

# Introduction

- **System Analysis** is a process of finding the problems and requirement rather than solution in order to define system's goals, purpose and operation. If new ATM System is desired analysis focus and how it can be used and what are its function.
- **System Design** is the process of defining the modules, data, interface for the system to satisfy requirements. It focus on solution and implementation.

# What is OOAD?

- **OOAD** is an approach used in the analysis and design of an application or system through the use of the object oriented paradigm.
- It focus on describing the object rather than problem domain.
- OOAD focus on define the software objects and how they are collaborate or interact to fulfill the requirements.

# Class

- Class describes a group of objects with similar Properties(attributes), common behavior (operations), common relationships with common relationships to other objects.

# Objects

- Object is a concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand.
- Object promote understanding of real world and provide a practical basis for computer implementation.
- All the objects are identity and are distinguishable.
- Objects are distinguished by their inherent existence and by descriptive properties that they have.

# Difference between Class and Objects

## Class

- A class is a template for creating objects in Program.
- A class is a logical entity.
- It does not create memory when it is created.
- Class generate objects.
- Example Car

## Object

- The Object is an instance of class.
- Object is a physical entity.
- Object create memory when it is created
- Objects provide life to the class.
- Example Hundai, Toyata



# UML( Unified Modeling Language)

- A UML is a visual or graphical language for specifying, constructing and documenting the artifact of the system that are going to be developed.
- It is standard diagrammatic notation for drawing or presenting the picture related to Software.
- It is a pictorial language used to represent software's blue print.
- For Ex. In class diagram class is represented as rectangle box.

# Some UML Diagrams-

- Class Diagram
- Use case Diagram
- State Diagram
- Sequence Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

# Unified Process (UP)

- It is a popular iterative and incremental software development framework.
- In this process team members work closely with stakeholders to ensure that the process is continuously updated and improved to reflect recent experiences and proven best practices.
- Team productivity is increased by supporting in every sectors such as common language, process and view on how to develop system.



# Phases of Unified Process (UP)

- Inception Phase
- Elaboration Phase
- Construction Phase
- Transition Phase

# Phases of Unified Process (UP)

- Inception Phase
  - In this phase following analytical activities are done
    - Success Criteria
    - Risk Assessment
    - Estimates of the resources needed
    - Preparation of development environment
    - Function that must be Included and Excluded
    - Define lifecycle objective

# Inception Phase

The milestone in this phase is a "Lifecycle objectives". The indication that the project has reached this milestone includes.

- The major stake holder agrees on the scope of the proposed system.
- If a set of critical high level requirements are addressed by business scope.
- The business case of the project is strong enough to continue the development.

**This process also define what process to use and what tools to use.**

# Elaboration Phase

- In this phase problem domain is analyzed and sound architectural structure established.
- Architectural Design is constructed based on scope, functionality and non functional requirements.
- This phase involves refinement of vision, iterative implementation of the core architecture, resolution of high risk, identification of more requirement and scope.
- It is the initial series of iteration during which the team does the serious investigation, implement the core architecture and tackle with high risk.

# Elaboration Phase

- The major milestone associated with the elaboration phase is “ LifeCycle Architecture”. The indication that the project have reached this milestone include.
  - Most of the functional requirement has been captured in use case model.
  - The architecture baseline is small.
  - Project team has initial project plan that describe how the construction phase will proceed and if business case has received green signal.



# Construction Phase

- It is major phase of software development which is concerned with system design, programming and testing.
- The project team perform the task that involves building the system iteratively and incrementally.
- All the component and application feature are developed and integrated into the product and are thoroughly tested.
- On completion of this phase we have a working software system.

# Construction Phase

- The major milestone associated with this phase is “ Initial Operational Capability which indicates whether the project is ready to be deployed in beta test environment.
  - If a set of beta customers has a more or less fully operational system in their hands.
  - If the project release is stable and mature enough to be deployed.
  - If actual resource expenditure vs standard resource expenditure is acceptable.

# Transition Phase

- Its purpose is to transit the final software product to customer environment.
- The Project team focuses on correcting the defect and modifying the system to correct previously unidentified problem.
- It also include system conversion and system training.
- Feedback also incorporated to make best system.

# Transition Phase

- The major milestone associated with the transition phase is “Product Release” which indicates whether the project users requirement or not and start plan for new development. The indication that the project have reached this milestone include.
  - If the users are satisfied.
  - If actual resources vs standard resources expenditure is still acceptable.

# Why to use unified process

- Support Iterative and incremental development.
- Manage requirement
- Use case Driven
- Architecture Centric
- Control Change to software
- Continuously verify Software Quality



# Use cases

- A behaviorally related sequence of steps ( a scenario), both automate and manual, for the purpose of completing single business task.
- It is a collection of success and failure scenarios that describe actor using a system to support a goal.
- It is made up of set of possible sequence of interaction between the system and actor in a particular environment.

