

Assignment1

(To be submitted on or before 6/4/2012)

Q1) Briefly describe the following fundamental concepts of object oriented world.

- Objects: Objects represent an entity and the basic building block.
- Class: Class is the blue print of an object.
- Abstraction: Abstraction represents the behavior of a real world entity.
- Encapsulation: Encapsulation is the mechanism of binding the data together and hiding them from outside world.
- Inheritance: Inheritance is the mechanism of making new classes from existing one.
- Polymorphism: It defines the mechanism to existing one in different forms.

Q2) List the differences between Object Oriented and Traditional Analysis & Design Techniques.

Traditional Approach	Object oriented approach
Either data-centric or process- centric	Views system as a collection of self contained objects, including both data and process
Views a system as a top-down approach to system development	Views a system as a bottom-up approach to system development
Adopts step by step approach to SDLC phases	No clear cut step form one phase to another
Specific documentations are produced at the end of each phase	There is no specific documentation after each phase.
Data needed for designing and development of the system is in a single data dictionary	No single repository for data
Input and output of the system are identified from the DFD	Input and output are scattered in many sequence diagram

Q3) List the principles of Object Oriented Software Development Process

S	The Single Responsibility Principle	Every object should have a single responsibility, and that responsibility should be entirely encapsulated by the class.
O	The Open Closed Principle	Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification.
L	The Liskov Substitution Principle	Derived class objects must be substitutable for their base class objects.
I	The Interface Segregation Principle	Make fine grained interfaces that are client specific. (Many client specific interfaces are better than one general purpose interface)
D	The Dependency Inversion Principle	Depend on abstractions, not on concretions. That is, A. High level modules should not depend upon low level modules. Both should depend upon abstractions. B. Abstractions should not depend upon details. Details should depend upon abstractions.

Q4) Why we need Models?

- Models help us to visualize a system as it is or as we want it to be.
- Models permit us to specify the structure or behavior of a system.
- Models give us a template that guides us in constructing a system.
- Models document the decisions we have made.

Q5) List the principles of Object Oriented Modeling

- The model should illuminate the most complicated development problems, offering very good insight into the problem.
- Every model may be expressed at different levels of precisions.
- The best models are connected to reality.
- No single model is sufficient. Every system is best approached through a small set of independent models.

Q6) What is UML?

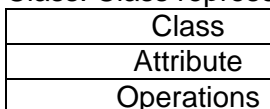
UML (stands for Unified Modeling Language) is a standard modeling language for specifying, visualizing, constructing, and documenting the artifacts of systems.
 UML is different from the other common programming languages like C++, Java, COBOL etc.
 UML is a pictorial language used to make system blue prints.

Q7) What are the building blocks of UML :

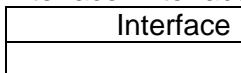
- **Things:** Things are the most important building blocks of UML. Things can be: Structural, Behavioral, Grouping, and Annotational.
 The structural things define the static part of the model. They represent physical and conceptual elements.
 The behavioral thing consists of the dynamic parts of UML models.
 Grouping things can be defined as a mechanism to group elements of a UML model together.
 Annotational things can be defined as a mechanism to capture remarks, descriptions, and comments of UML model elements.
- **Relationships:** It shows how elements are associated with each other and this association describes the functionality of an application.
- **Diagrams:** UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system.

Q8) List the Things we use in UML.**a) Structural things:**

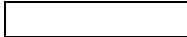
- i) **Class:** Class represents set of objects having similar responsibilities.



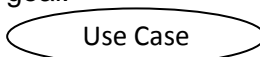
- ii) **Interface:** Interface defines a set of operations which specify the responsibility of a class.



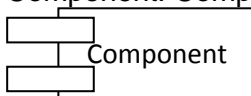
- iii) **Collaboration:** Collaboration defines interaction between elements.



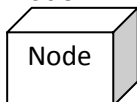
- iv) **Use case:** Use case represents a set of actions performed by a system for a specific goal.



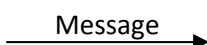
- v) **Component:** Component describes physical part of a system.



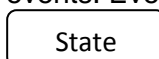
- vi) **Node:** A node can be defined as a physical element that exists at run time.

**b) Behavioral things:**

- i) **Interaction:** Interaction is defined as a behavior that consists of a group of messages exchanged among elements to accomplish a specific task.

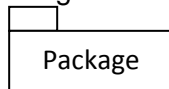


- ii) **State machine:** State machine is useful when the state of an object in its life cycle is important. It defines the sequence of states an object goes through in response to events. Events are external factors responsible for state change.



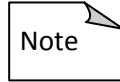
c) Grouping things:

- i) Package is the only one grouping thing available for gathering structural and behavioral things.

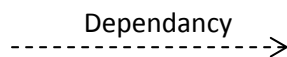


d) Annotational things:

- i) Note is the only one Annotational thing available.

**Q9) List the four kinds of relationships we use in UML.**

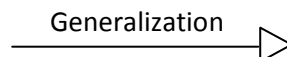
1. Dependency: Dependency is a relationship between two things in which change in one element also affects the other one.



