# Chapter 3: Design

Design specifies the structure or behavior of how the system or software functions and how it will be written before writing the complete implementation. The design for the Mobile shop management system project consists of structural, behavioral, database and architectural diagrams which helps in implementation and understanding the systems architecture.

# Structural Design

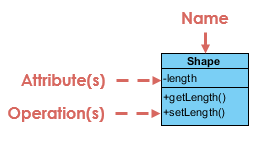
## Class Diagram

Class diagram is a graphical representation of structure of a system which includes classes, attributes and methods or operations and their relationship between different classes or objects. It’s a building block and important diagram for implementation on object-oriented software systems.

The reason for using class diagram are as follows:

* Helps during implementation or makes it easier to identify classes relationship.
* Displays the needed attributes and methods or operations for the system.
* Helps in building system as it can be used as blueprint for final system.
* It describes the static view of the system.

The notations used for the class diagram of mobile shop management system are as follows:

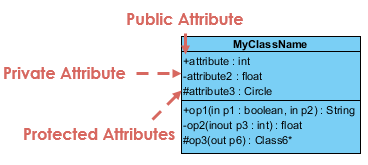


**Class name:** The UML representation of a class is a rectangle containing three compartments stacked vertically and the first compartment include class name.

**Class attribute:** The attribute segment of a class lists each of the class's attributes on a separate line along with attribute types.

**Class operations:** The operations are written at the last compartment and it shows the methods or operation that following class will perform on system.

1. **Class Visibility:**



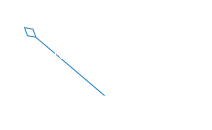
The visibility symbols are +, -, #. **+** represents public types attributes or operations, **-** represents private types attributes or operations and **#** represents protected attributes or operations.

1. **Association:**



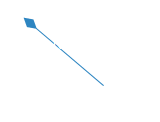
It’s a simple relationship between two classes which is represented by a single solid line or one-sided arrow line.

1. **Aggregation:**



It is a special type of association in which objects are assembled or configured together to create a more complex object.

1. **Composition:**



It represents whole-part relationships and is a form of special type of aggregation where parts are destroyed when whole is destroyed.

Class diagram of Mobile shop management system is shown below:

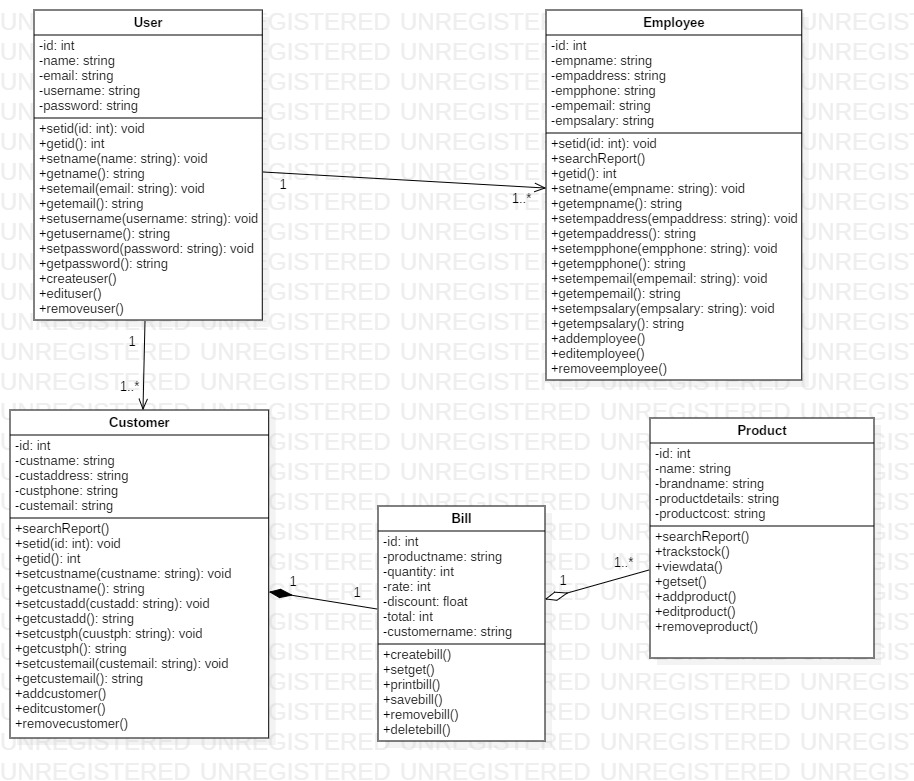


Figure 1: Final Class Diagram

The above diagram shows the final class diagrams of mobile shop management system with proper relation between classes or objects. It shows the association, aggregation and composition relationship along with multiplicity.

## Data Flow diagram (DFD)

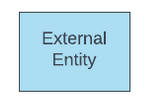
Data flow diagram is a structural representation of a software system. It shows the flow of information about where the data comes from and goes as well as how it is stored. DFD shows how data is handled in a system and defines a difficult explanation through diagrams.

The reason for using data flow diagram are as follows:

* Simple notation and easy to understand data flow of system.
* Shows the scope and boundaries of a system.
* Act as the starting point for redesigning a system.
* Logical information flow of a system.

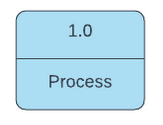
The notations used for the data flow diagram of mobile shop management system are as follows:

1. **External Entity:**



The sources and destination of systems input and output is external entities. It is object outside the system with which the system communicates.

