# Introduction

Design can be defined as the stage of software development where system is designed on the basis of user requirement and feasibility study. If the system is feasible enough to make then the solution must be presented considering the requirements, features wanted by the user are modelled which will help for code development in another stage.

The design stage is the most important part of software development where the logical system derived as a result of analysis is converted into physical diagram. Design is also a detailed description of what is needed to solve the original user requirement which are business activity of client.

In order to understand system from different view and create a solution different types of design pattern are used. The types of design pattern are

## Structural Modelling

Structural Modelling is the type of application modelling which deals with the class and object composition. Structural class creation pattern uses inheritance to compose the interface for the user which user uses to complete the business activity.

Structural model of the application can be shown using different kinds of diagrams and pictures. For representing my structural model class diagram and flowcharts are used.

## Class Diagram

Class diagram is the structural model of the program that shows the structure of the system by showing system’s classes along with the attributes and operations and their relationship with the objects that are instantiated the runtime of the program.

Class diagram is the main building block of object oriented modelling that later can be used to translating objects into actual code that gets run on the computer to perform the business activity.

Different notations are used to construct class diagram. The notations used in designing the class diagrams are explained below:-

1. Class Visibility

Class visibility is shown by writing characters in front of class name.

To show the visibility of class different signs are used in the diagram. Some notations used are

1. +

+ sign represents the class is public which means that class can be called from within the namespace.

1. –

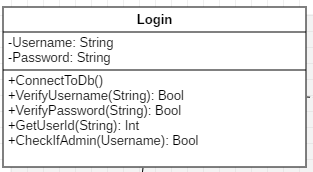
* Sign represents that the class is private which means the variables and methods are not accessible to other class.

1. #

# sign represents the class is protected which means a class is accessible within the solution inside a single namespace.

1. Class

A class contains code to manipulate data according to user requirement. The class in class diagram is shown as following inside a rectangular box with name on top row attributes on second and methods in third.



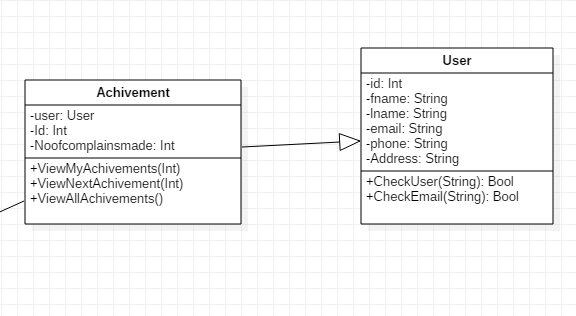
1. Class Relationship

One class may be associated with one other to perform specific functionality with in the application. The relation of two class can be of different types and some are explained below

1. Inheritance or Generalization

If one class inherits the methods and attributes of another class then relationship between them can be said as inheritance or generalization. A class can inherit properties of windows form and act like one windows form .

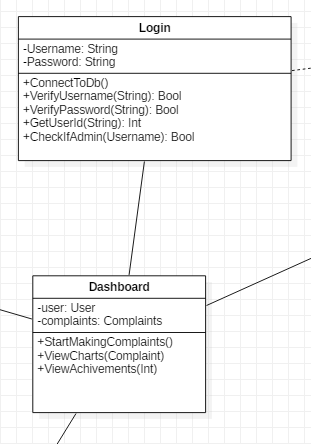
Generalization on class diagram is shown as below:-



1. Association

Association can be defined as the relation between two classes where one object wants another object to perform a service for it.

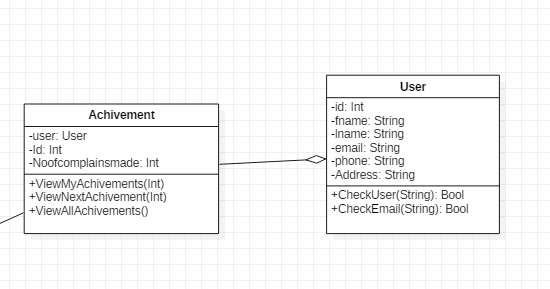
Association on class diagram is shown as below



1. Aggregation

The relationship between two classes is said to be aggregation relation if one class owns another class and owner class is not destroyed as object even if the owned class is destroyed. The aggregation function on class diagram is shown with unfilled arrowhead.

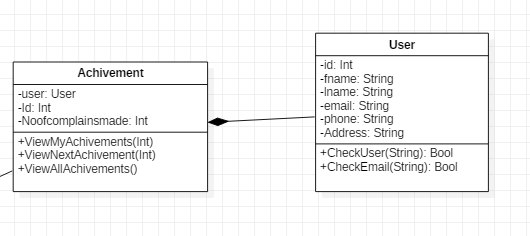
The aggregation relationship can be shown as below:-