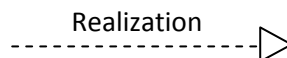
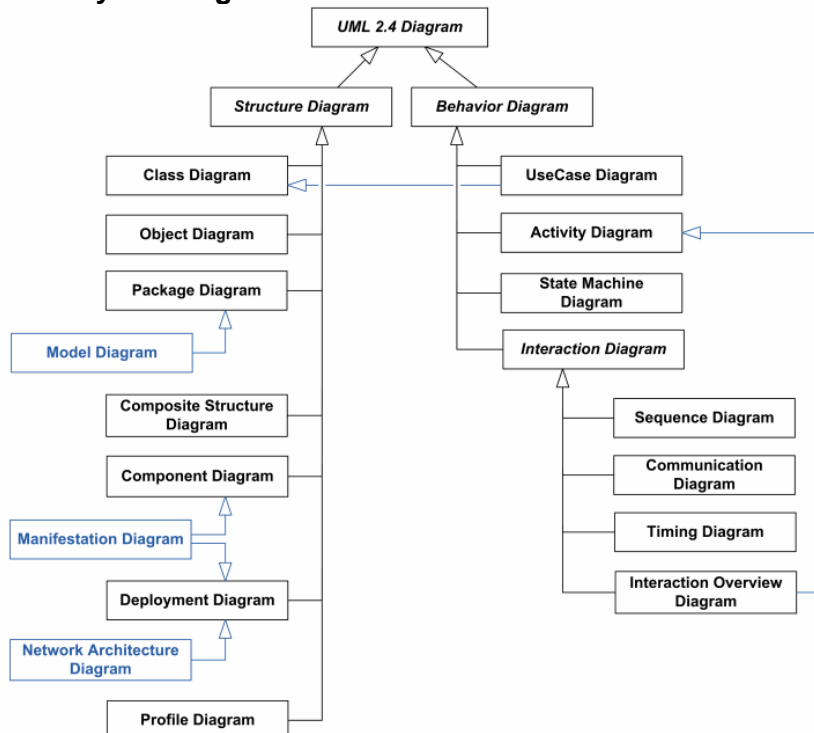
2. Association: Association is basically a set of links that connects elements of an UML model. It also describes how many objects are taking part in that relationship.



3. Generalization: Generalization can be defined as a relationship which connects a specialized element with a generalized element. It basically describes inheritance relationship in the world of objects.



4. Realization: Realization can be defined as a relationship in which two elements are connected. One element describes some responsibility which is not implemented and the other one implements them. This relationship exists in case of interfaces.

**Q10) Classify the diagrams included in UML.**

Note: Structural diagrams consist of static diagrams. Behavioral diagrams consist of dynamic diagrams.

Q11) Briefly describe the following diagrams.

1. Class diagrams: It describes the objects in a system and their relationships. Class diagram consists of attributes and functions. Class diagrams are the only UML diagrams which can be mapped directly with object oriented languages. So it is widely used by the developer community.
2. Use case diagrams: It consists of use cases, actors and their relationships. Use case diagram is used at a high level design to capture the requirements of a system. So it represents the system functionalities and their flow.
3. Objects diagrams: An object diagram is an instance of a class diagram. So the basic elements are similar to a class diagram. Object diagrams are consists of objects and links. It captures the instance of the system at a particular moment. Object diagrams are used for prototyping, reverse engineering and modeling practical scenarios.
4. Composite structure diagram: It is a set of interconnected elements that collaborate at runtime to achieve some purpose.
5. Component diagram: Component diagrams consist of physical components like libraries, files, folders etc.

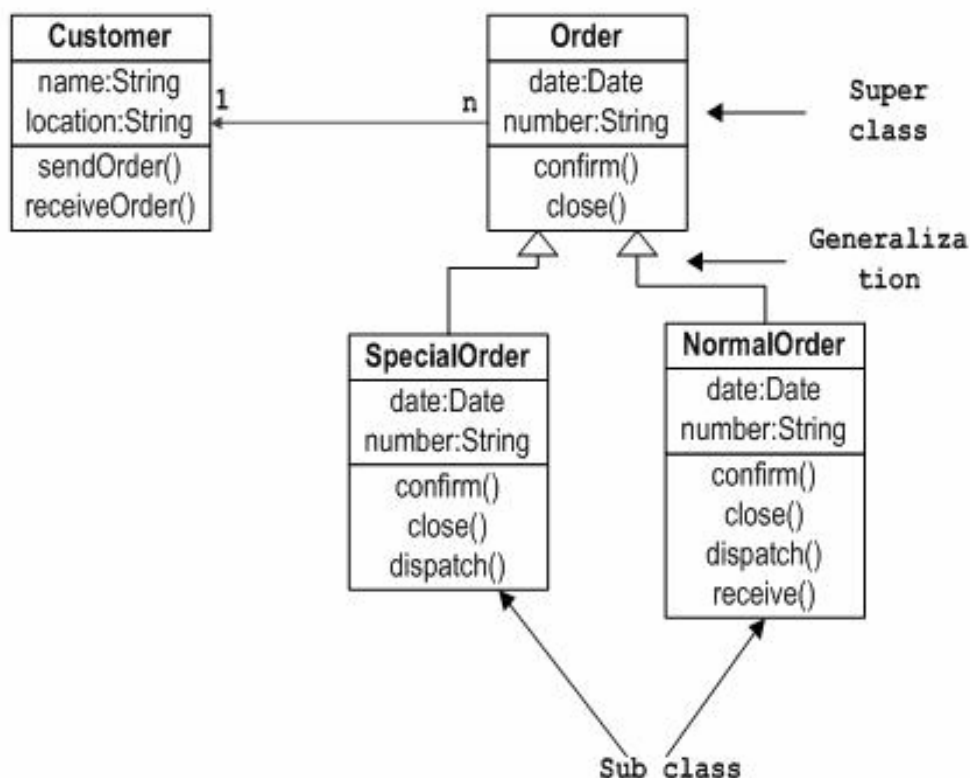
Q12) Read the following and draw diagrams given below.