1. **Process:**

****

A process represents the transformation of incoming data flow into outgoing data flow.

1. **Datastore:**

****

Datastore are repositories of data in the system also known as files sometimes.

Data flow diagram of mobile shop management system is shown below:

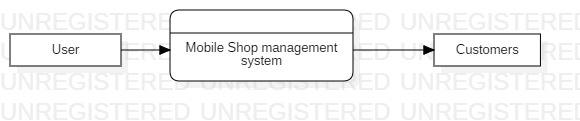


Figure 2: Level 0 Data flow diagram

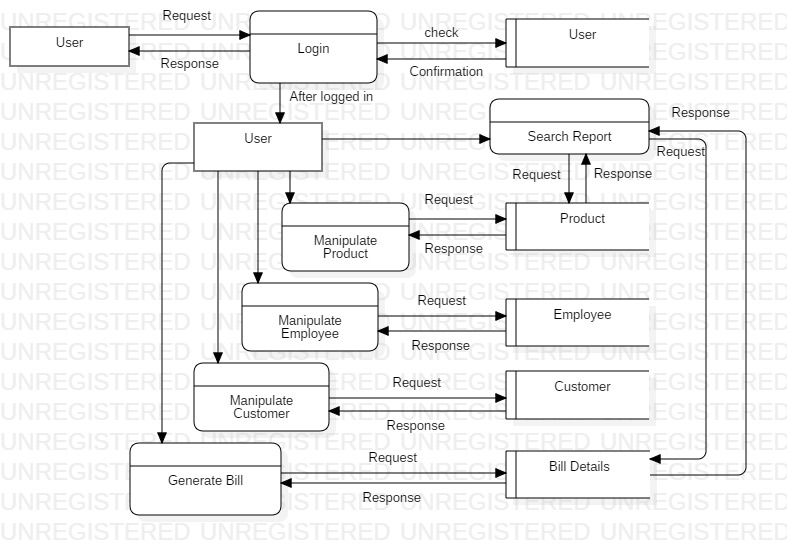


Figure 3: Level 1 Data Flow Diagram

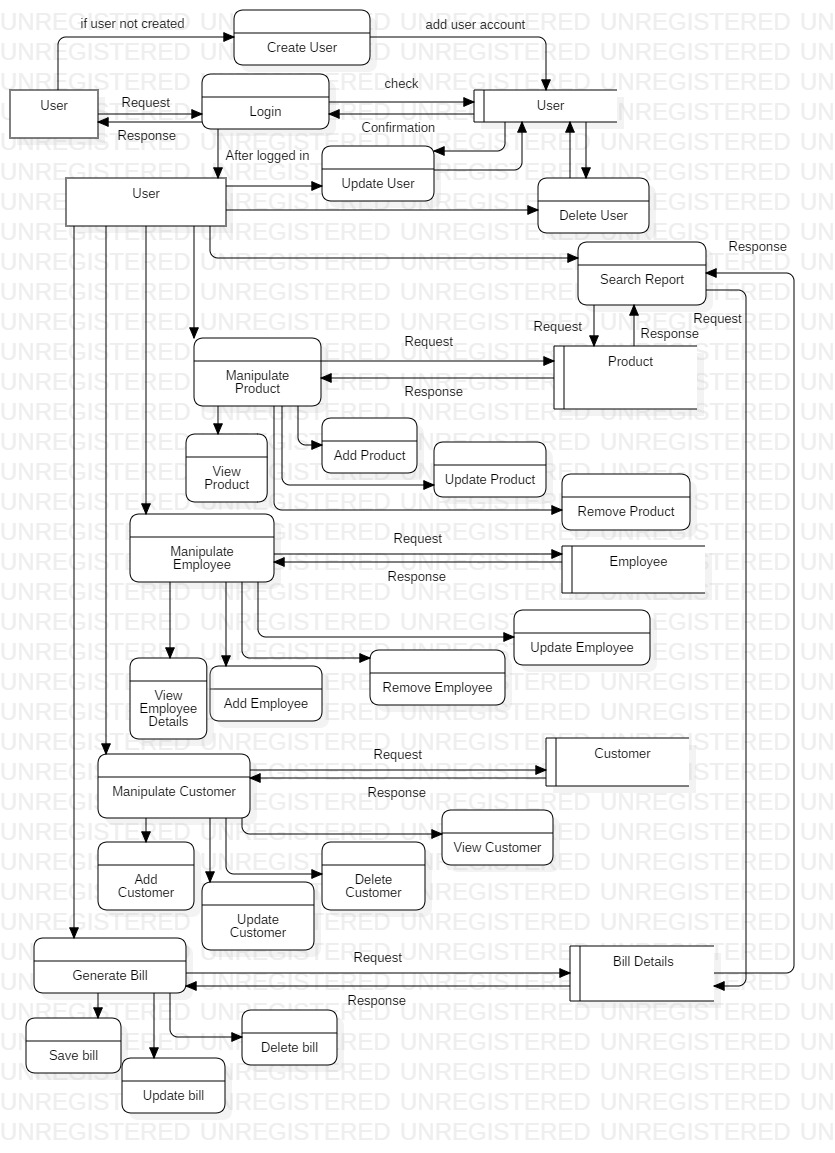


Figure 4: Level 2 Data Flow Diagram

The above data flow diagram shows the full information flow of how data, process and entity interact for the mobile shop management system. The diagrams provide the clear view of how system performs.