# Actor

- Actor is something with behavior such as person computer system or organization that use the service of the system or provide the service to the system.
- It specifies a role played by a user or any other system that interact with the subject.
- Actors are play not only by people but also by organization, software and machine.
  - Types of Actor
    - Primary Actor
    - Supporting Actor
    - Offstage Actor

# Types of Actors

- **Primary Actors** are the stakeholder that primarily benefits from the execution of the use case by receiving something of measurable or observable value.
- **Supporting Actors** are those actors which provide assistance from other supporting system to fulfill primary actors goal. Example- ATM System in ATM Machine.
- **Offstage Actor** are persons who has an interest but not primary actor.

# How to find Use Cases





- Choose the system Boundary.
- Identify the primary actor.
- Identify the goal for each primary actor.
- Define the use case that satisfy the user's goal.

# Use Case Diagram

- A diagram that depicts the interaction between the system, external system and users. In other words, It graphically describes who will use the system and in what ways the user expects to interact with the system.
- It show the relationship between the user and use case in which the user is involved.
- The use cases is represented by circle or eclipse in use case diagram.

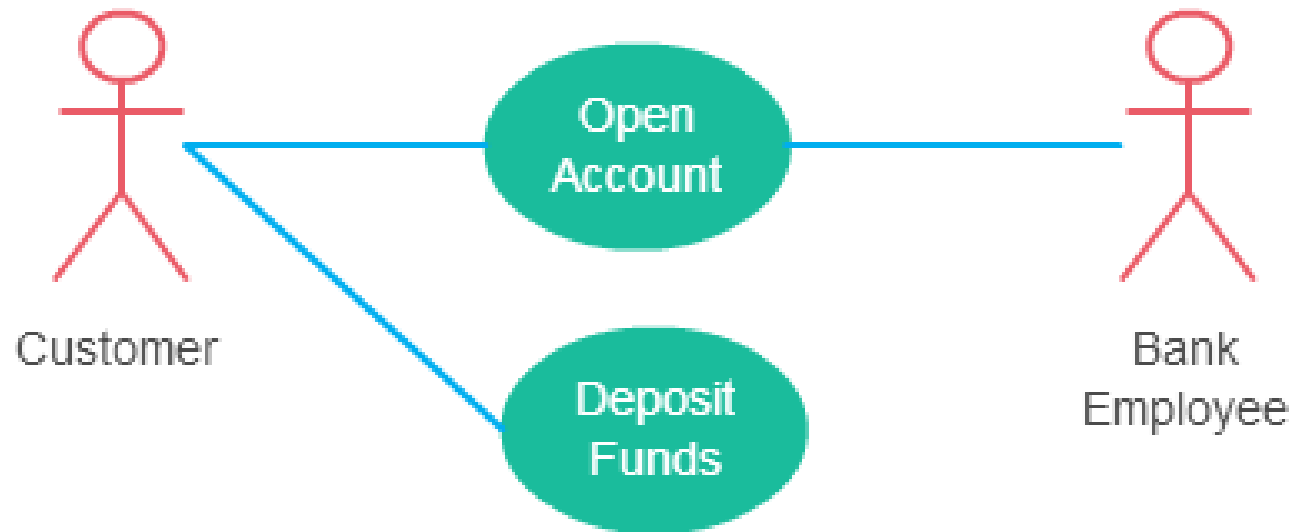


# Use Case Diagram

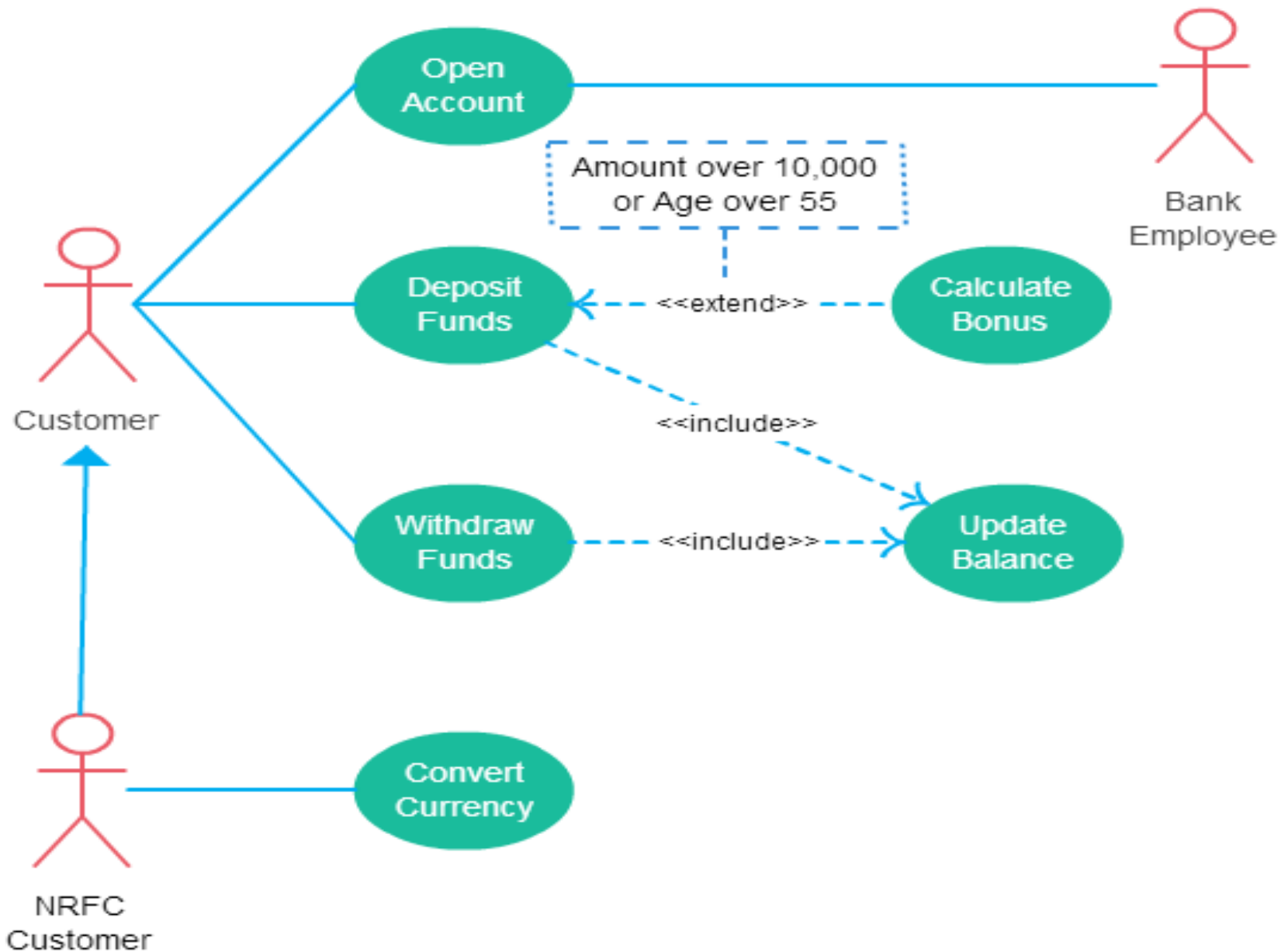
Relationship	Symbol	Meaning
Communicates		An actor is connected to a use case using a line with no arrowheads.
Includes		A use case contains a behavior that is common to more than one other use case. The arrow points to the common use case.
Extends		A different use case handles exceptions from the basic use case. The arrow points from the extended to the basic use case.
Generalizes		One UML "thing" is more general than another "thing." The arrow points to the general "thing."

# Use Case Relationship

- Association Relationship
  - It shows the connection between actor and system. It shows which actor will interact with system. It is represented by solid line.



