

KIST COLLEGE OF MANAGEMENT

(Affiliated to Tribhuvan university)



A lab report on
UML diagrams

OBJECT ORIENTED ANALYSIS AND DESIGN

Submitted by:

Name:.....

Semester:.....

Course:.....

Symbol Number:.....

Submitted to:

Prawesh Dhungana

Class Diagram:

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

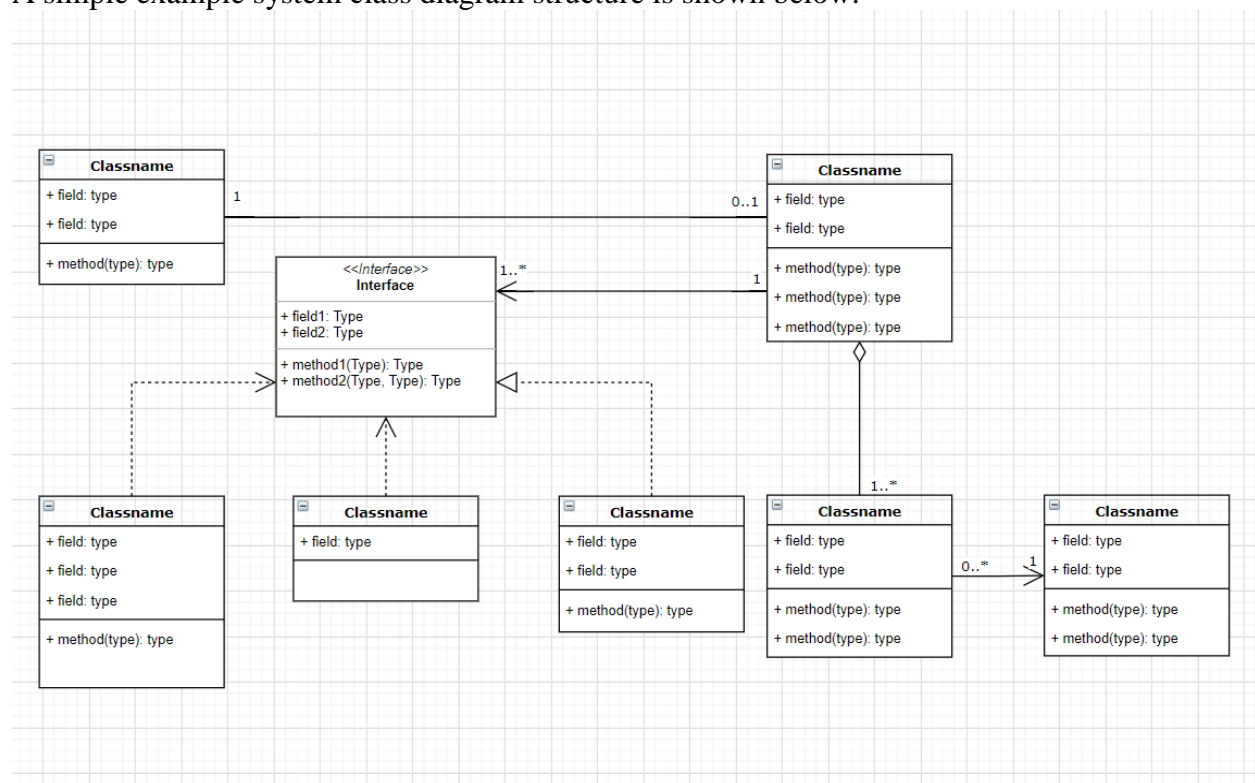
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

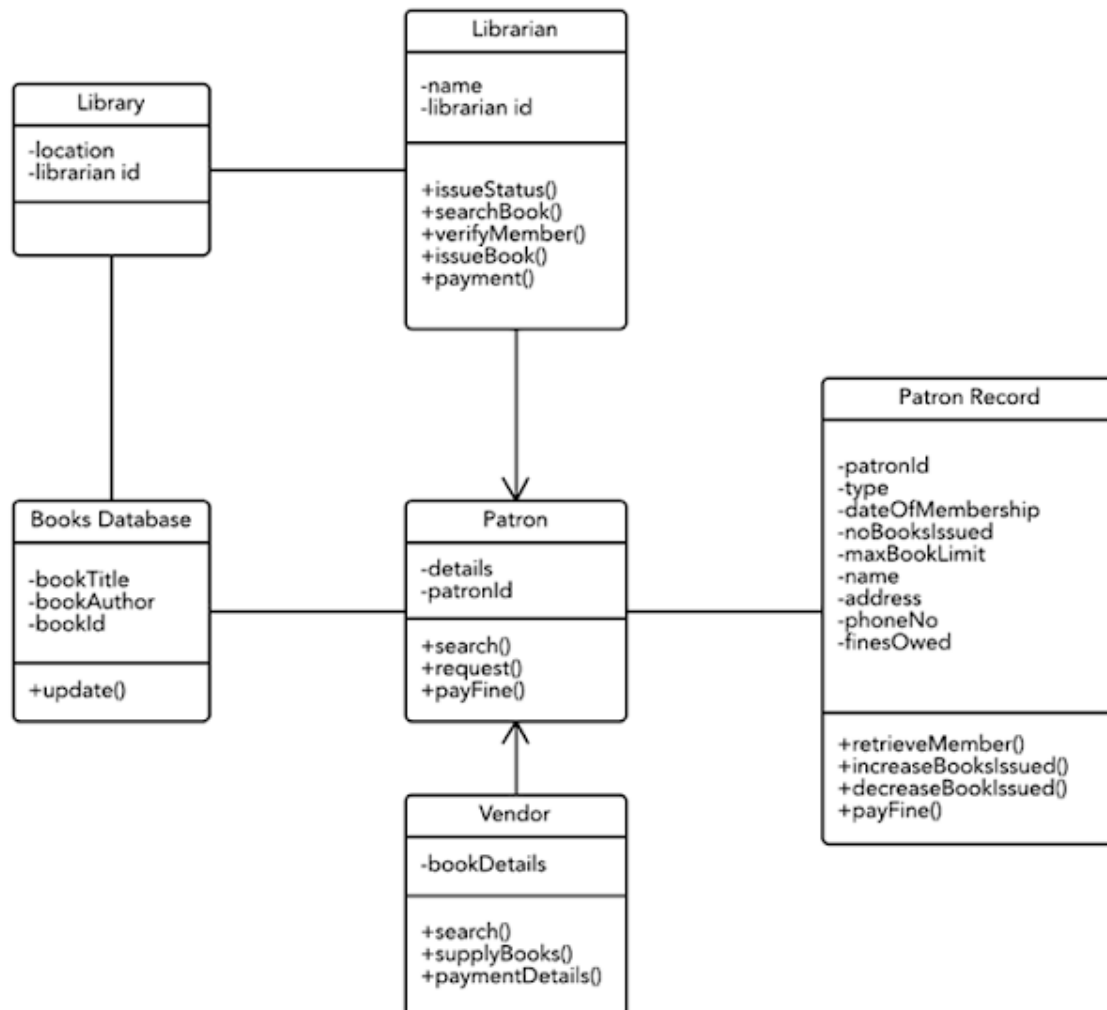
Eg1:

A simple example system class diagram structure is shown below:

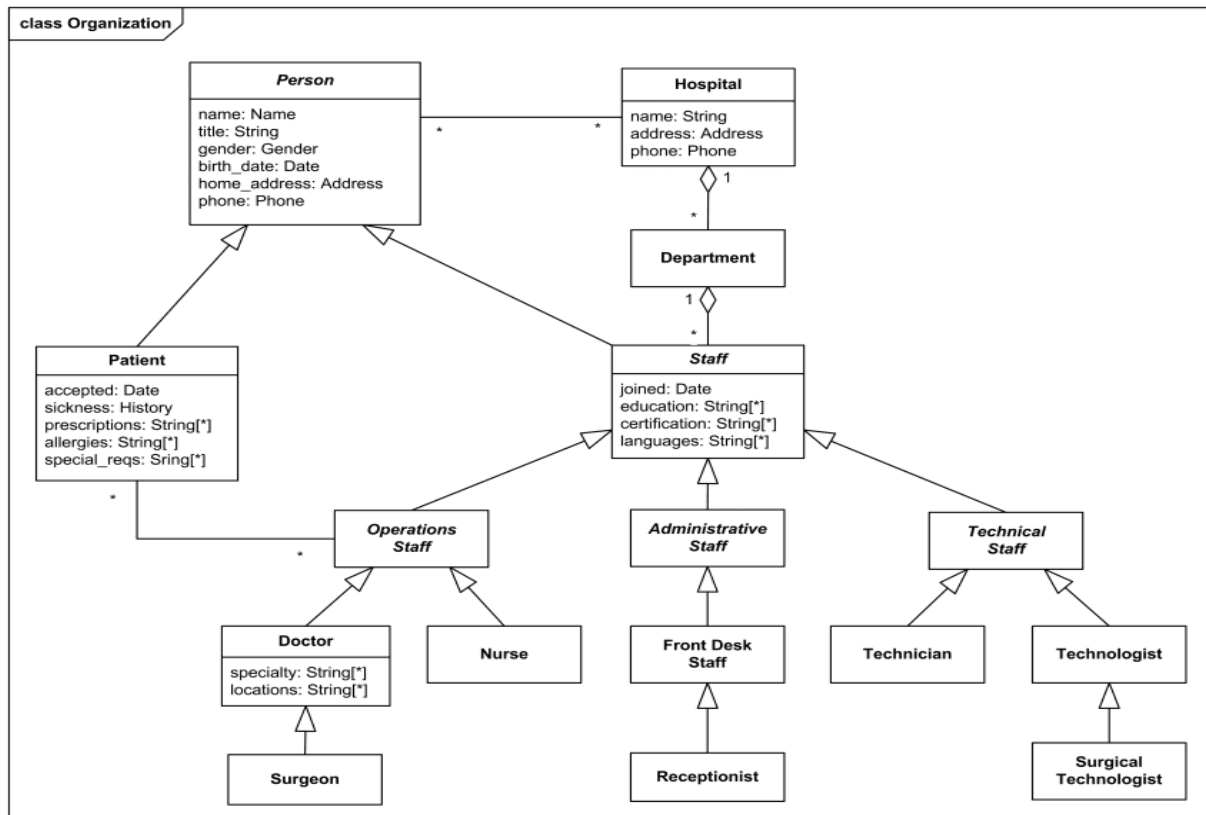


Eg2:

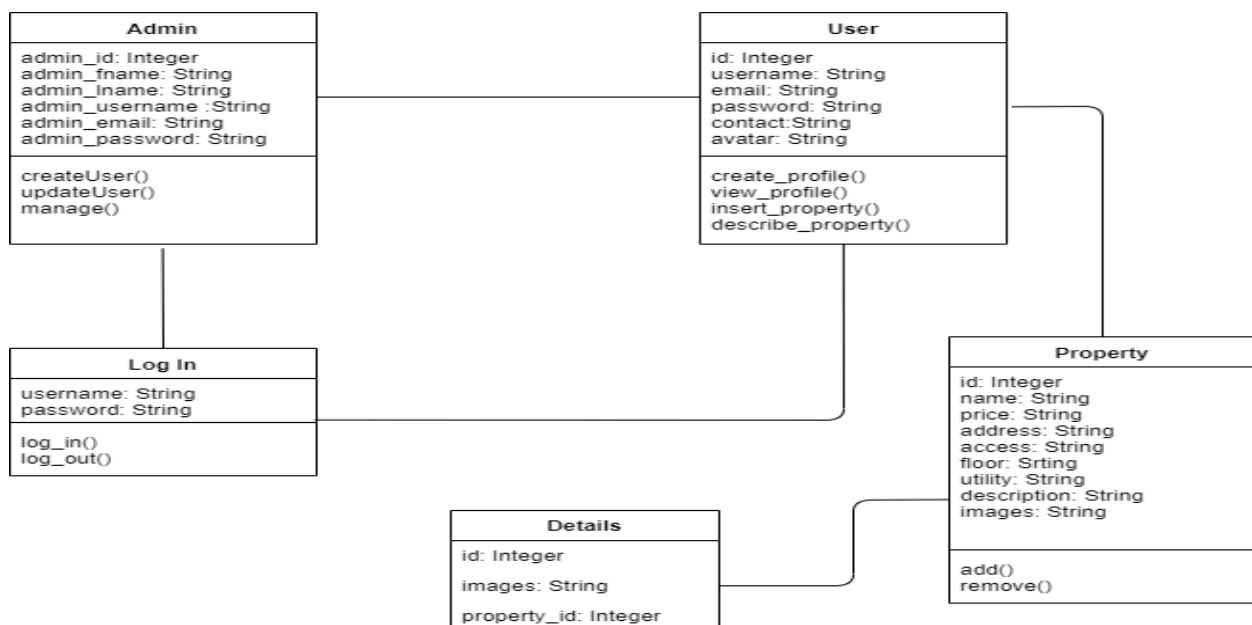
A Library Management System class diagram:



Eg3: A hospital management class diagram:



Eg3: A real estate management system class diagram:



Use Case Diagram

A use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. Furthermore, It depicts the functional requirements of system.

A use case diagram contains: use cases, relationships, boundary and actors.

UML use case diagrams are ideal for:

- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case

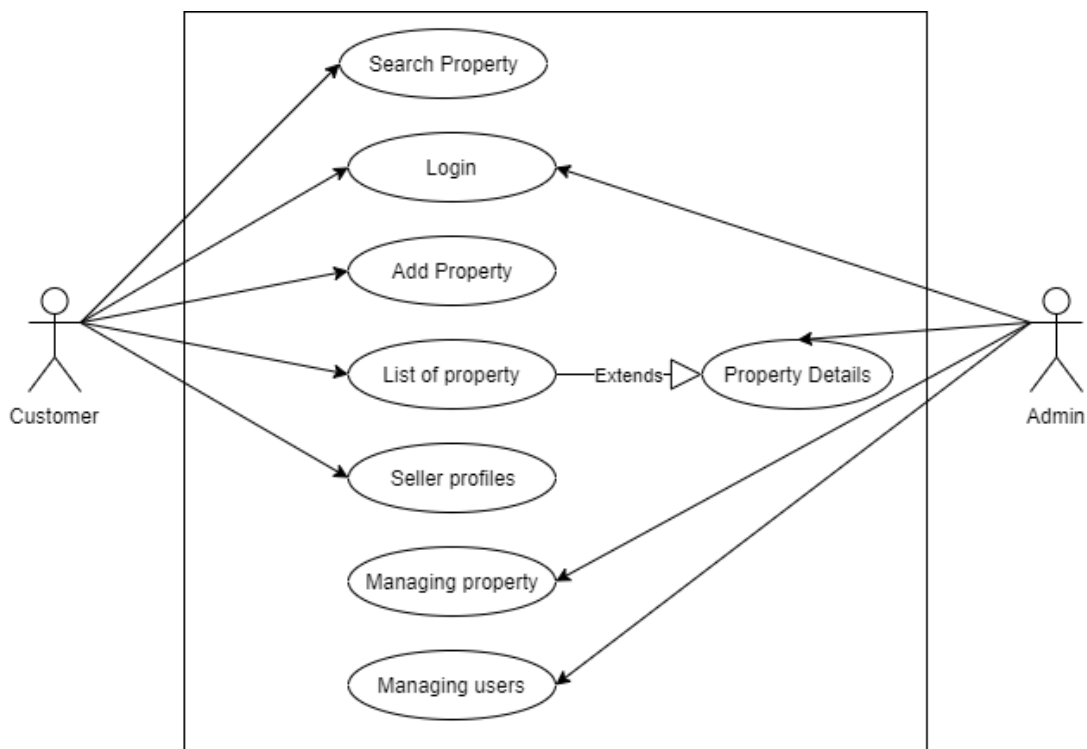
EG:

A use case diagram for Real Estate System.

Use cases: search property, list property, manage property, provide property details.

