SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune

A MINI PROJECT REPORT ON

Online Banking Management System UML diagrams

Under the guidance of
Prof. Snehal Shintre

DEPARTMENT OF COMPUTER ENGINEERING
ACADEMIC YEAR 2022-23

Title:

Online Banking Management System

Team:

1. Abhishek Saware - 41269
2. Tejas Yadav 41282
3. Sharad Ugalmugle - 41280

Theory:

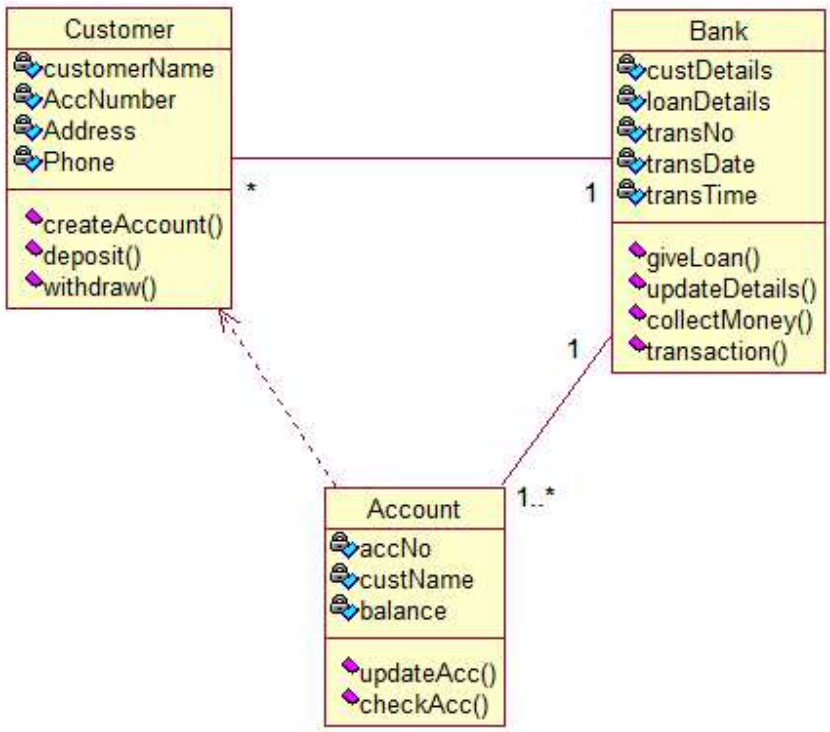
- UML (Unified Modeling Language) is a general purpose modeling language which was adopted as a standard in 1997
- UML is not exactly a programming language, but a visual language
- It helps in visualizing a system's design using a set of conventions and rules
- UML helps businesses, architects and engineers for
    - Modeling
    - Design
    - Analysis
- Need for UML -
    - Complex applications necessitate collaboration and planning from multiple teams, entailing a clear and concise means of communication among them.

- Code is not understood by businesspeople. As a result, UML becomes essential for communicating the system's essential requirements, functionalities, and processes to non-programmers.
- When teams can visualize processes, user interactions, and the system's static structure, they save a lot of time in the long run.

Diagrams :

