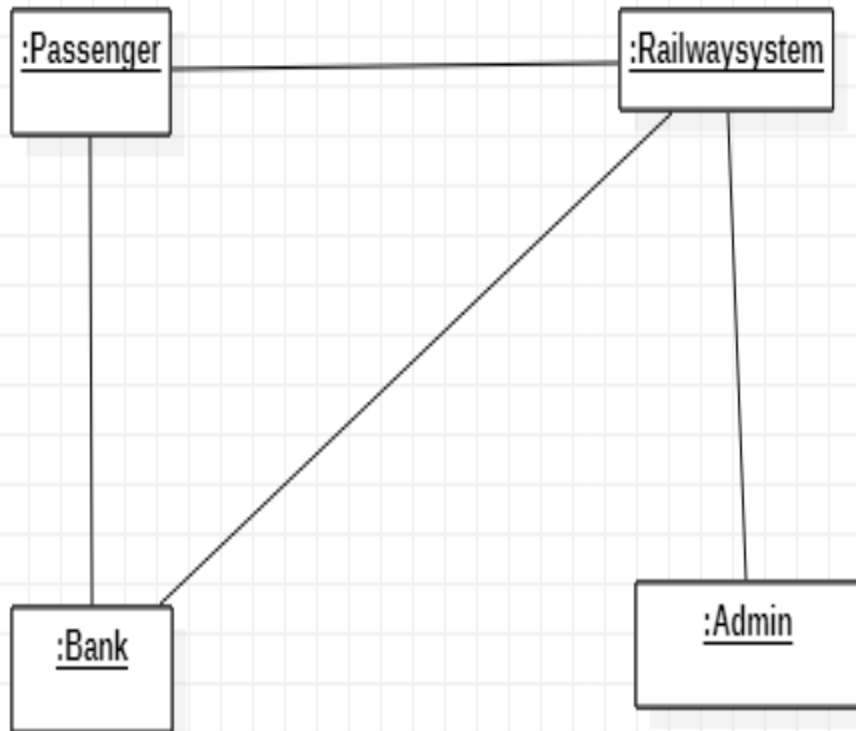


Task-10 Case study for Railway Reservation System.

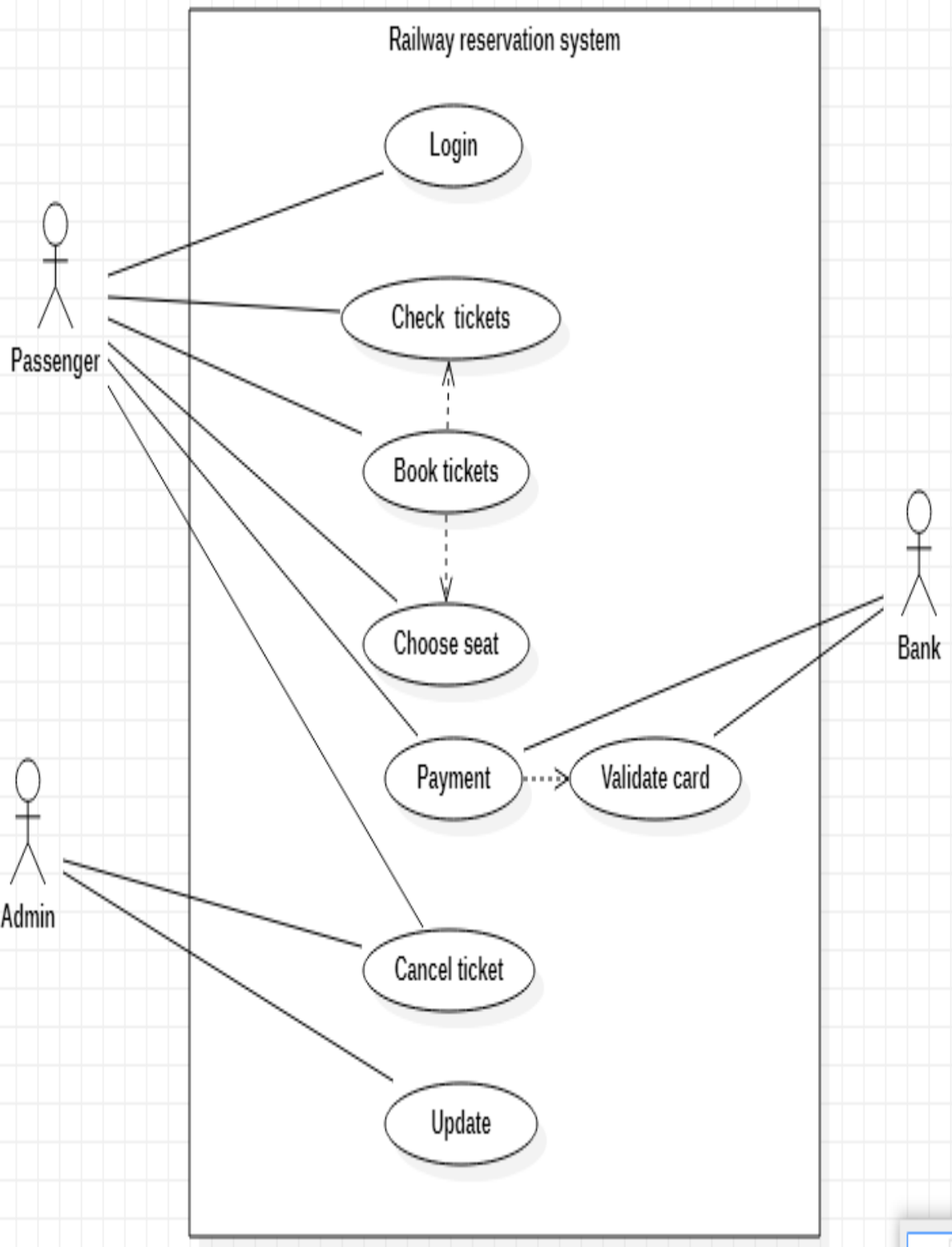