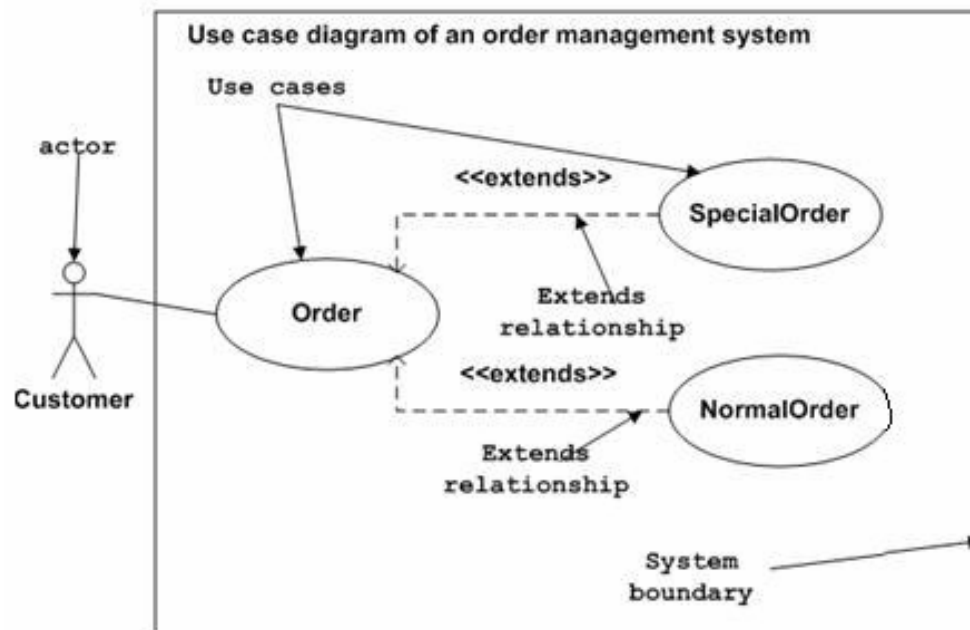
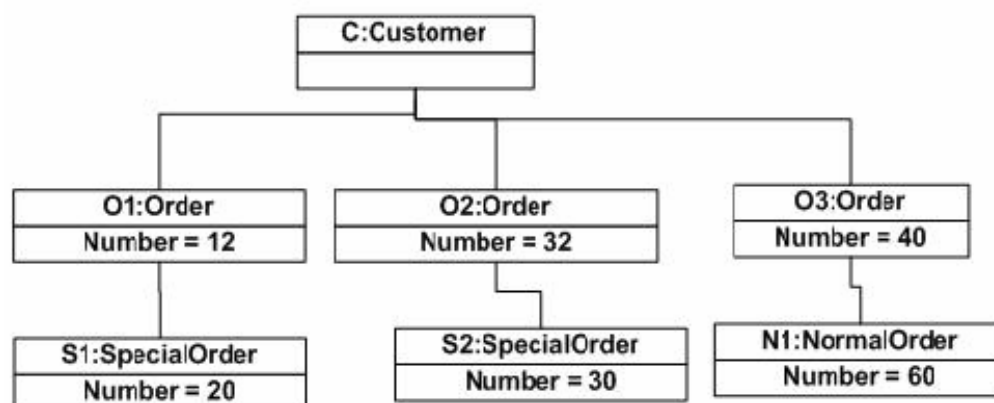
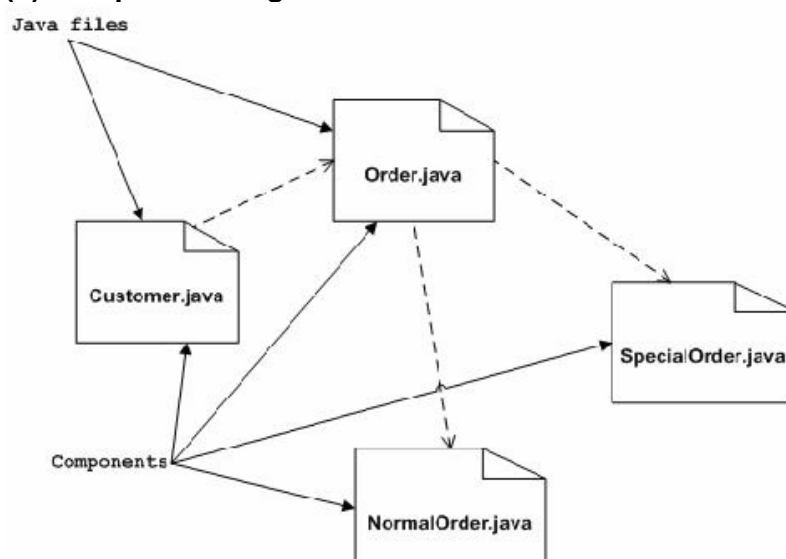
This is an Order Management System of an application.

Here a Customer can have multiple Orders, but an Order is related to exactly one Customer.

There are two types of Orders: SpecialOrder and NormalOrder.

SpecialOrder and NormalOrder have all the properties as the Order. In addition they have additional functions like dispatch and receive.

(a) Class Diagram:

(b) Use Case Diagram:**(c) Object Diagram:****(d) Component Diagram**

Assignment 2

(To be submitted on or before 26/5/2012)

Q13) Briefly describe following diagrams. (Continuation of Q11)

6. A Statechart diagram: It describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.
7. Interaction diagrams: Sequence and collaboration diagrams are the interaction diagrams used for this purpose. Sequence diagrams are used to capture time ordering of message flow and collaboration diagrams are used to understand the structural organization of the system.
8. Activity diagrams: Activity diagram consists of activities, links, relationships etc. It models all types of flows like parallel, single, concurrent etc. Activity diagram describes the flow control from one activity to another without any messages.
9. Deployment diagrams: Deployment diagrams are consists of nodes and their relationships.
10. Package diagrams: It depicts the dependencies between the packages that make up a model.