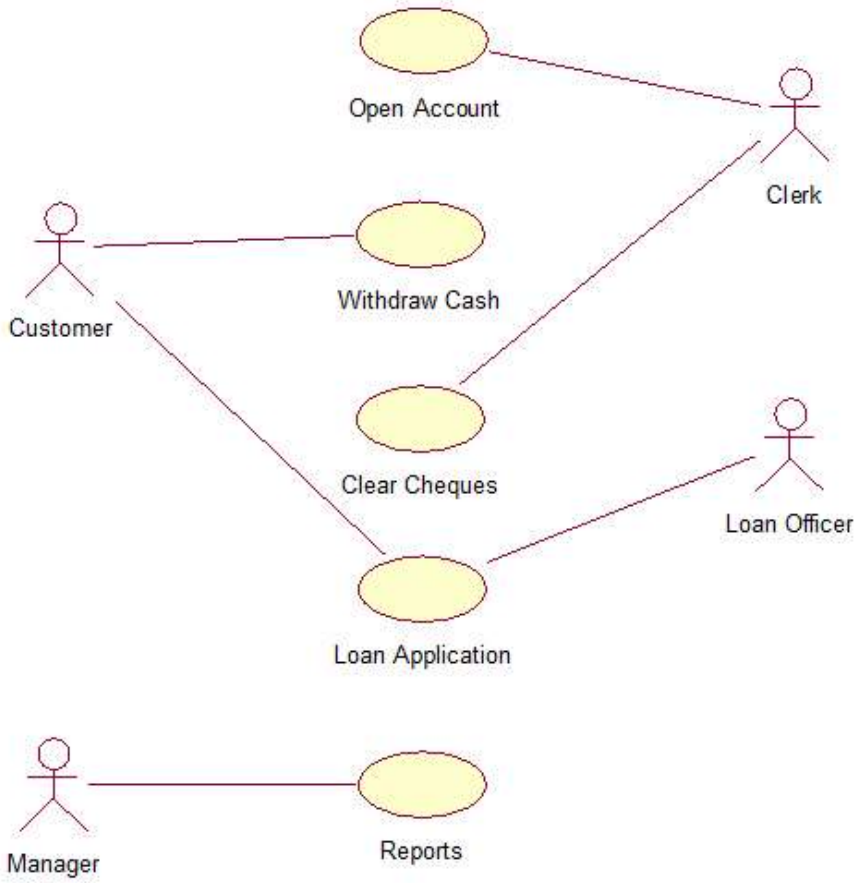
(i) Class diagram

The class diagram is the most commonly used UML diagram. It serves as the foundation for all object-oriented software systems. Class diagrams are used to depict the static structure of a system by displaying the system's classes, methods, and attributes. Class diagrams also assist us in determining the relationships between various classes or objects.