10.1 Class Diagram For Railway Reservation System



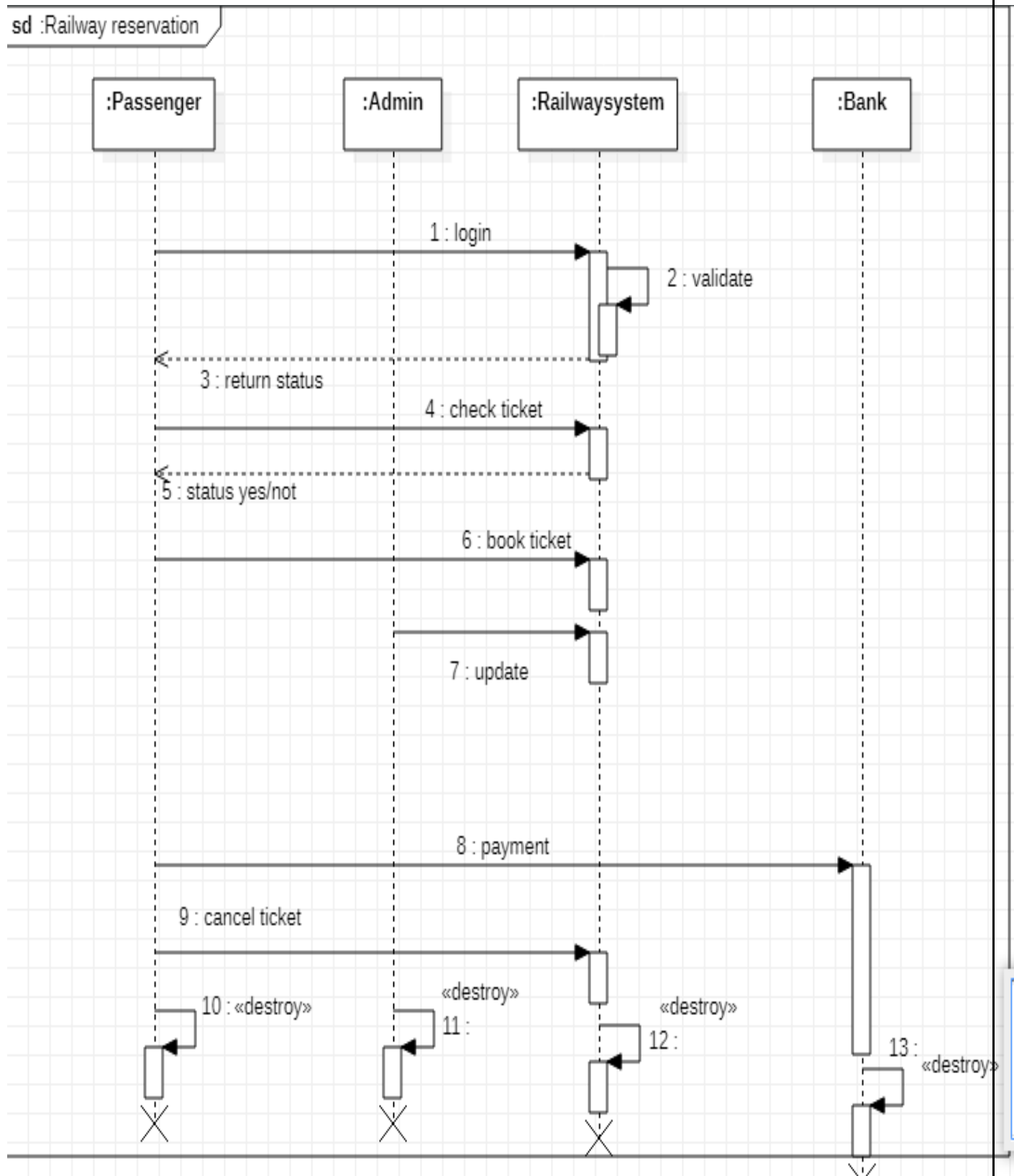
Object Diagram For Railway Reservation System