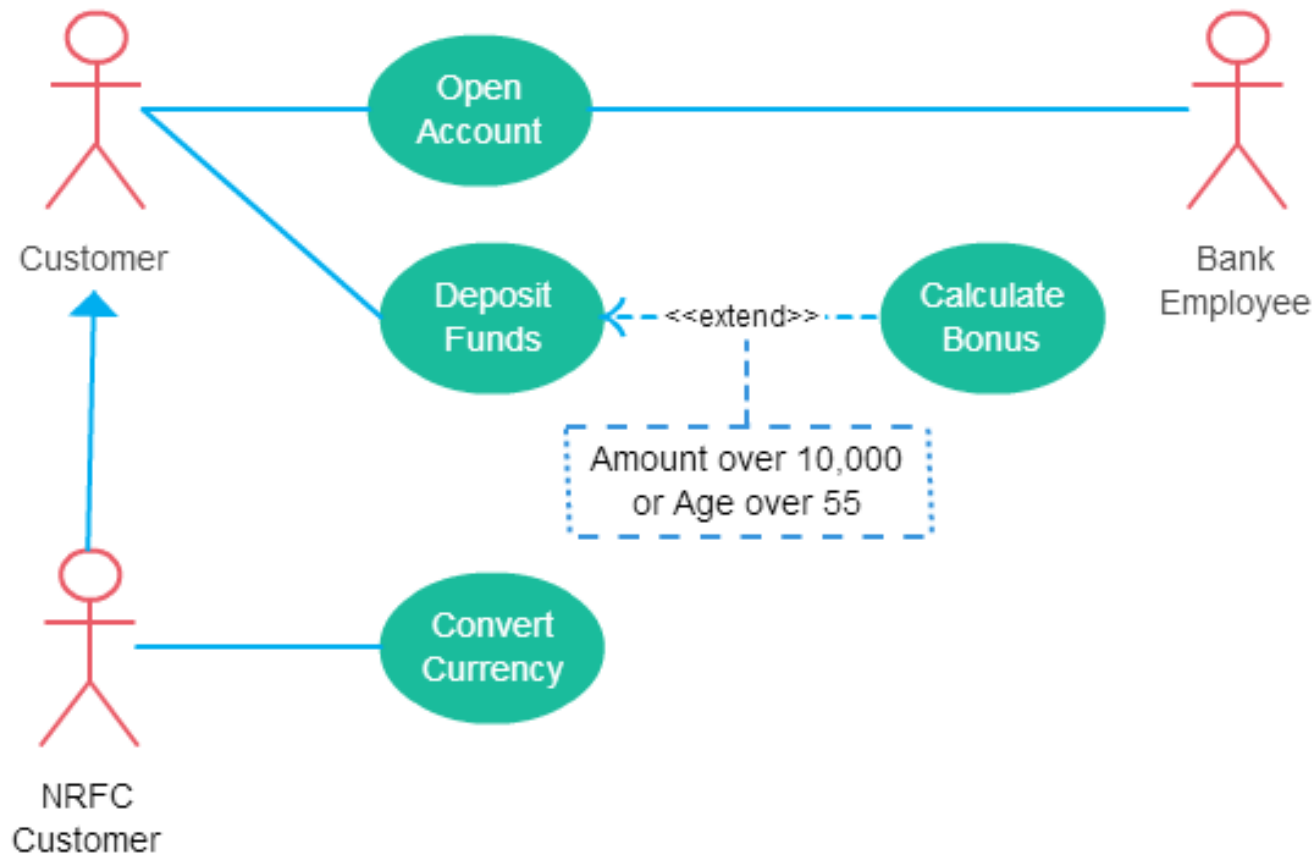
# Use Case Relationship



# Use Case Relationship

- Extend Relationship

- The function of the base use case is extended by optional use case. It add more functionality to the system.



# Use case Modelling

- Use case modeling is the process of modeling a system's functions in terms of business events, who initiated the events, and how the system responds to those events.
- Use case modeling is an approach that facilitates usage centered development. Use case modeling has proved to be a valuable aid in meeting the challenges of determining what a system is required to do from a user and stakeholder perspective.
- It is now widely recognized as best practice for the defining, documenting, and understanding of an information system's functional requirements.
- Use case modeling facilitates and encourages user involvement, which is one of the primary success factors for ensuring project success.



# Advantages of use case Modelling

- Provides a tool for capturing functional requirements.
- Assists in decomposing system scope into more manageable pieces.
- Provides a means of identifying, assigning, tracking, and controlling all the activities related to system.
- Provides a tool for requirement traceability.
- Provides a starting point for the identification of data objects or entities.
- Provides a framework for driving the system development project.

# Process of use case model (Steps)

- Identify business actors.
- Identify business requirements use cases
- Construct use case model diagram
- Document business use case narratives.

# Identify Business Actor

- Actors can be used to verify and validate the use cases when they are finished.
- Context Diagram, existing document, user manuals and meetings are conducted to search actors.
- Various questions related to system can be used to verify the actors. For library student, librarian and staff can be taken as actors.

# Identify Business Reqn. Use case

- A use case describes how a real world actor interacts with the system, an excellent technique for finding business requirements use cases is to examine, actors and how they will use the system.
  - When looking for a use case following question can be asked.
  - What are the main tasks of the actor?
  - What information does the actor need from the system?
  - What information does the actor provide to the system?

# Construct Use case Diagram

- Once the use cases and actors have been identified, a use case model diagram can be used to graphically the system scope and boundaries.
- A single system may contain dozens of use cases.

# Document use case Diagram

- This document verifies the process use case modeling.
- To keep the records of the used case following things must be stored.

Author, Date, Version, Use case name, use case type, use case ID, Priority, Source, Primary Business Actor, Other Participating Actor, Interested stakeholders, Descriptions,



# Class

Class is a classification of object which encapsulate data and behavior. It gives the blueprint or description of the object that can be created from it.

## Relationship Among Class

- Association
- Aggregation
- Generalization

# Generalization

- A generalization relationship is a parent-child relationship between use cases.
- The child use case is an enhancement of the parent use case.
- Generalization is shown as a directed arrow with a triangle arrowhead.
- The child use case is connected at the base of the arrow. The tip of the arrow is connected to the parent use case.