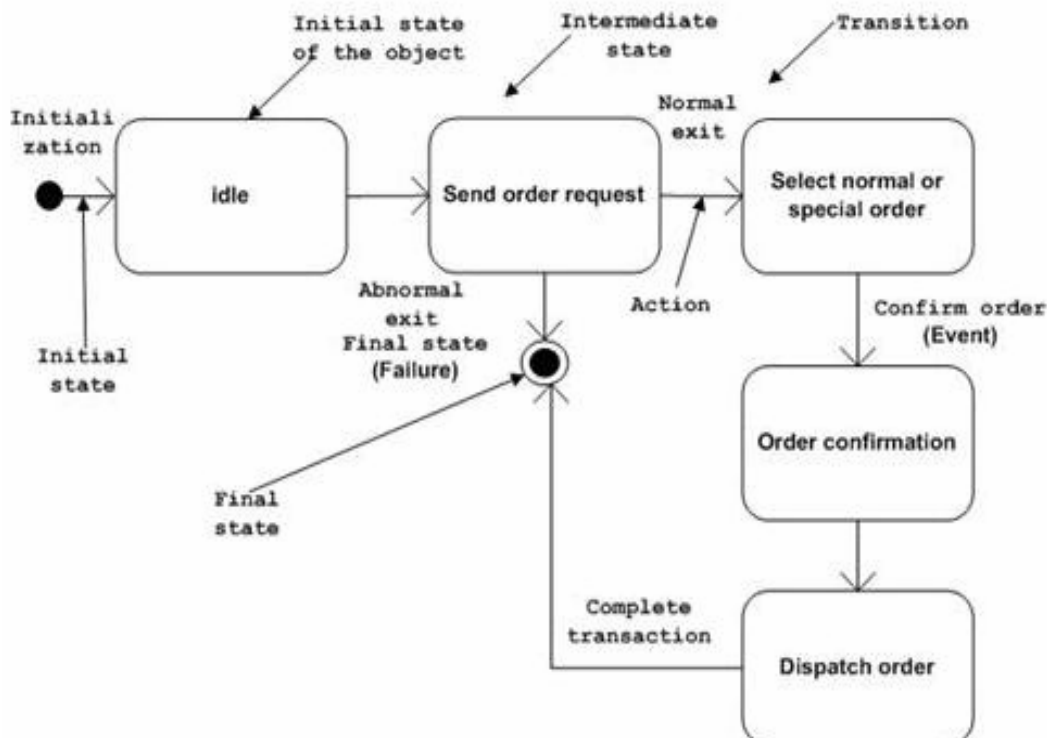
Q14) Read the following and draw diagrams given below. (Continuation of Q12)

This is an Order Management System of an application.

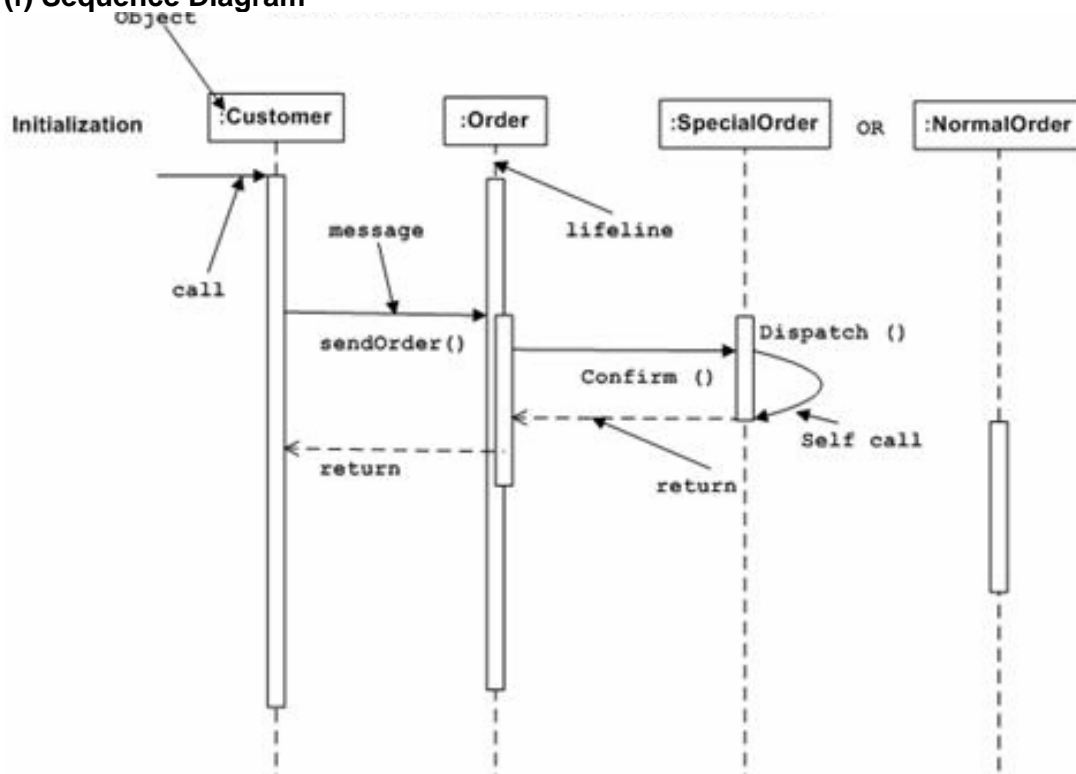
Here a Customer can have multiple Orders, but an order is related to exactly one customer.

There are two types of Orders: SpecialOrder and NormalOrder.