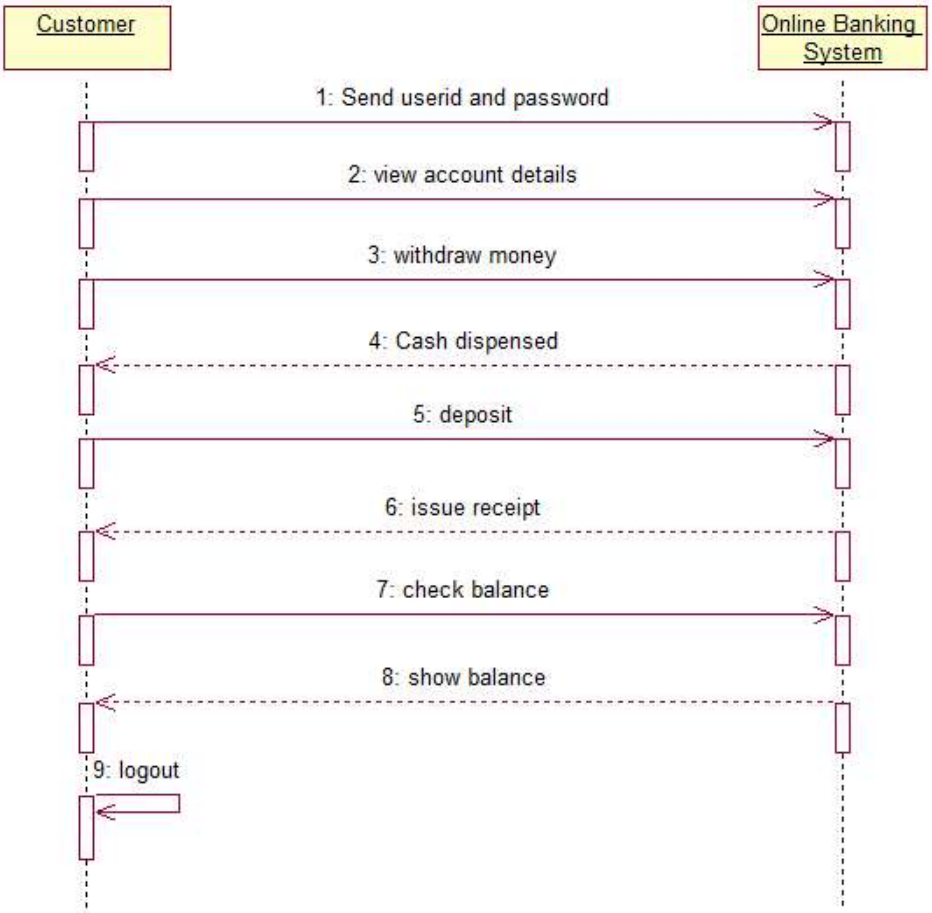
(ii) Use case diagram

Use Case Diagrams depict the functionality of a system or a component of a system. They are commonly used to demonstrate the system's functional requirements and interaction with external agents (actors). A use case is essentially a diagram that represents various scenarios in which the system can be used. A use case diagram provides a high-level overview of what the system or a component of the system does without delving into implementation details.



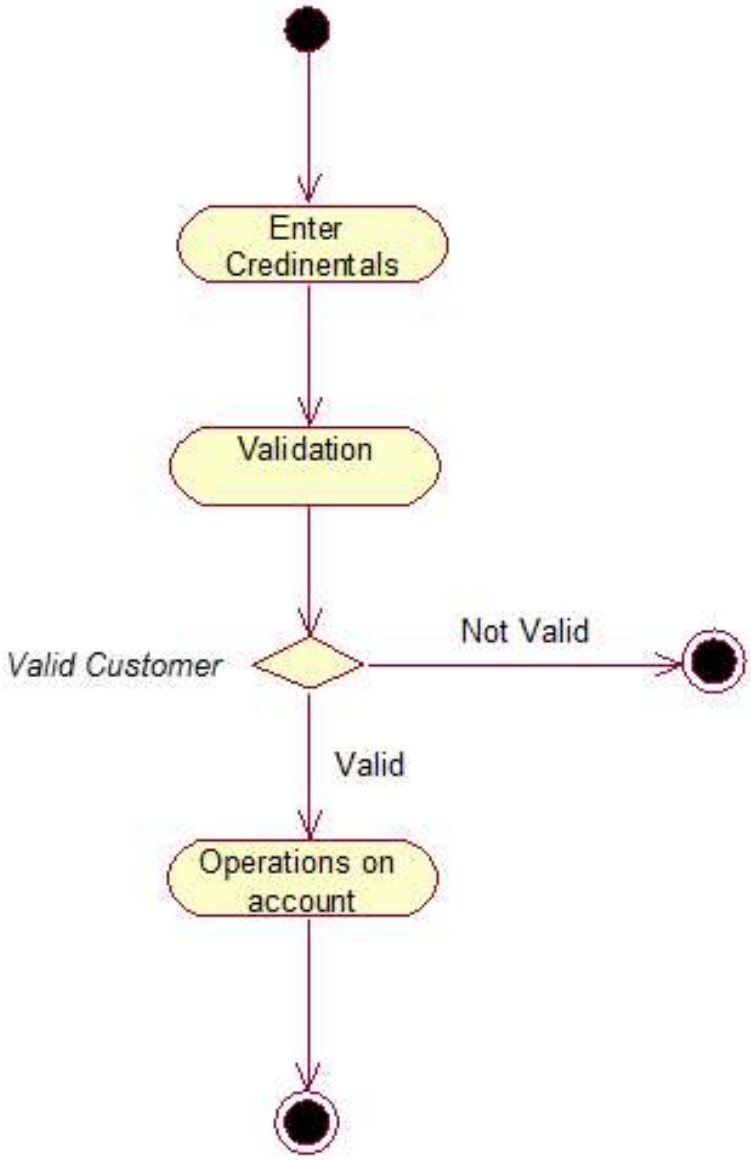
(iii) Sequence diagram

A sequence diagram simply depicts object interactions in a sequential order, i.e. the order in which these interactions occur. A sequence diagram can also be referred to using the terms event diagrams or event scenarios. Sequence diagrams show how and in what order objects in a system work. Businesspeople and software developers frequently use these diagrams to document and understand requirements for new and existing systems.



(iv) 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.



(v) Component diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.