Actors: End customer(primary actor), Admin(secondary actor)

Relationship: Association

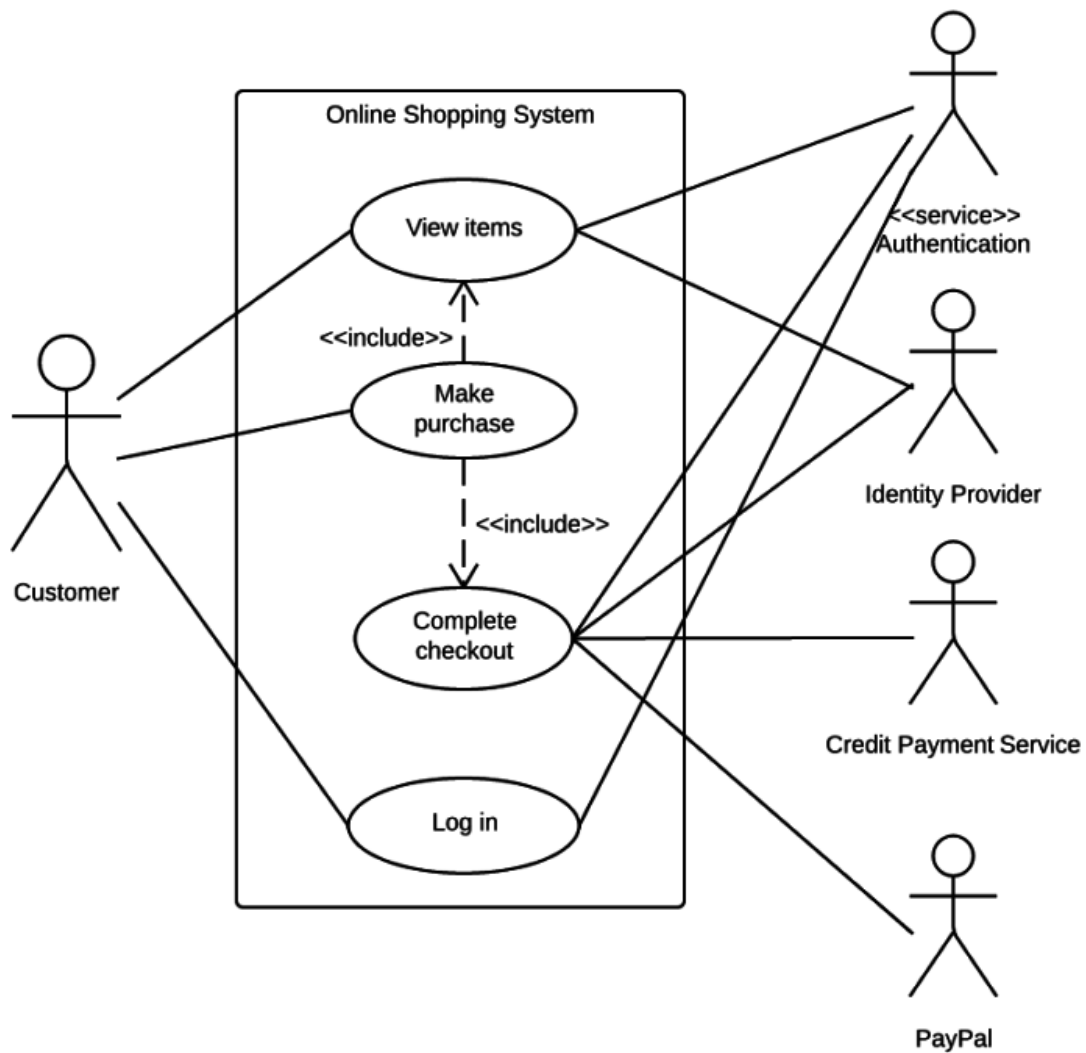


Eg:

A use case diagram on online shopping system

Use cases: student registration, manage examination, manage question bank, system login, test, reports.

Actors: Customer(Primary), Identity Provider, Credit Payment Service (secondary)



Sequence Diagrams

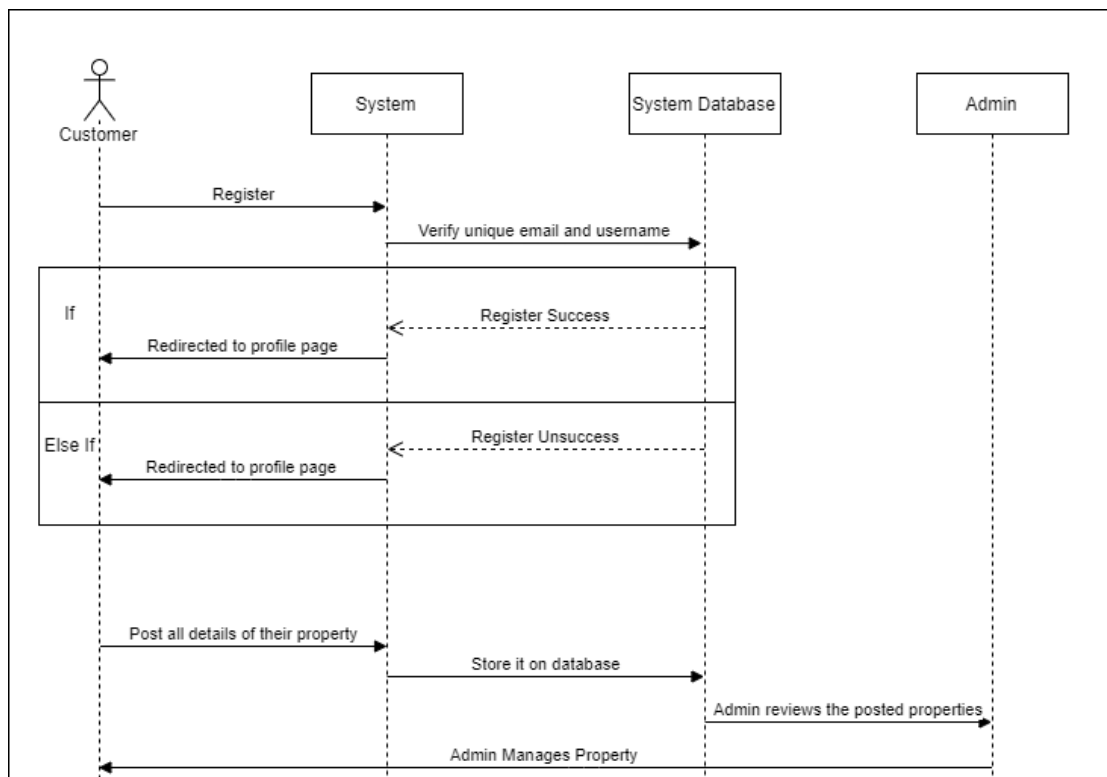
Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction)

Eg:

Sequence diagram for simple real estate website.