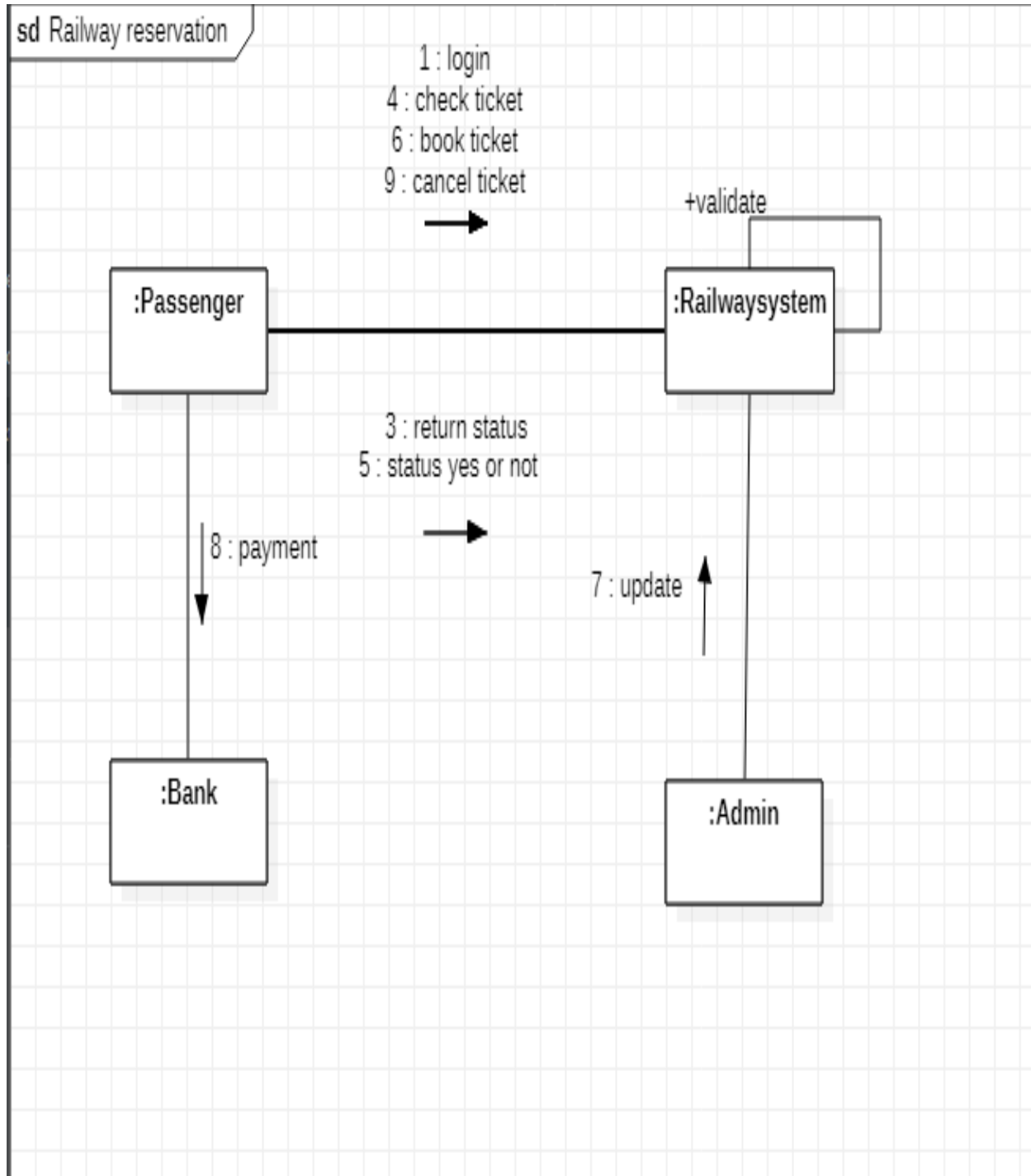
10.2 Usecase Diagram For Railway Reservation System



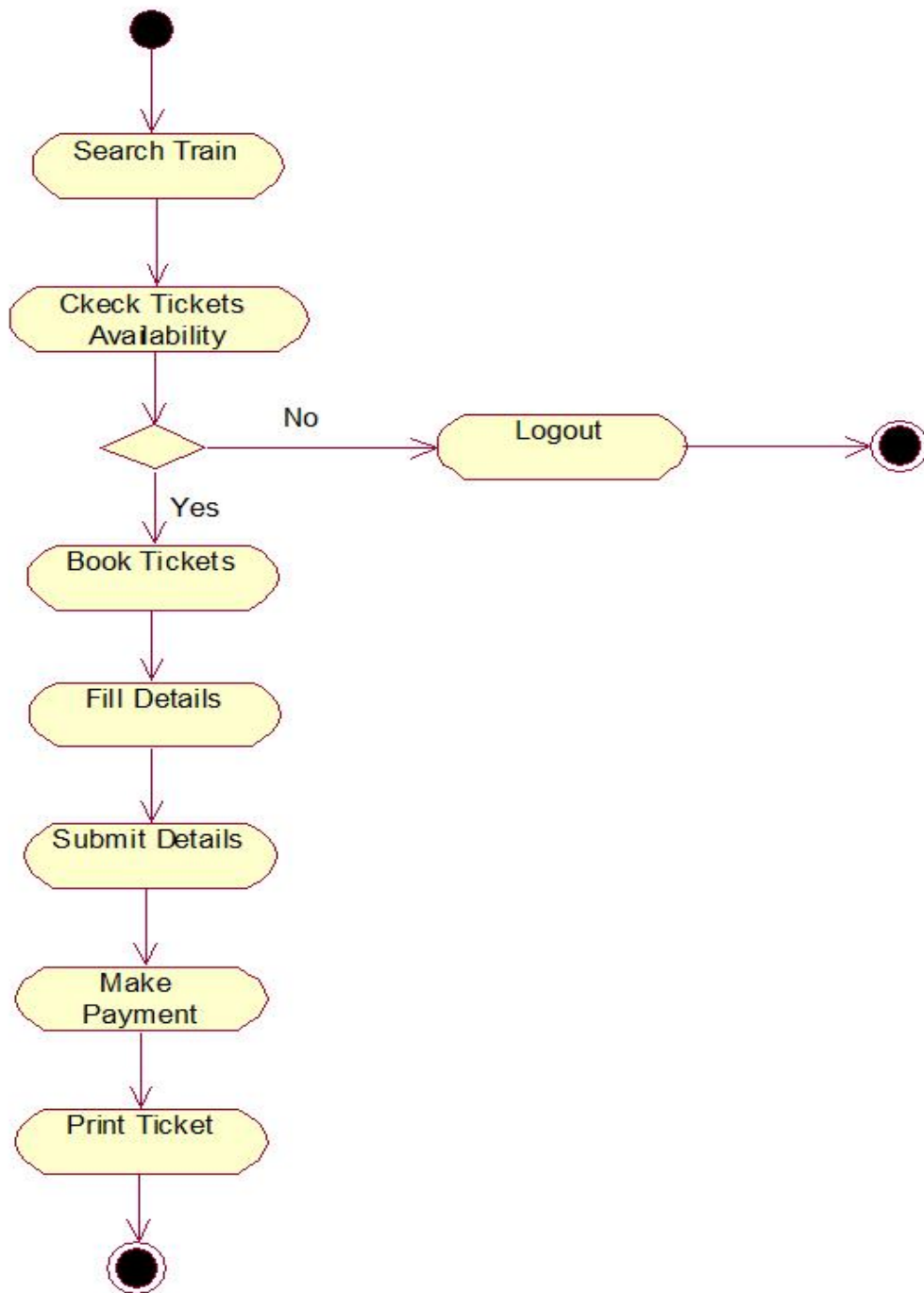