# Behavioral Design

## Activity Diagram

Activity diagram is a behavioral diagram that describes the flow of different activities and actions. It shows the series of actions and flow of control in a system like flow chart. Activity diagram can be used to describe every steps of use case and concurrent activities.

The reason for using activity diagram are as follows:

* For high level understanding of the systems functionalities.
* Describe the sequence from one activity to another and draw activities flow of a system.
* Describe the concurrent, branched and parallel flow of the system.
* Mainly used for modelling business requirements.

The notation used for the activity diagram of Mobile shop management system are as follows:

1. Start Symbol:



Represents the start of a process or workflow in an activity diagram.

2. Activity Symbol:



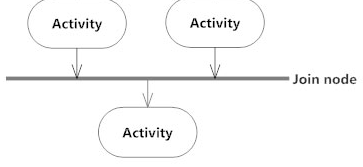
The main component of an activity diagram which indicates activities that make up a modeled process.

3. Connector Symbol:



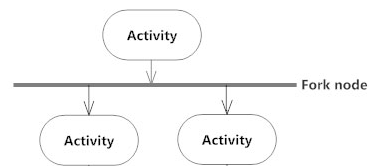
It represents the flow of one activity to another and as well as starts a step of an activity once the step is completed and flow continues with outgoing arrow.

1. Join Symbol:



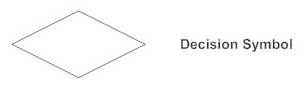
It combines the two concurrent activities and reintroduces them to a flow of one activity.

1. Fork Symbol:



It splits the single flowing activity into two concurrent activities.

1. Decision Symbol:



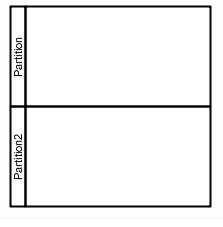
It’s a diamond shape representing the splitting or merging of various flows with the symbol acting as a container or frame.

1. End Symbol:



Represents the completion of a process or stop all control flows or object flows in an activity.

1. Swim lane:



Indicates which objects or actors are responsible for the action and it is separated by a series of line partitioning the diagram.

The activity diagram for the mobile shop management system are shown below:

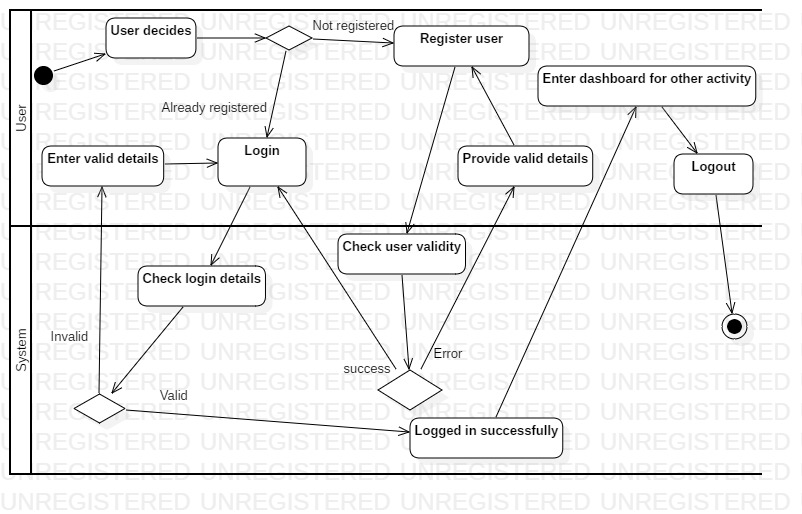


Figure 5: Activity Diagram of User Login

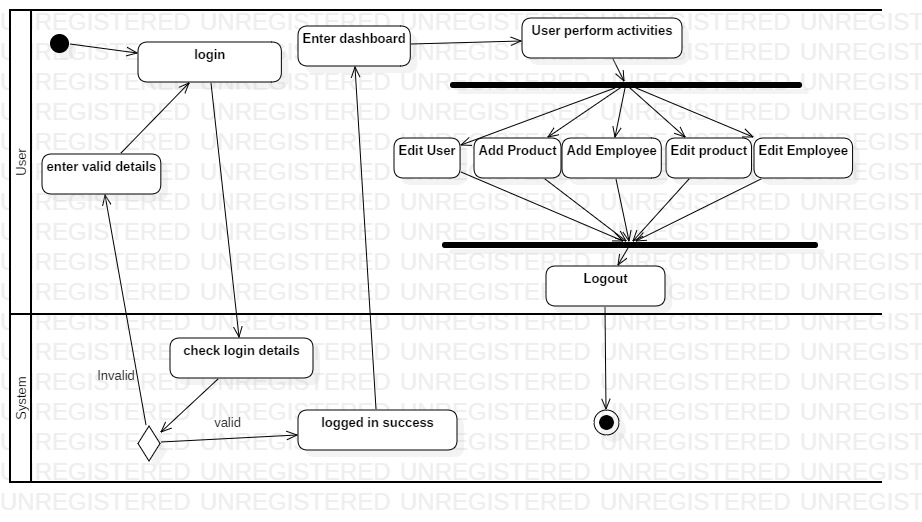


Figure 6: Activity Diagram of User

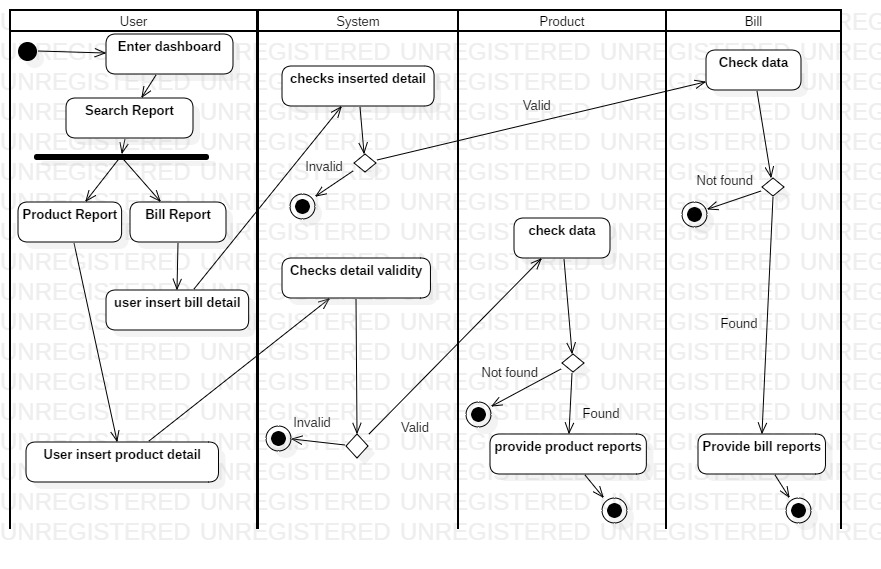


Figure 7: Activity Diagram of Searching Report

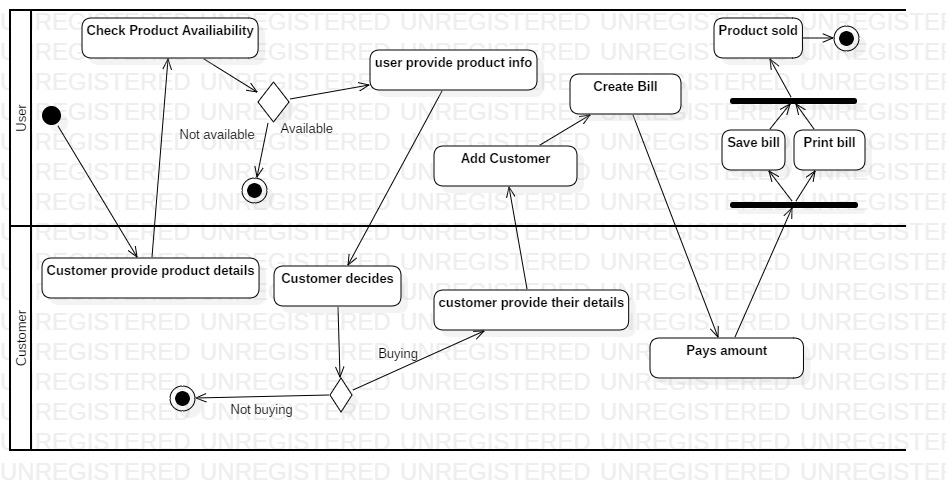


Figure 8: Activity Diagram of Product selling

The above activity diagram shows the different activity performed by user in mobile shop management system such as user login activity, user’s activity and search as well as product selling activity. It describes the use cases from use case diagram.

## Sequence Diagram

Sequence diagram describes the sequence of messages and interactions that happen between objects and actors. It represents a communication sequentially and shows how and in what order the objects in a system function. It portrays the architecture of a system and how components are interconnected.

The reason for using sequence diagram are as follows:

* It shows the details of a UML use case.
* To see how components and objects interact with each other to complete a process.
* Model the logic of a sophisticated function, procedure or operation.
* It shows the order in which methods are invoked and scope or lifetime of objects.

The notation used for sequence diagram of mobile shop management system are as follows:

1. An Actor:



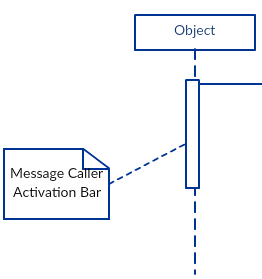
Actors are entities that are both interactive with and external to the system represented by a stick figure.

1. A lifeline:



Dashed vertical line that represents the passage of time and sequential events that occurs to an object during the charted process.

1. Activation bar:



It is the bar or box placed on lifeline and it indicates that an object is active during interaction or shows time needed to object for completing task.

1. Synchronous Message:



Represents or used when sender waits for the receiver to process the message and return before carrying on with another message.

1. Return Message:



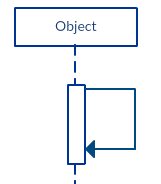
It represents a return message used to indicate that the message receiver is done processing the message and is returning control over to the message caller.

1. Asynchronous message:



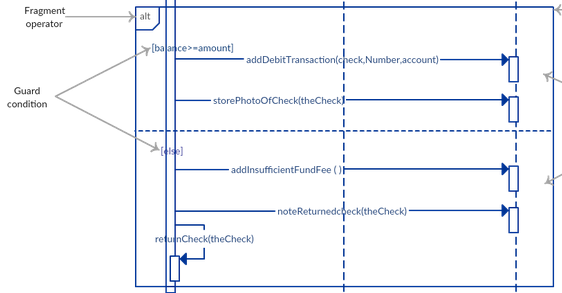
It is used when the message sender does not wait for the receiver to process the message and return before sending other messages to other objects.

1. Reflexive message:



It represents the message sent by an object to itself.

1. Alternatives:



The alternative combination fragment is used when a choice needs to be made between two or more message sequences.