The diagram shows these lifelines:

- System
- Customer

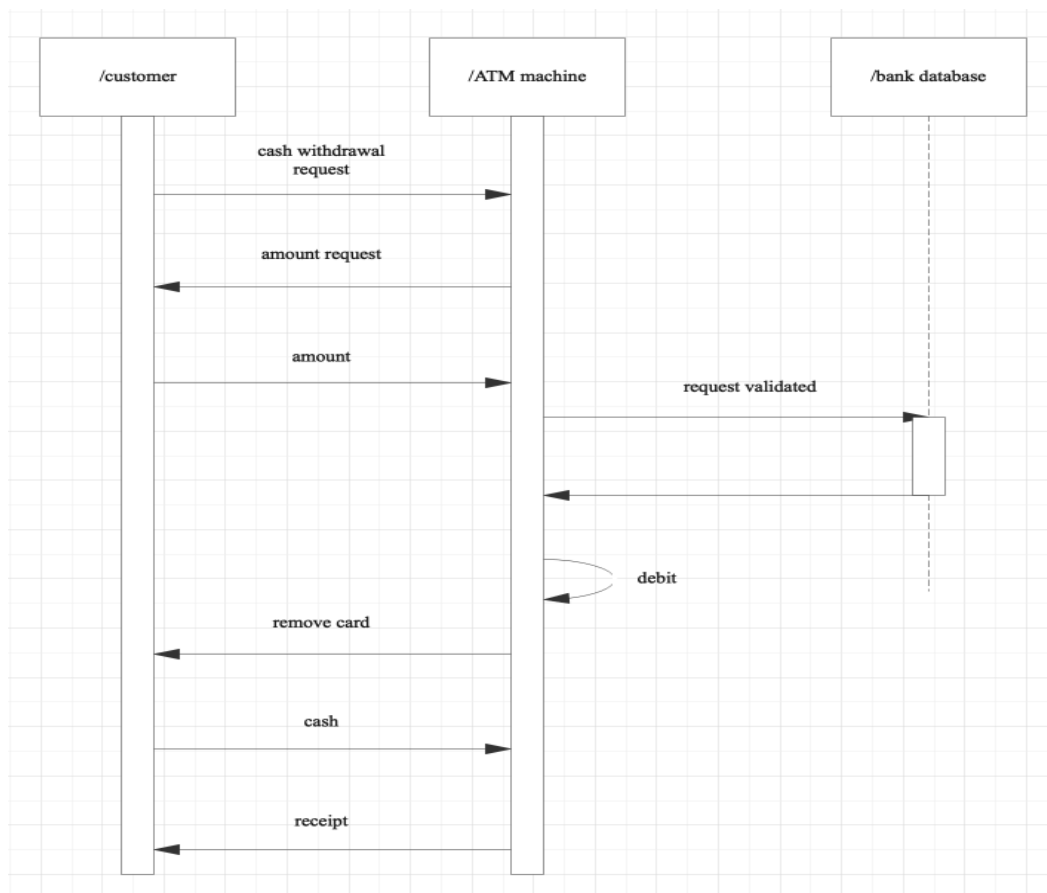
- System Database
- Admin

There are these interactions between the lifelines:

- Register
- Verify email and username
- Register success
- Redirect to profile page
- Register unsuccess
- Redirect to profile page
- Post all details of their property
- Store it on database
- Admin reviews the posted properties
- Admin manages the property

Eg 2:

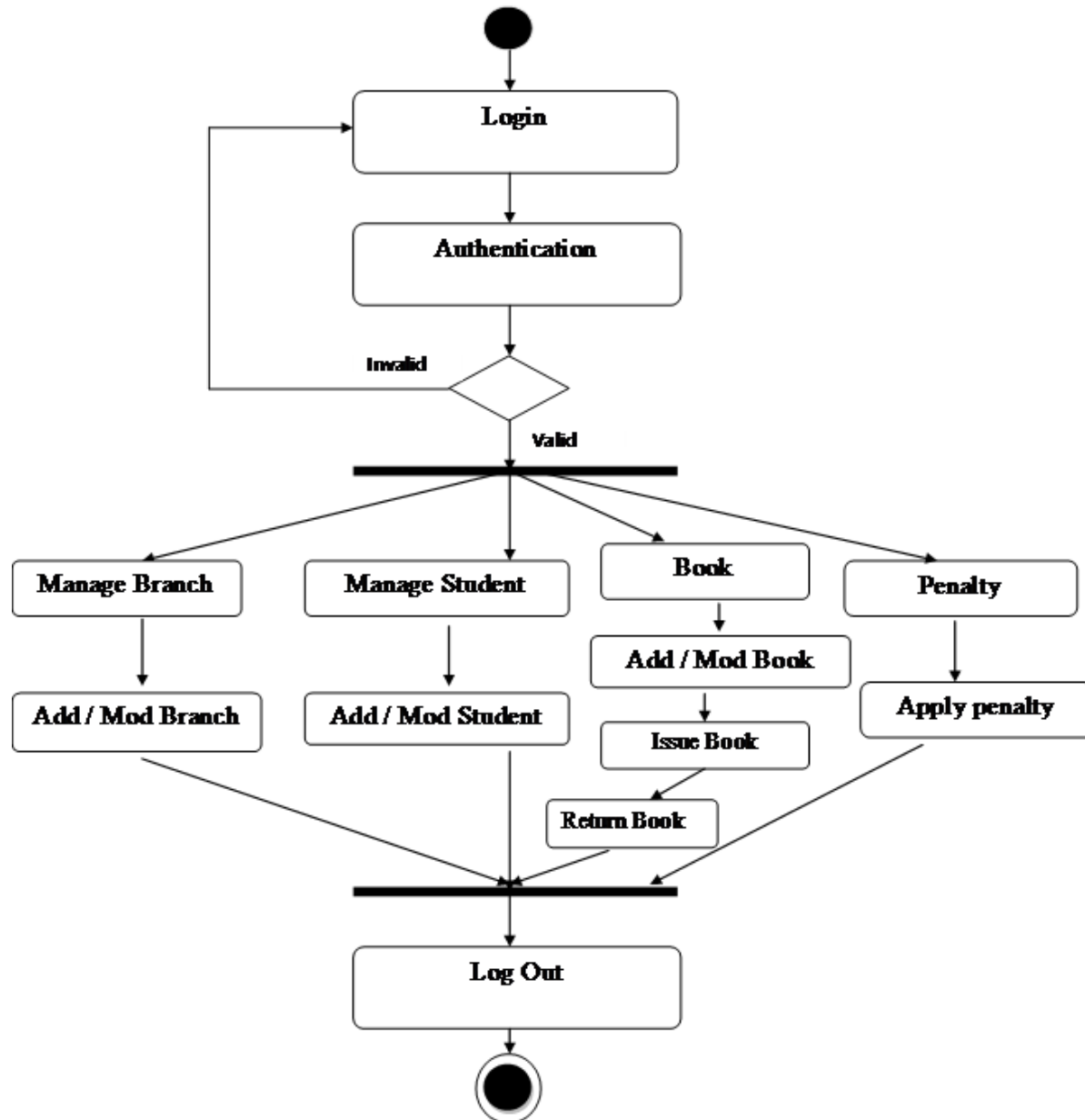
ATM Machine



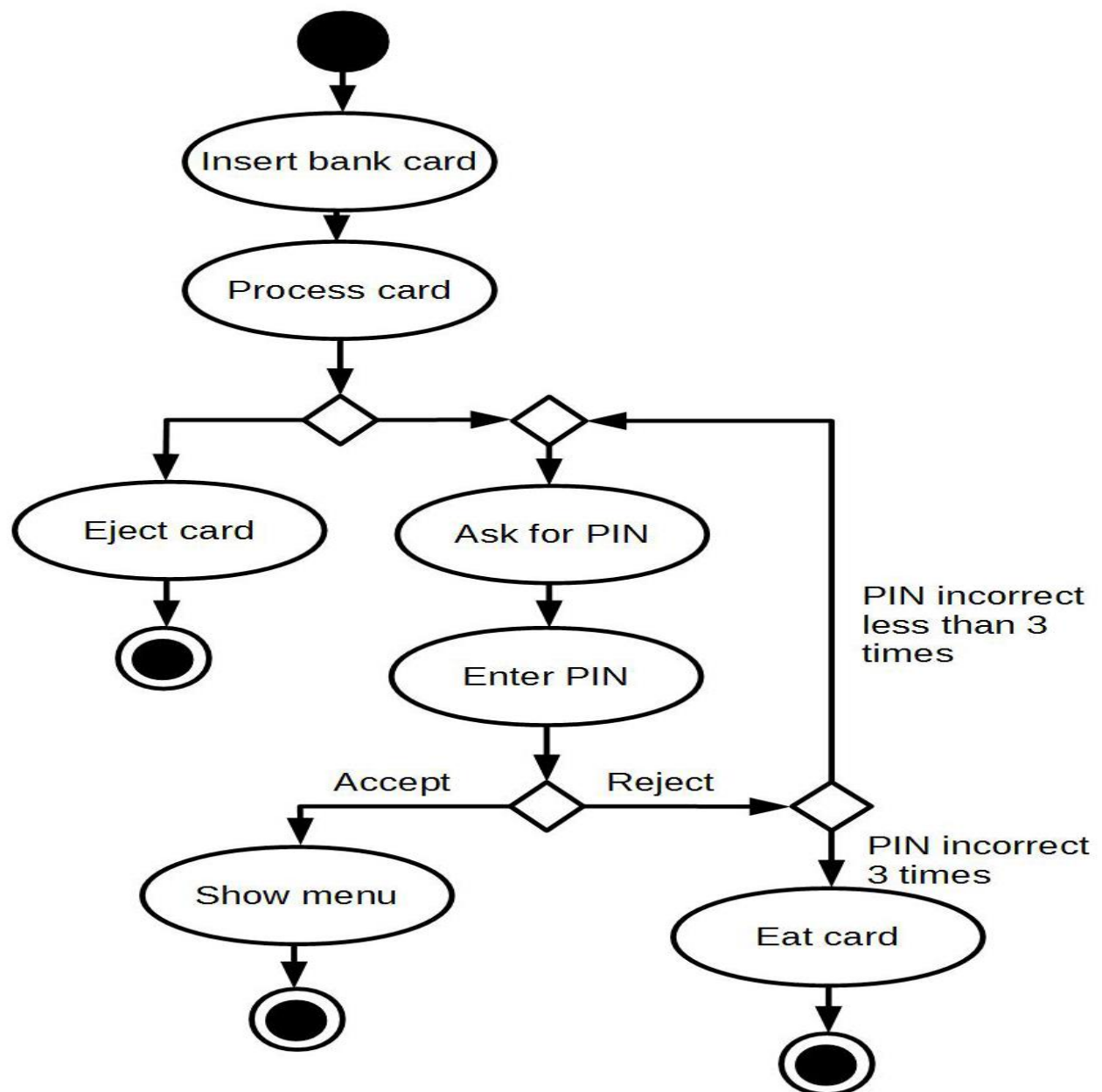
Activity Diagram

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

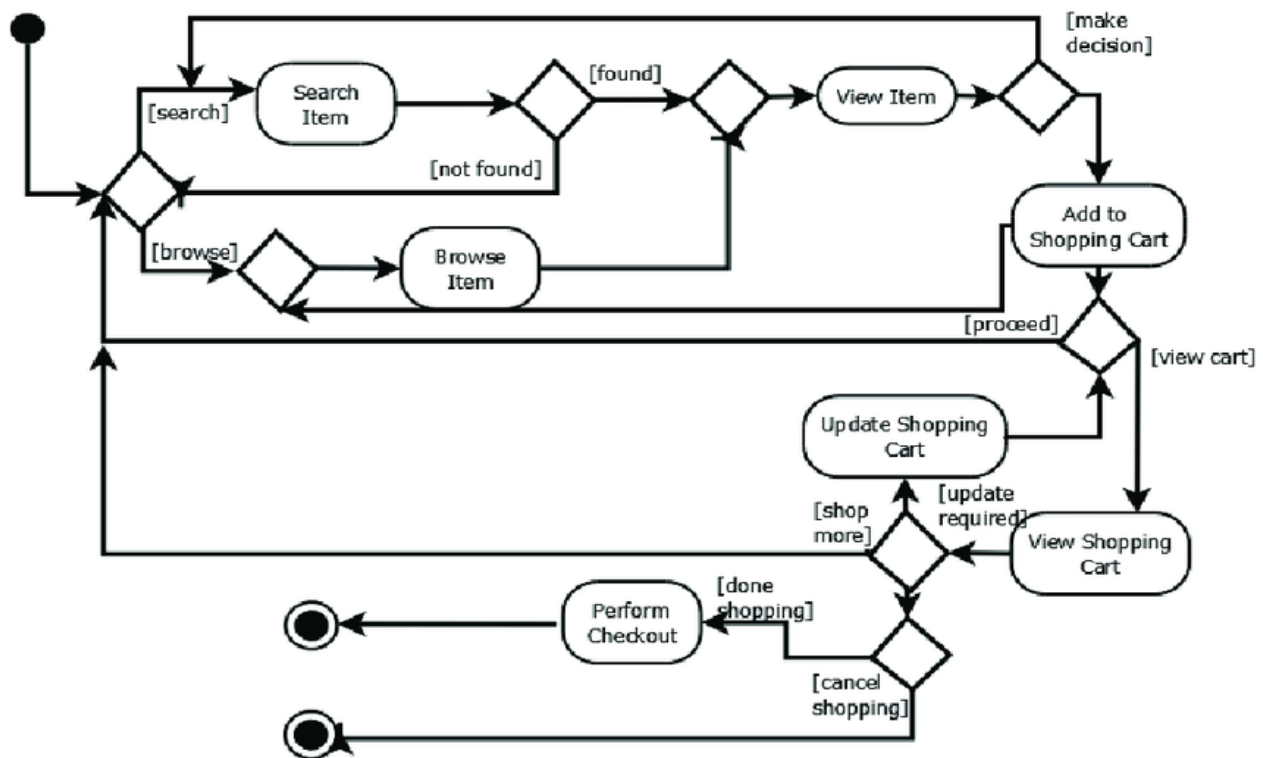
Eg 1:Activity Diagram for library management system.



Eg 2: Activity diagram for ATM machine system



Eg 3: Activity diagram customer doing online shopping



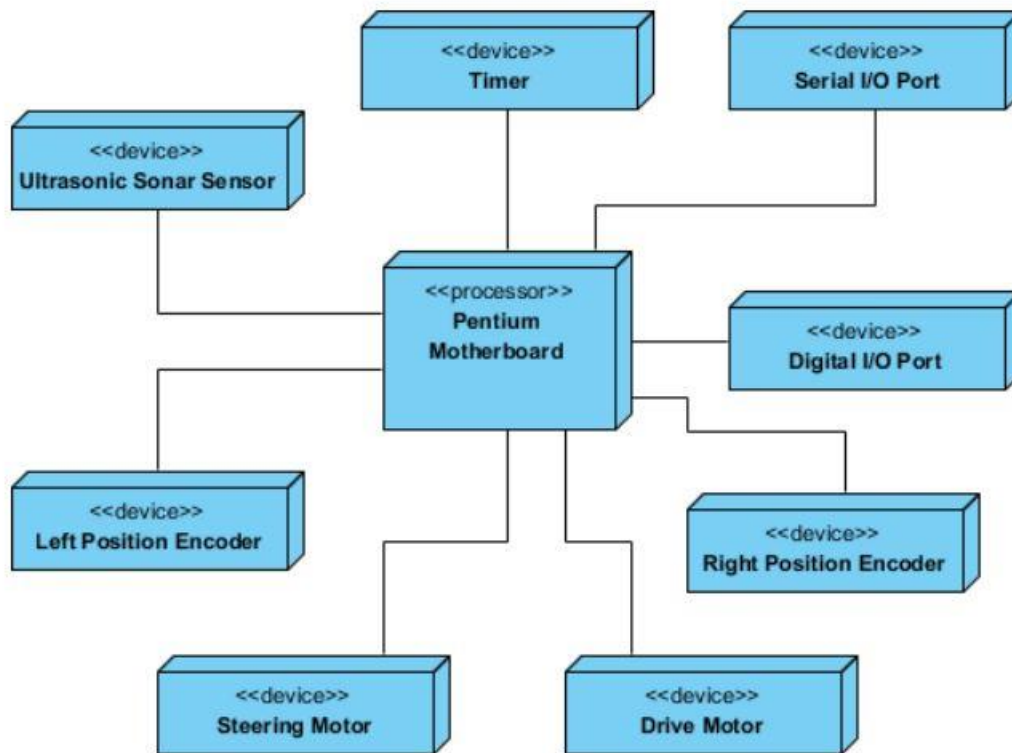
Deployment diagram

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

Purpose of Deployment Diagrams

- They show the structure of the run-time system
- They capture the hardware that will be used to implement the system and the links between different items of hardware.
- They model physical hardware elements and the communication paths between them
- They can be used to plan the architecture of a system.
- They are also useful for Document the deployment of software components or nodes.