10.3 Sequence Diagram For Railway Reservation System



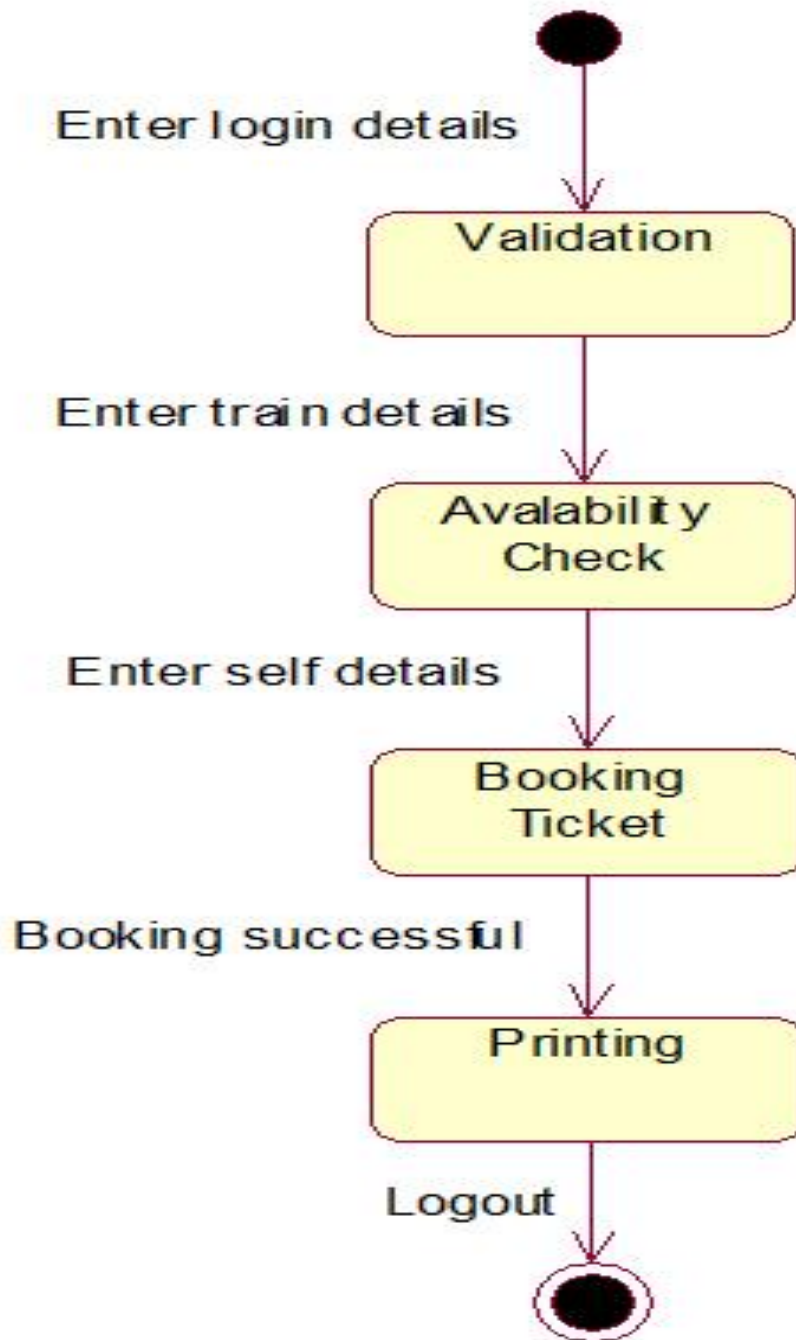
Collaboration Diagram For Railway Reservation System