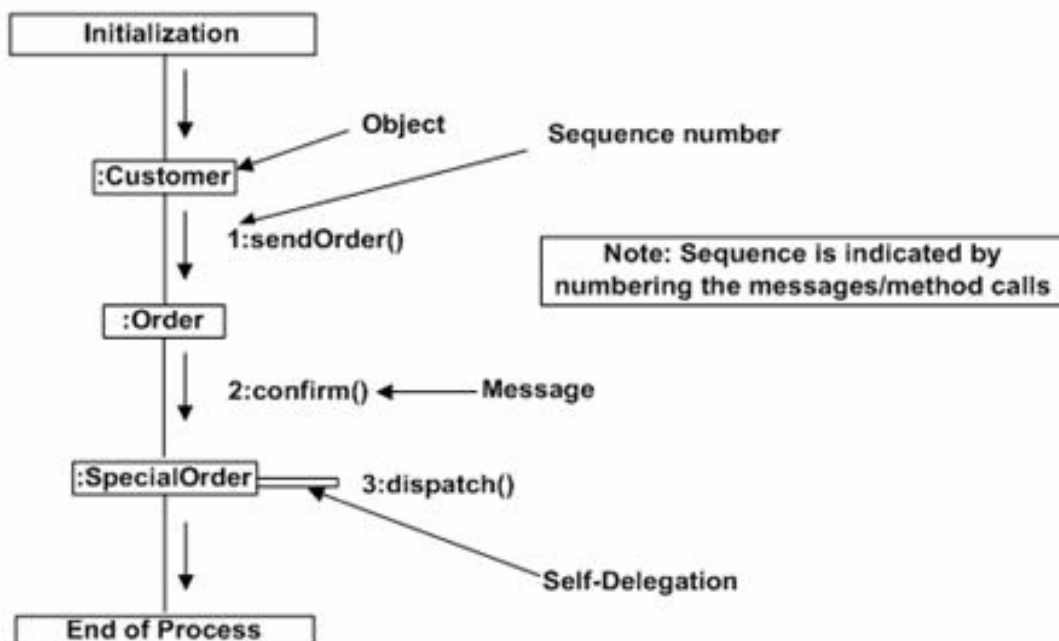
SpecialOrder and NormalOrder have all the properties as the Order. In addition they have additional functions like dispatch and receive.

(e) Statechart Diagram:

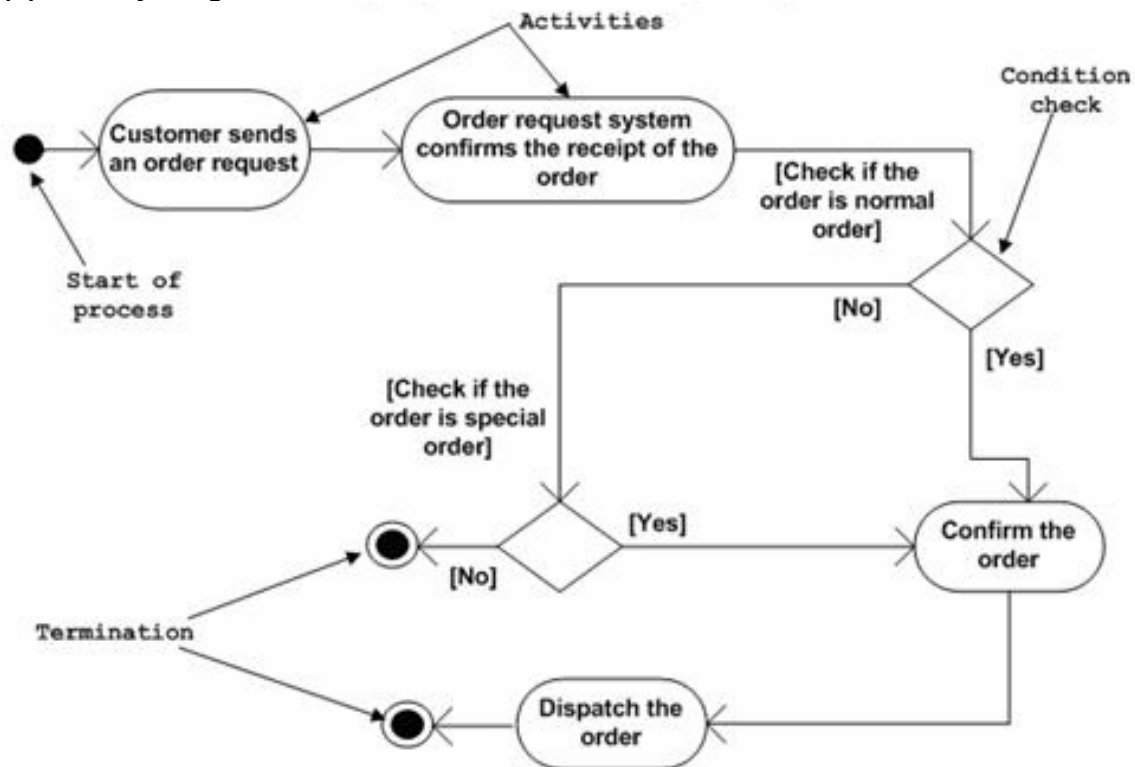
(f) Sequence Diagram



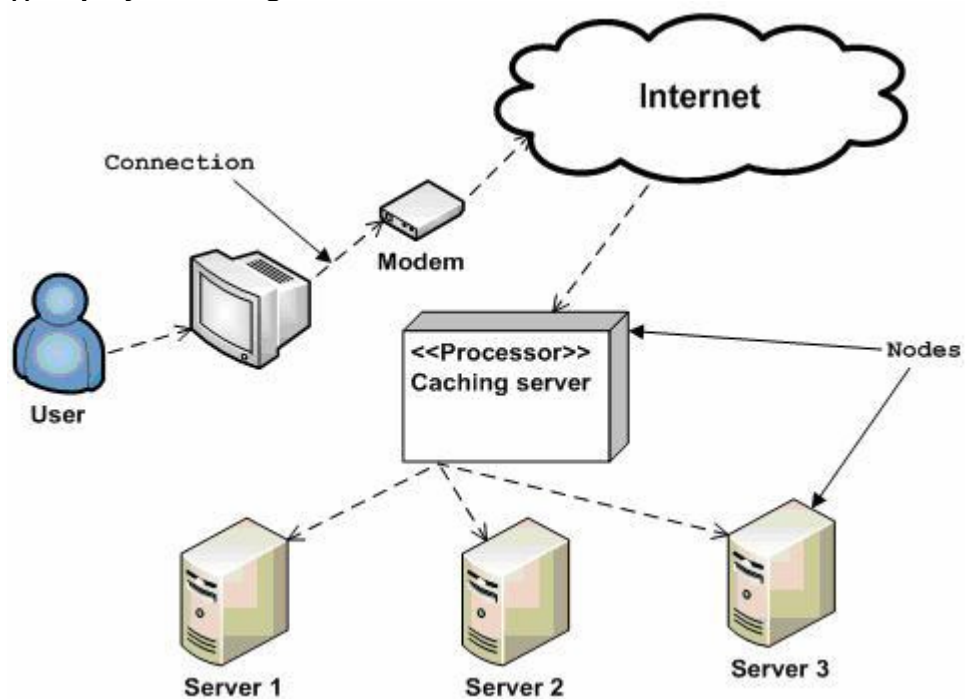
(g) Colaboration Diagram



(h) Activity Diagram:



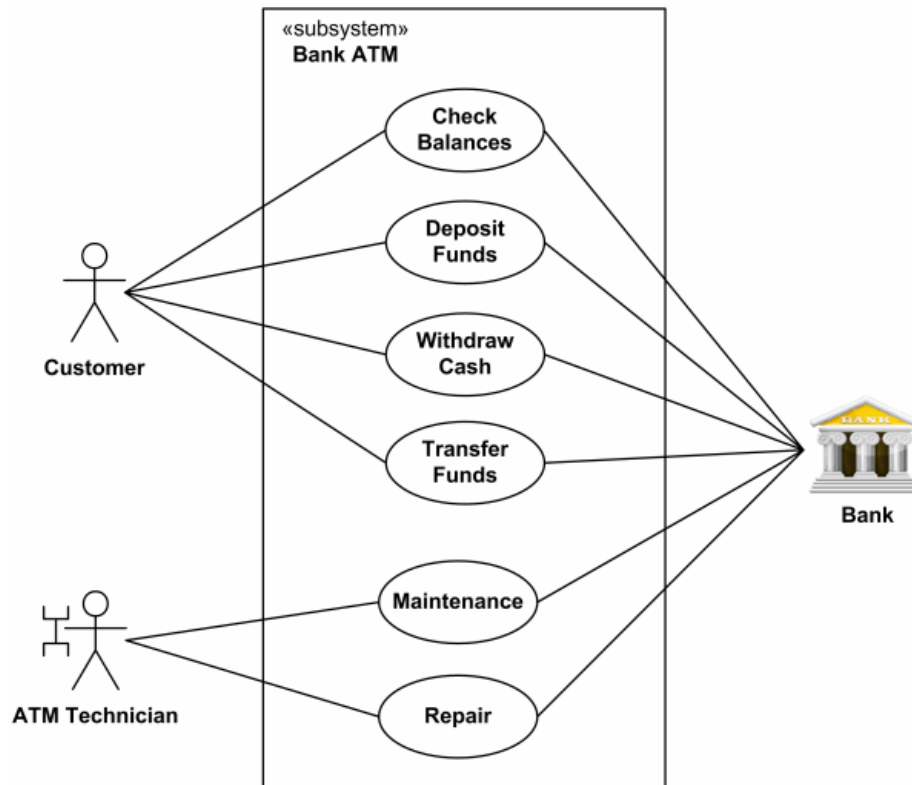
(i) Deployment Diagram



Q15) Read the following paragraph and draw a Use Case Diagram.

An automated teller machine (ATM) or the automatic banking machine (ABM) is banking subsystem that provides bank customers with access to financial transactions in a public space without the need for a cashier, clerk or bank teller.

Customer (actor) uses bank ATM to check balances of his/her bank accounts, deposit funds, withdraw cash and/or transfer funds (use cases). ATM Technician provides maintenance and repairs. All these use cases also involve Bank actor whether it is related to customer transactions or to the ATM servicing.



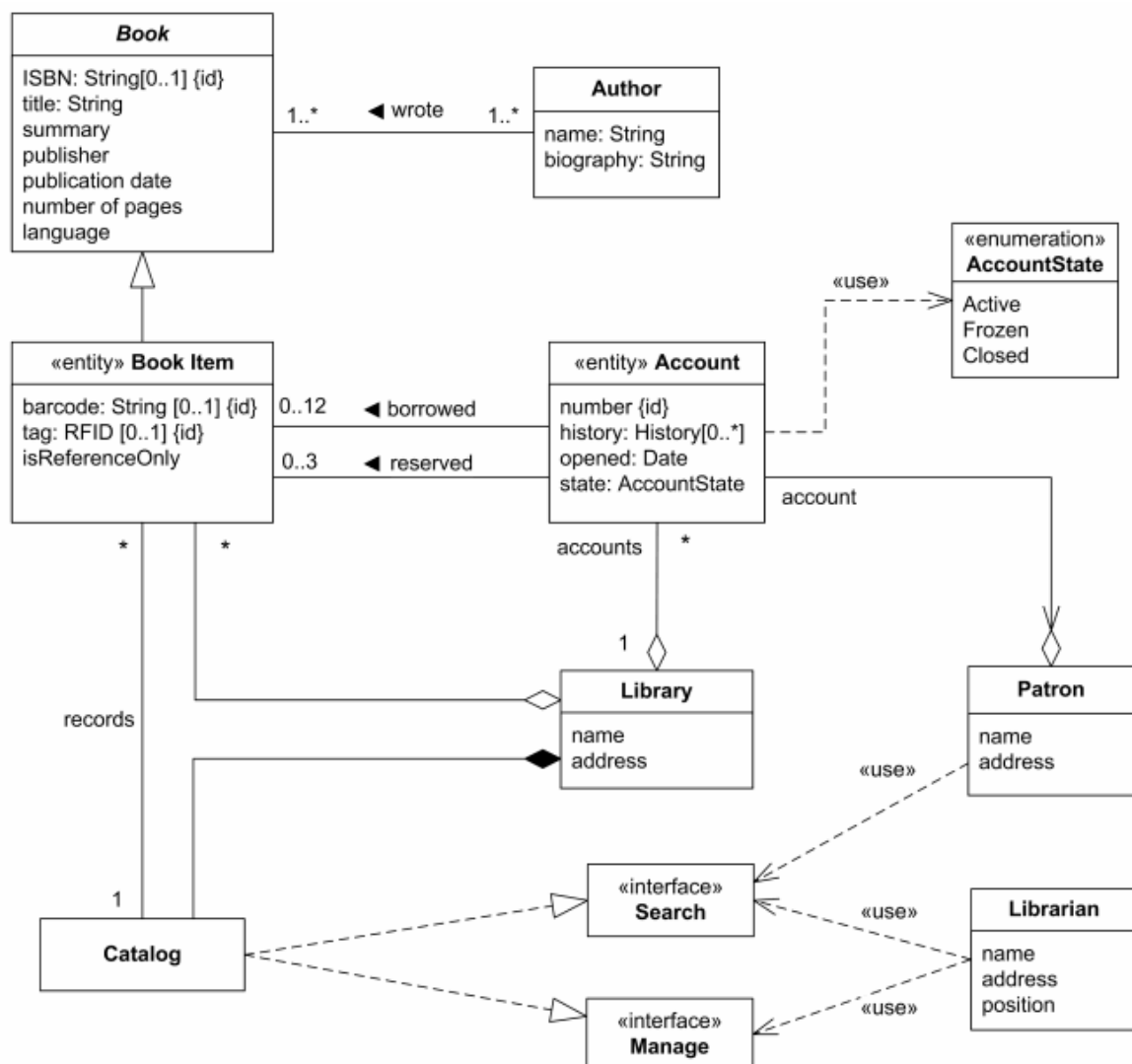
Q16) Read the following paragraph and draw a Class Diagram.