Eg: Deployment diagram of embedded system

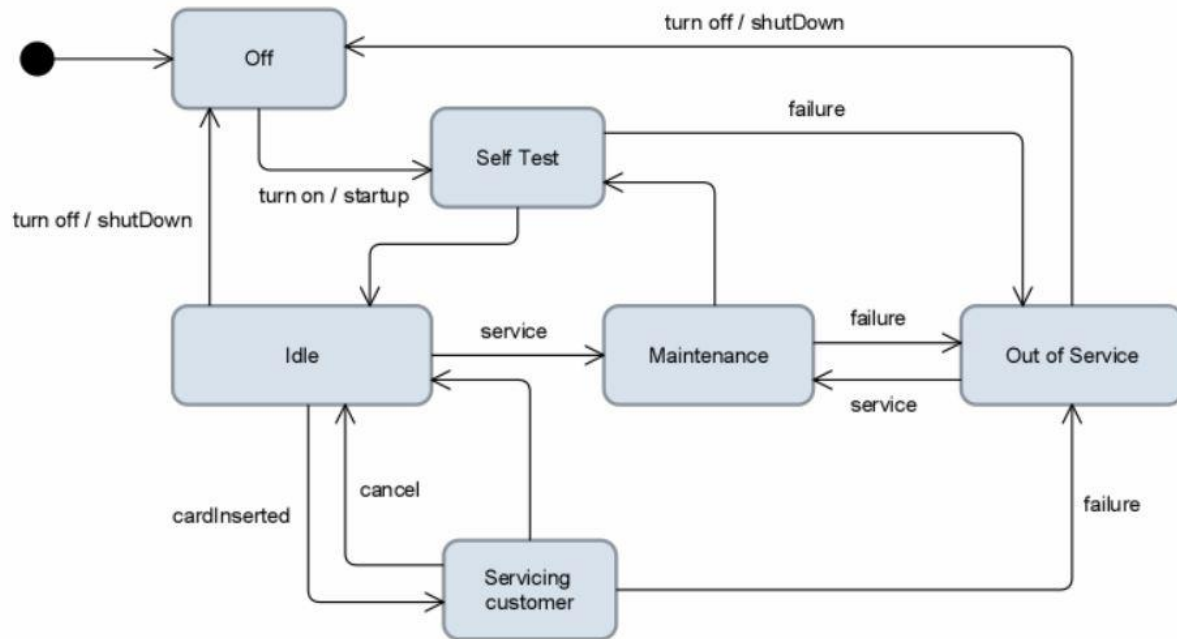


State Diagrams

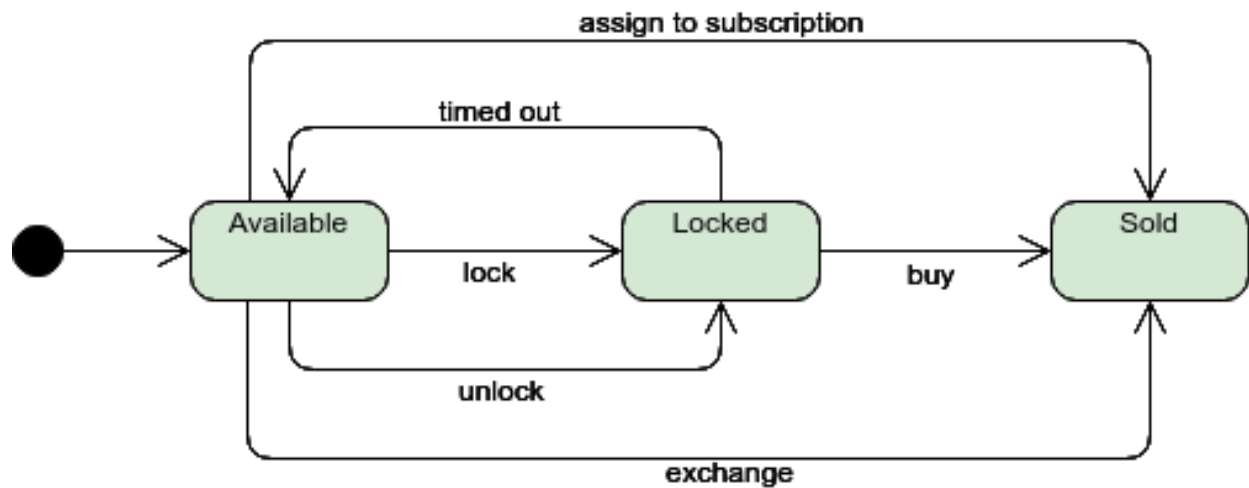
A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams.

The basic purpose of a state diagram is to portray various changes in state of the class and not the processes or commands causing the changes. However, a flowchart on the other hand portrays the processes or commands that on execution change the state of class or an object of the class.

Eg: State diagram of ATM



Eg2: State diagram of ticket selling system



Eg 3: System State Diagram for user authentication process