The sequence diagrams of the mobile shop management system are as follows:

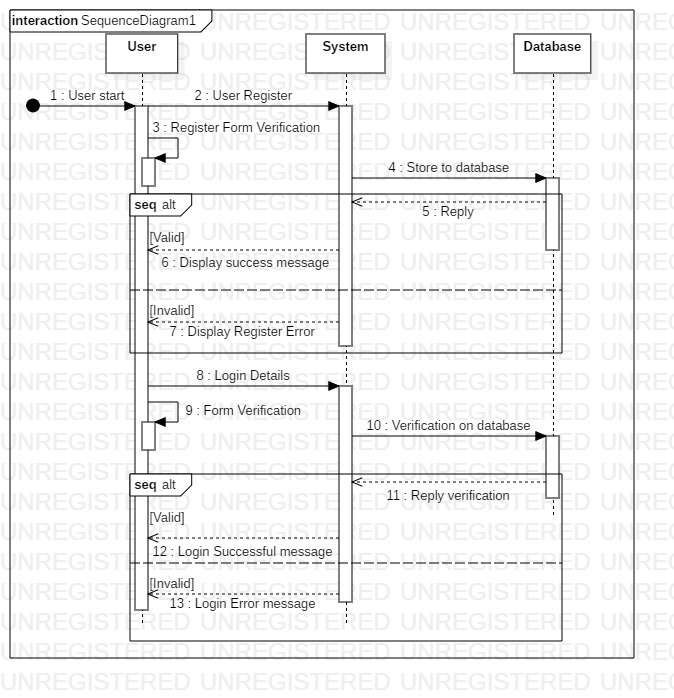


Figure 9: User login sequence diagram

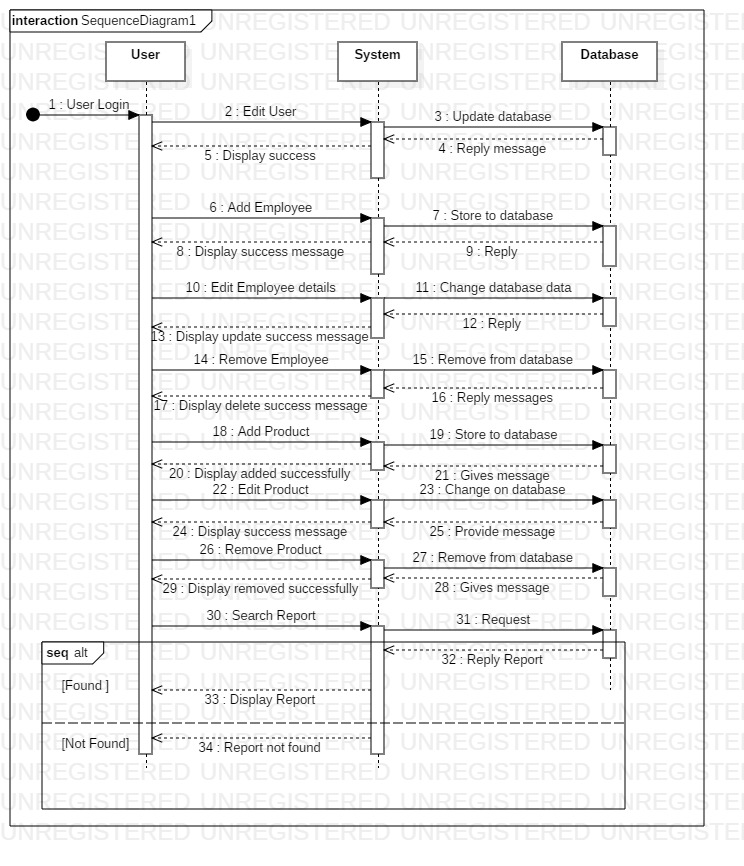


Figure 10: Sequence diagram of User

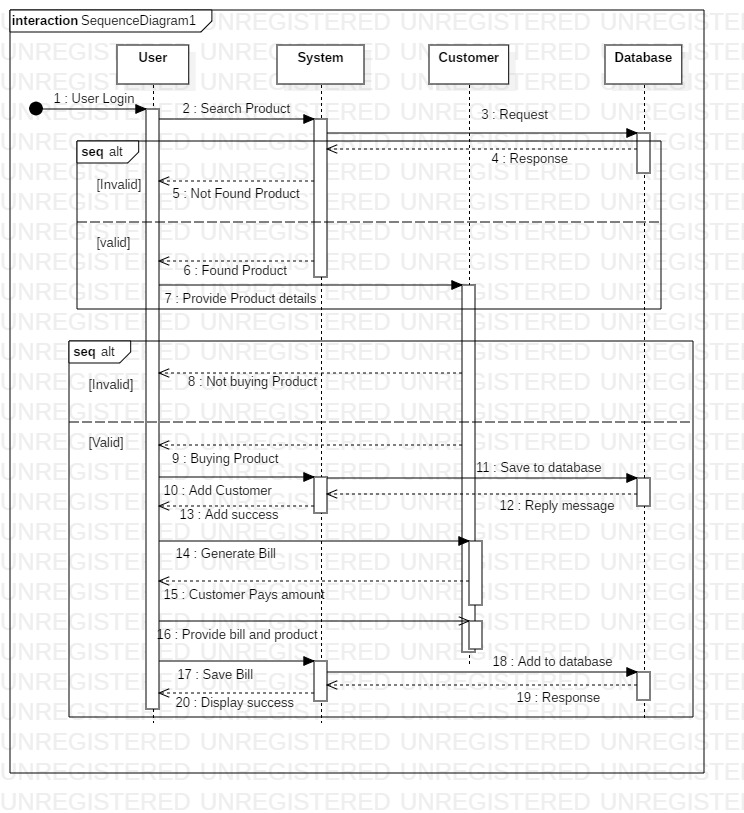


Figure 11: Sequence diagram for Product Selling

Sequence diagram created for mobile shop MS above shows the sequence of messages and interaction that happens between users and other objects of Mobile shop MS. The diagram also shows sequential order of interactions and communiation.