Library Domain Model describes main classes and relationships which could be used during analysis phase to better understand domain area for Integrated Library System (ILS), also known as a Library Management System (LMS).

Each physical library item - book, tape cassette, CD, DVD, etc. could have its own item number. To support it, the items may be barcoded. The purpose of barcoding is to provide a unique and scannable identifier that links the barcoded physical item to the electronic record in the catalog. Barcode must be physically attached to the item, and barcode number is entered into the corresponding field in the electronic item record.

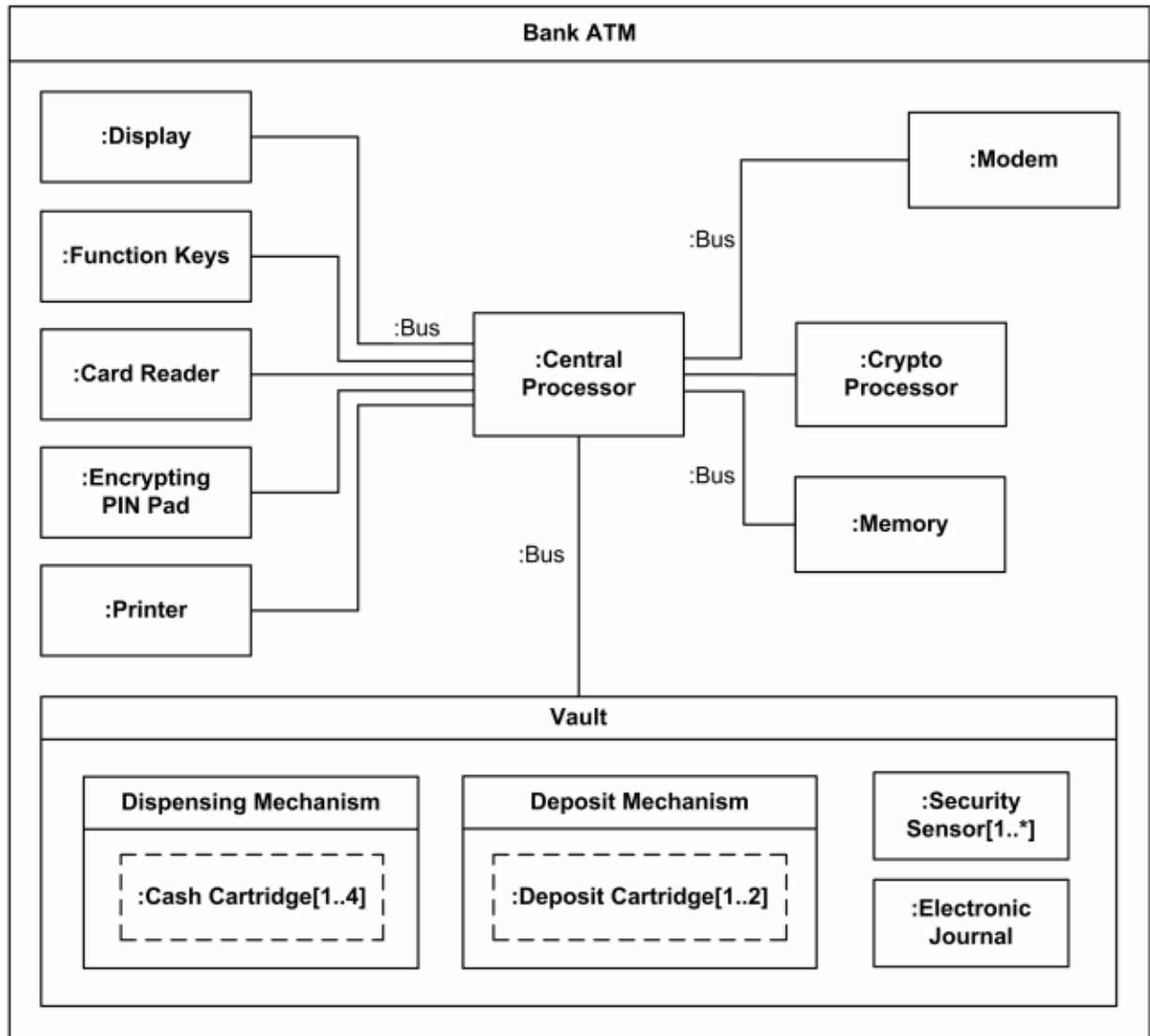
Barcodes on library items could be replaced by RFID tags. The RFID tag can contain item's identifier, title, material type, etc. It is read by an RFID reader, without the need to open a book cover or CD/DVD case to scan it with barcode reader.

