SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune

LP-IV Laboratory Mini Project on

Bank Management Application

SUBMITTED BY

41443 - Anuj Mutha

Under the guidance of
Prof. Laxmi Pawar

DEPARTMENT OF COMPUTER ENGINEERING
ACADEMIC YEAR 2022-23

Contents

List of Figures

1 Title

UML for Bank management Application

2 Problem Statement

Draw following UML Diagrams for Bank Management application

1. Class Diagram
2. Object Diagram
3. ER Diagram
4. Component Diagram

3 Objectives and Scope

Objective of this project is to draw various UML diagrams for bank management application

This project will help us to learn about various UML Diagrams used in software development lifecycle.

4 Software and Hardware Requirements

- MacOSX (64 bit).
- RAM-8GB
- ERDPlus
- Visual Paradigm

5 Methodological Details

There are various types of UML diagrams as follows:

5.1 Class Diagrams

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

5.2 Object Diagrams

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

5.3 ER Diagrams

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

5.3.1 Entity

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

5.3.2 Attributes

Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes. There exists a domain or range of values that can be assigned to attributes.

5.4 Component Diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

6 Outcome

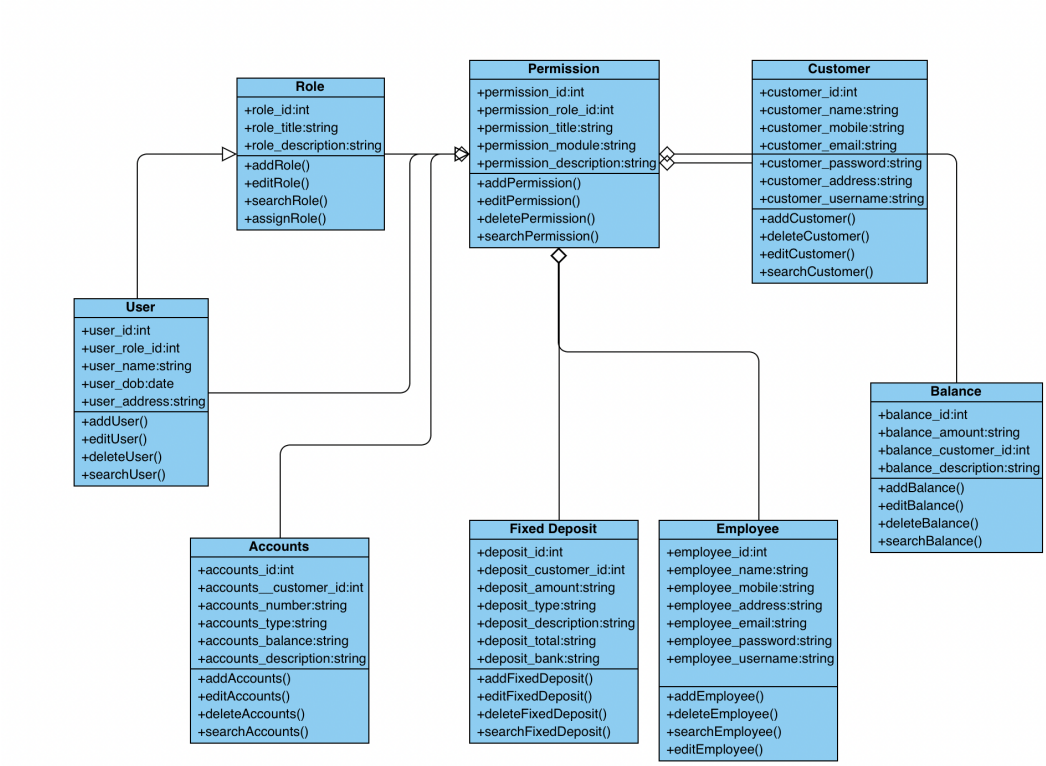

Figure 1: Class Diagram

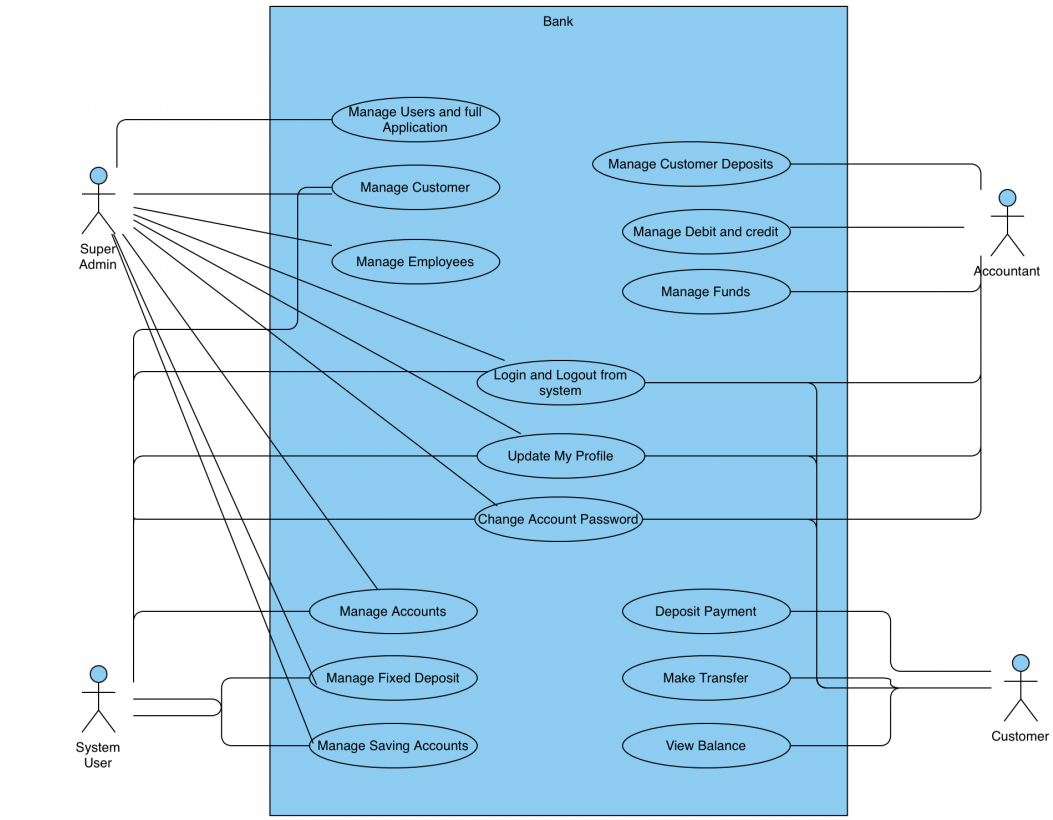