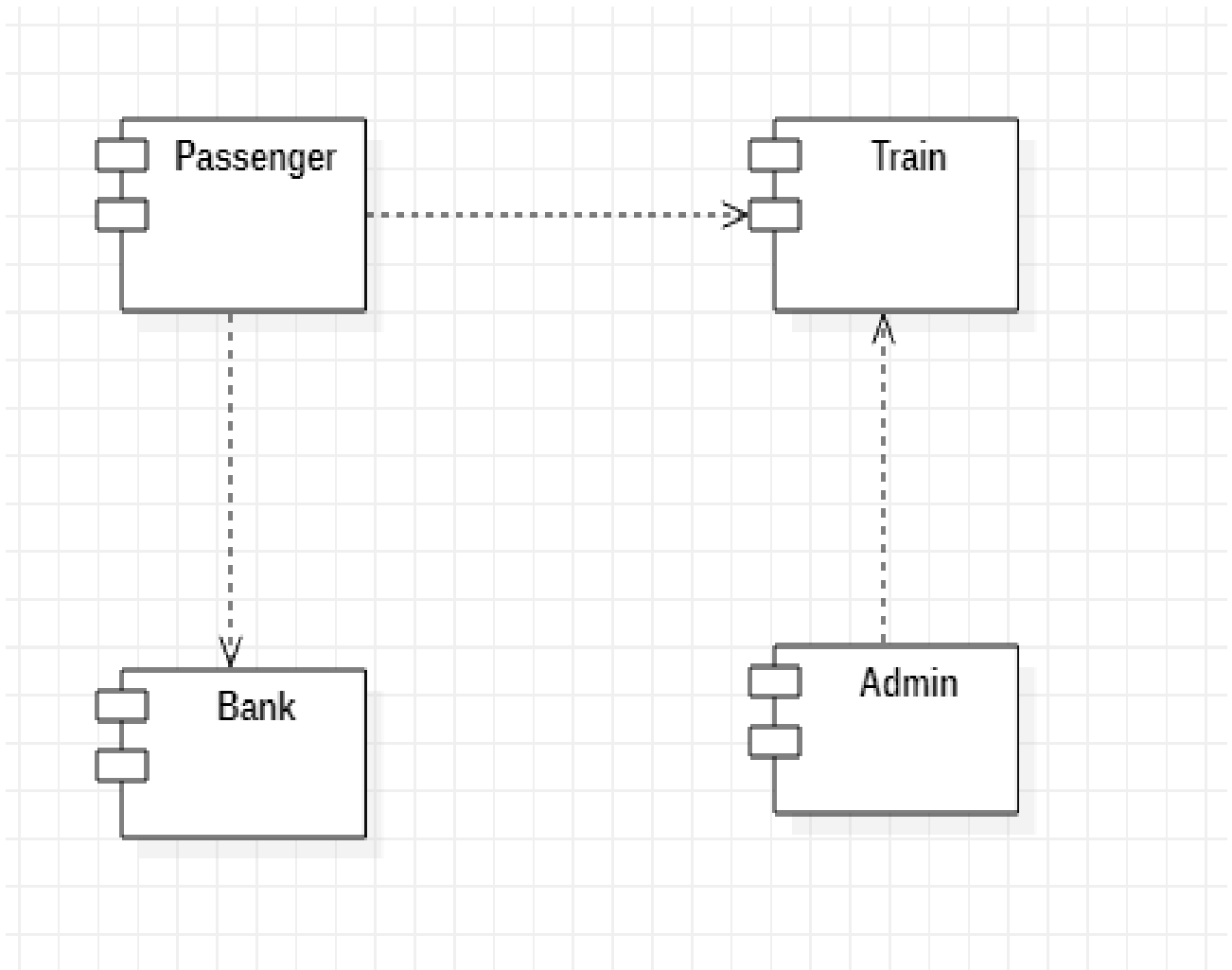
10.4 Activity Diagram For Railway Reservation System



10.5 State chart Diagram For Railway Reservation System



10.6 Component Diagram For Railway Reservation System



10.7 Deployment Diagram For Railway Reservation System