Library has some rules on what could be borrowed and what is for reference only. Rules are also defined on how many books could be borrowed by patrons and how many could be reserved.



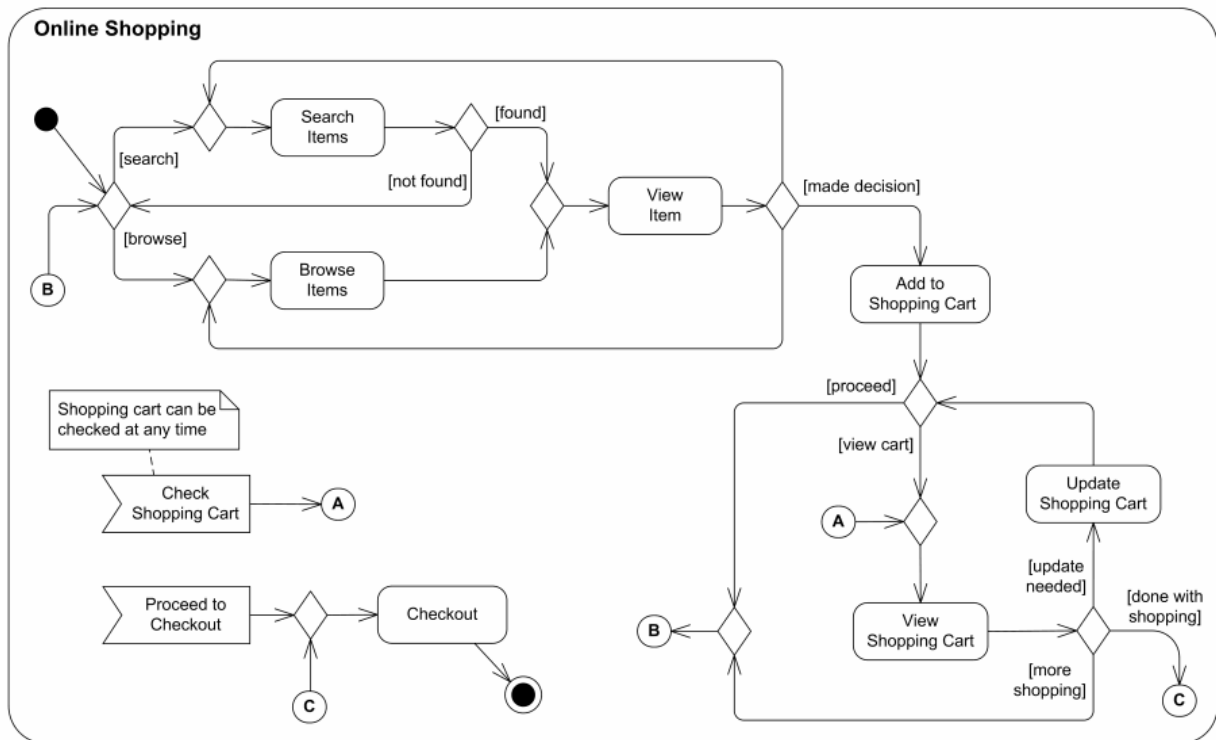
Q17) Read the following paragraph and draw a Composite Structure Diagram

Bank ATM is typically made up of several devices such as central processor unit (CPU), crypto processor, memory, customer display, function key buttons (usually located near the display), magnetic and/or smart chip card reader, encrypting PIN Pad, customer receipt printer, vault, modem.



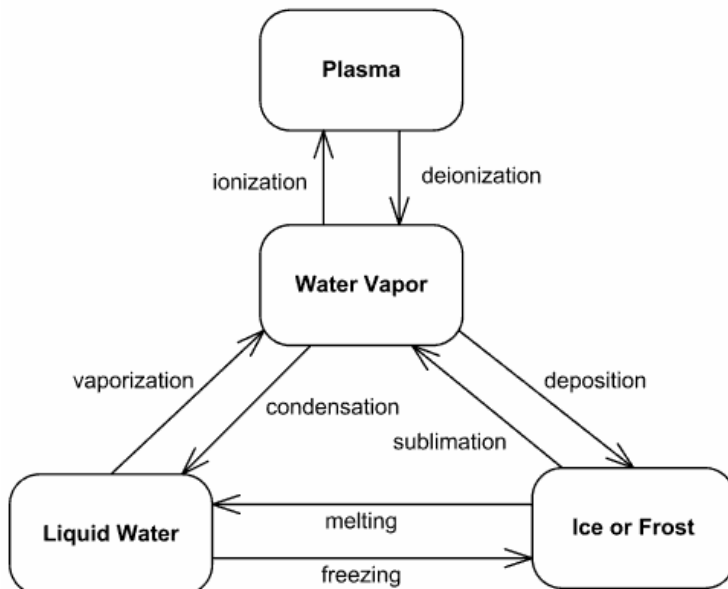
Q18) Read the following paragraph and draw an Activity Diagram

Online customer can browse or search items, view specific item, add it to shopping cart, view and update shopping cart, checkout. User can view shopping cart at any time. Checkout is assumed to include user registration and login.



Q19) Read the following paragraph and draw a Statemachine Diagram

Water can exist in several states - liquid, vapor, solid, and plasma. Several transitions are possible from one state to another. For example, freezing is phase change from liquid state to ice. Condensation is phase change from vapor state to liquid. Water vapor could turn directly into frost through deposition.



Q20) Read the following paragraph and draw a Package Diagram

Web Shopping, Mobile Shopping, Phone Shopping, and Mail Shopping packages merge Shopping Cart package. The same 4 packages use Payment package. Both Payment and Shopping Cart packages import other packages.