Figure 2: Use Case Diagram

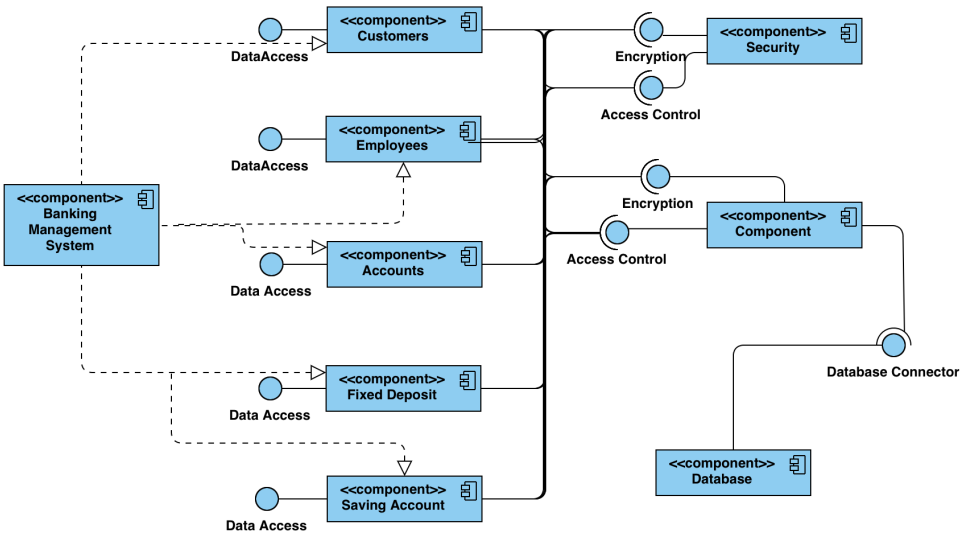

Figure 3: Component Diagram

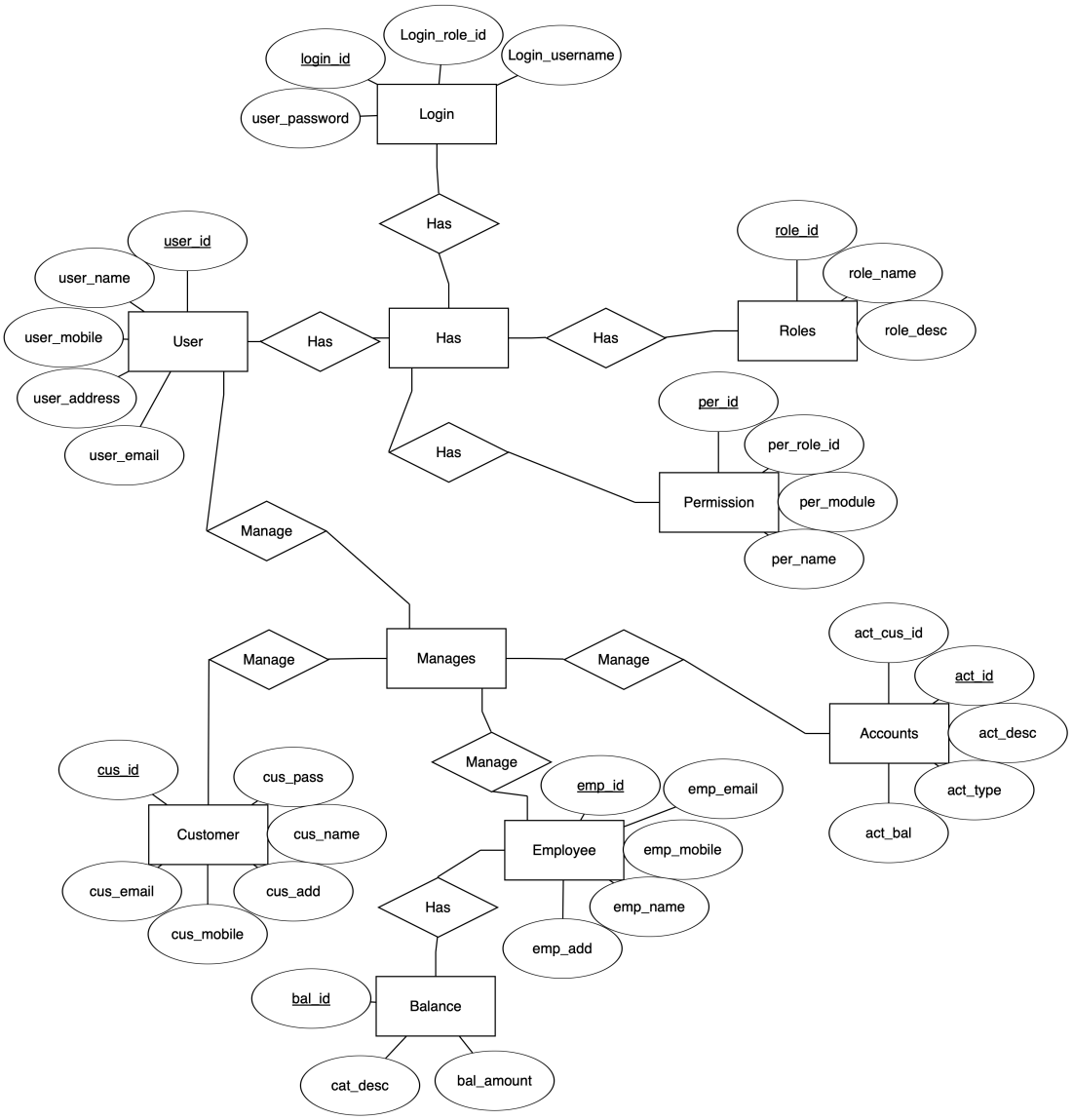

Figure 4: ER Diagram

7 Conclusion

While developing the system a conscious effort has been made to create and develop software package, making use of available tools, techniques, and resources – that would generate a proper system for cases.

While making the system, an eye has been kept on making it as user-friendly. A such may hope that the system will be acceptable to any user and will adequately meet his /her need. As in the case of any system development process where there are a number of shortcomings, there have been some shortcomings in the development of this system also.