# Database

## Data Dictionary

Data dictionary is the table that contains the metadata of the created database. In relational database it shows the metadata such as tables, column, datatype, views and constraints keys and relationship. The data dictionary of Mobile shop management system is shown below.

The reason for using data dictionary are as follows:

* To manage the details in large scale systems.
* To communicate a common meaning for all elements in the system.
* To document features of the system.
* To locate errors an omission in the system.

Data dictionary for the mobile shop management system are as follows:

User Table

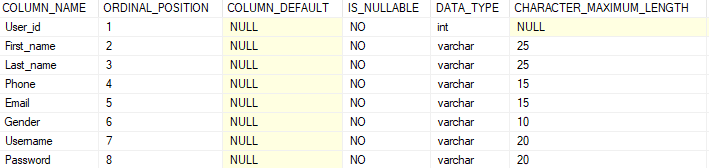


Figure 12: User table data dictionary

Employee Table

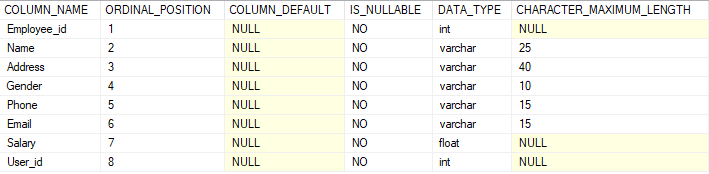


Figure : Employee data dictionary

Customer Table

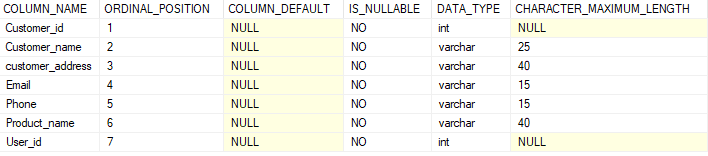


Figure : Customer data dictionary

Product Table

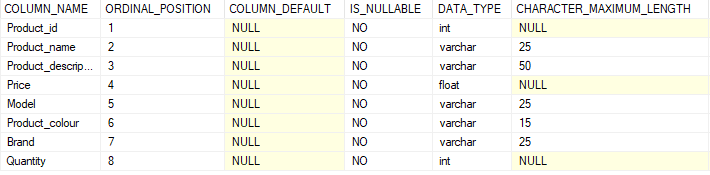


Figure : Product data dictionary

Bill Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Constraint** |
| Bill\_id | int | 10 | Not null | Primary key |
| Customer\_id | Int | 10 | Not null | Foreign key |
| Product\_id | int | 10 | Not null | Foreign key |
| Rate | Float | 10 | Not null | Null |
| Discount | Float | 10 | Not null | Null |
| Amount | Float | 10 | Not null | Null |
| Date | Date | Null | Not null | Null |
| quantity | int | 10 | Not null | Null |
| Tax | Float | 10 | Not null | Null |
| Grand Total | Float | 10 | Not null | Null |

## ER Diagram

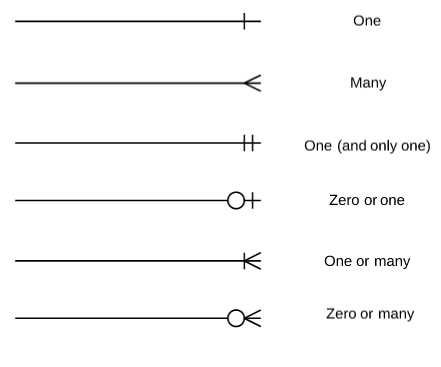
Entity relationship diagram is a high-level conceptual data modelling diagram that shows the relationships of entity set stored in a database. It is based on the notation of real-world entities and the relationship between them. It helps producing a well-designed database and it is created before implementing database.

The reason for using ER diagram are as follows:

* Describe the entities, attributes and relationships.
* ER diagrams are translatable into relational tables which helps in building databases quickly.
* It can be used by database designer as blueprint for implementing data in specific software applications.
* It provides a preview of how each table should connect and defines constraint keys and relationship.

The crow’s feet notation of ER diagram are as follows:

1. In crow’s feet ER diagram the notation are cardinality and ordinality as shown below for entities relationship.



ER Diagram for the mobile shop management system is shown below:

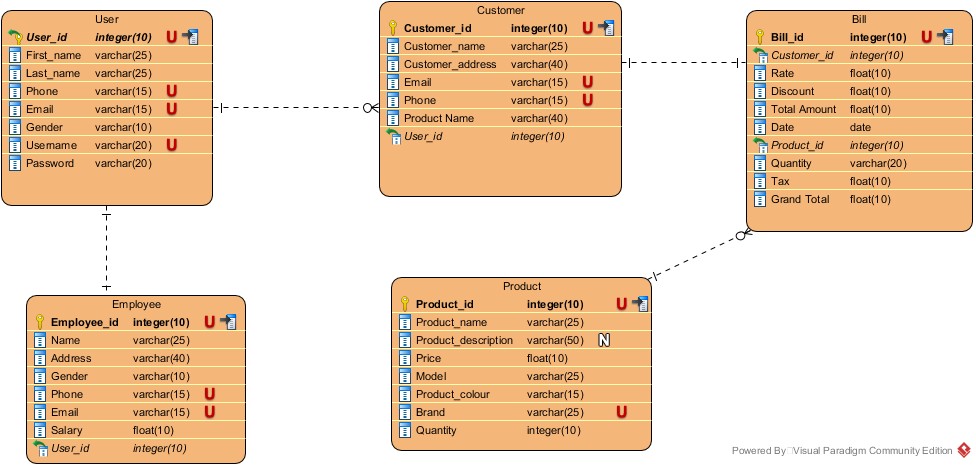
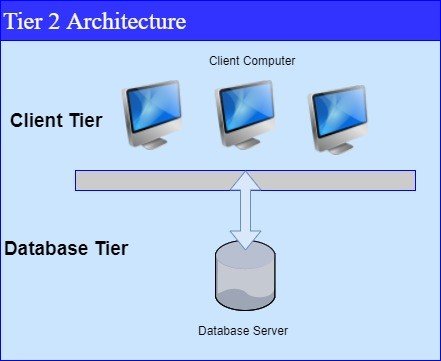


Figure : ER diagram of Mobile shop MS

# Architecture



System architecture is the structural design of systems and it is a conceptual model. System architecture serves as the blueprint for both the system and the project developing it, defining the work assignments that must be carried out by design and implementation teams. Software or system architecture consist of different tiers or layer. Since, the system I am developing is desktop-based application with server I will be using 2-tier architecture. It is divided into two parts client and database. Client system handles both Presentation and Application layers and Server system handles Database layer. The communication takes place between the Client and the Server. Client system sends the request to the Server system and the Server system processes the request and sends back the data to the Client System. It is also known as client server application.

# UI

## Prototype

Prototype is an initial rough draft of the final product. It is essential in product development as it gives rough idea to the creator how the product will be, what it will do, how it operates and can make alteration while the product is still in concept mode.

The reason for using prototype are as follows:

* It gives a proper idea of how the system layout or interface should be.
* It is easier to develop and refine the model at any time.
* It can be changed at any time from design to function since it’s a concept mode.
* It helps finding faults before real product is produced.

The digital prototype of the Mobile shop management system are as follows:

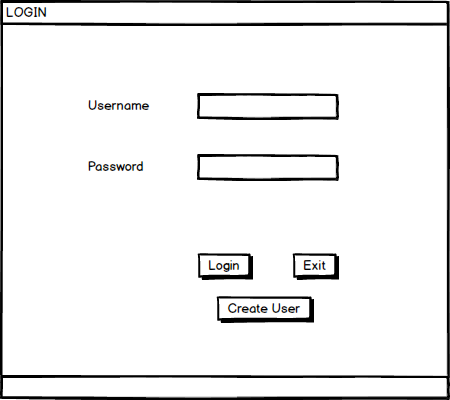


Figure : Login page prototype

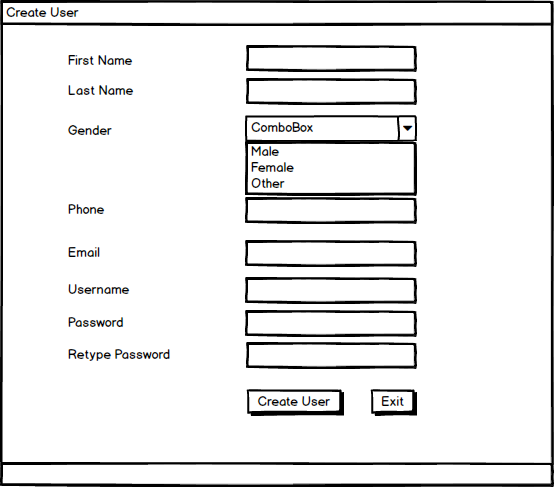


Figure : Create page prototype

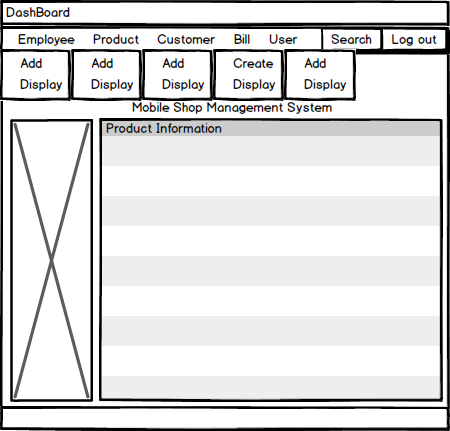


Figure : Dashboard prototype

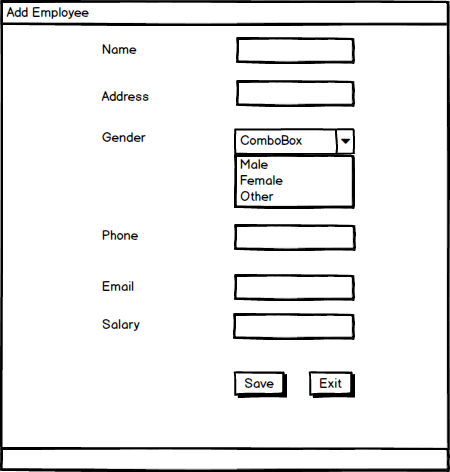


Figure : Employee add page

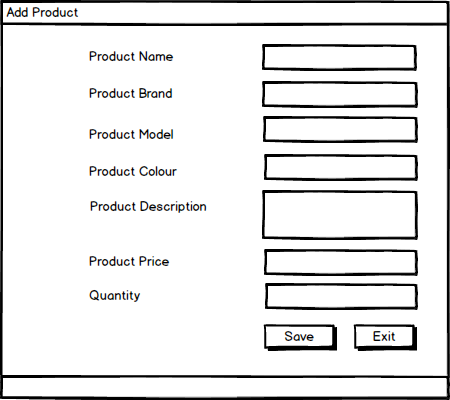


Figure : product add page

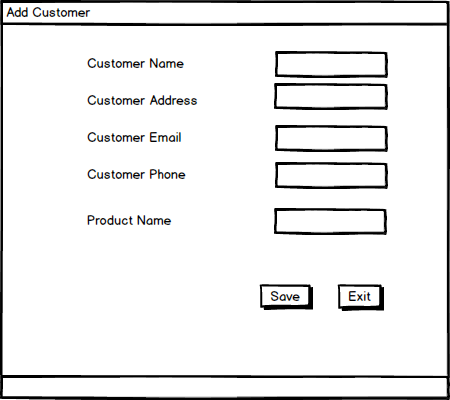


Figure : customer add page

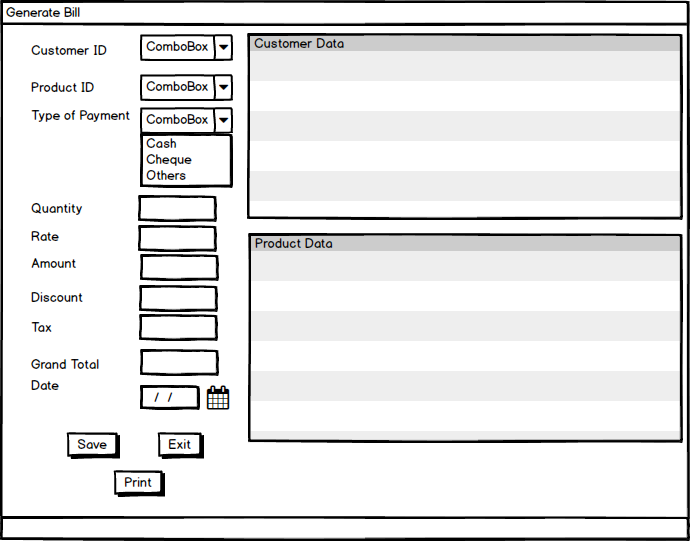


Figure : generate bill page

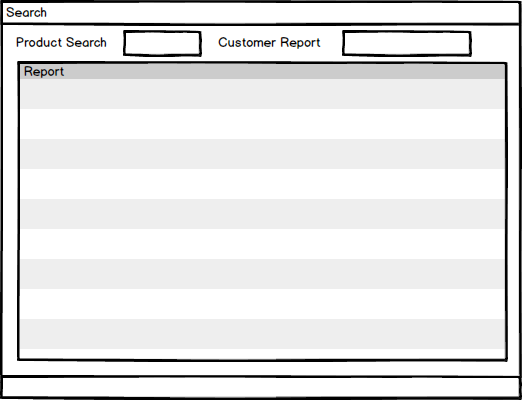


Figure : Search report page

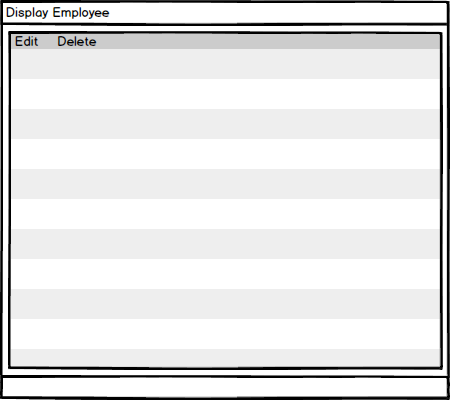


Figure : Display and edit page

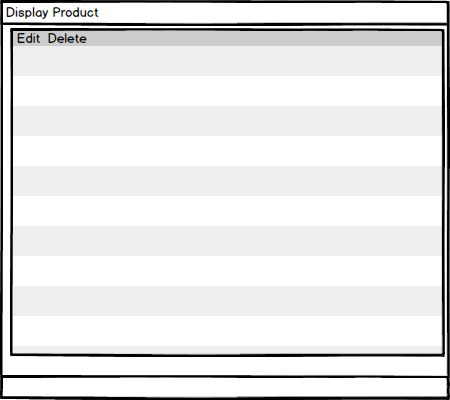


Figure : display and edit page

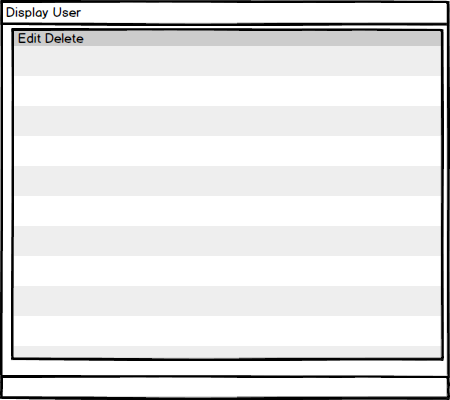


Figure : display and edit page

## 

# 