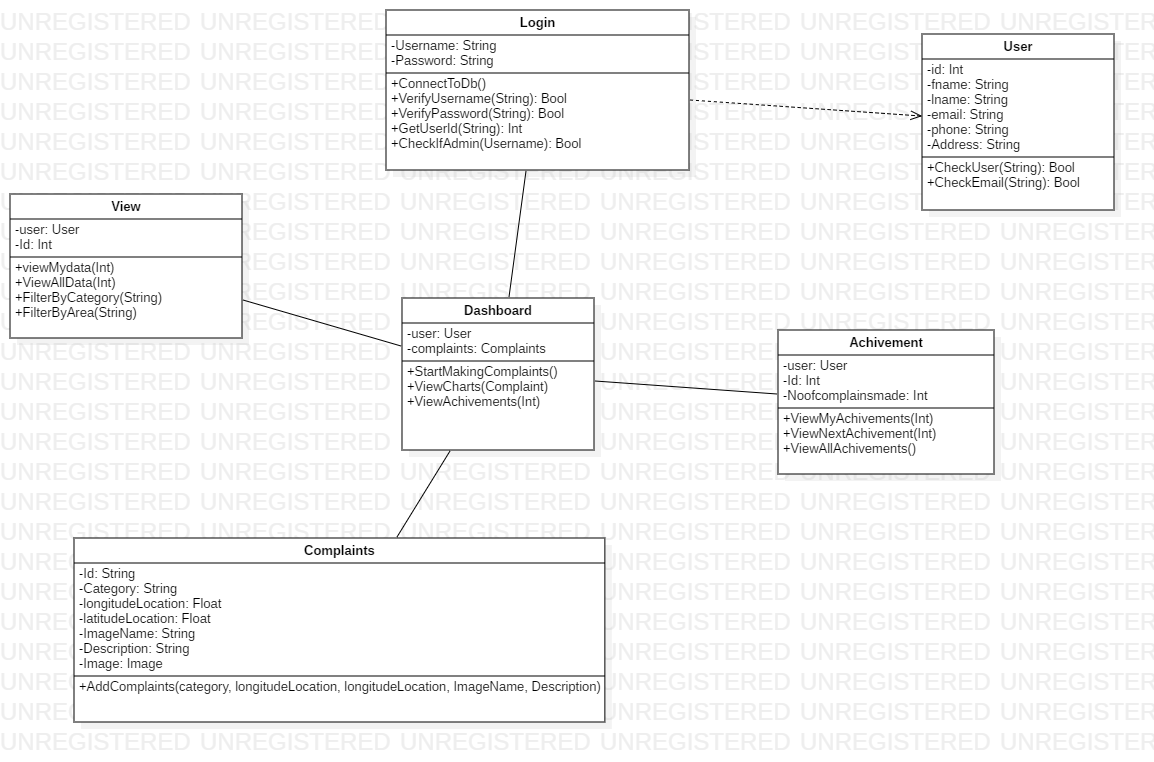
1. Composition

The composition of two class can be said as composition relation if one class owns other where child class can’t exist if owner class does not exist.

The composition in class diagram as shown below:-



The final class diagram of the application is shown as below:-



# 3.2 Behavioral Modelling

Behavioral modelling is such type of application modelling which deals with the representation of behaviors of the application with the help of diagrams to achieve a business requirement functionality which are shown on the use case. A behavioral model shows the interaction between the objects using different types of symbols and diagrams.

Behavioral modelling of the application can be done by using different types of models. Some of the modelling techniques used are explained as below:-

## 3.2.1 Activity Diagram

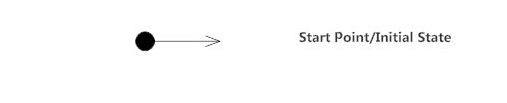
Activity diagram can be defined as the type of UML diagram which explains the dynamic aspects of the system where the flow of the program is explained from one activity to another activity.

Activity diagram shows the flow of activity from one activity to another activity. Different notations used in creating the activity diagram are explained below:-

1. Start Point

Start point shows the start of the activity within the application. Start point on UML diagram is represented by a filled circle with attached arrow which shows the direction of flow.

Symbol used to explain start point is shown below



1. Activity

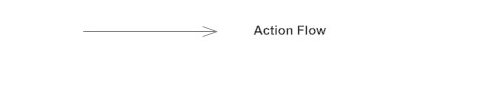
Activity is the action state of the system where it follows the direction on basis of what is to be done to the data. Activity in UML diagram is shown using a rounded edged rectangle.

The activity is shown below:-



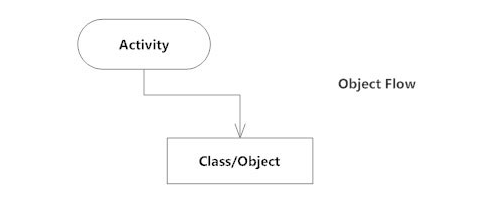
1. Action Flow

Action flow shows the transition of the process from one to another. Action flow is represented with arrowed line. Example is shown below in diagram



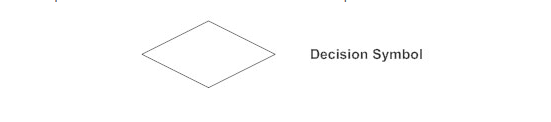
1. Object Flow

Object flow can be defined as the creation and modification of the objects by the activity to perform an action. Object flow is shown in image below:-



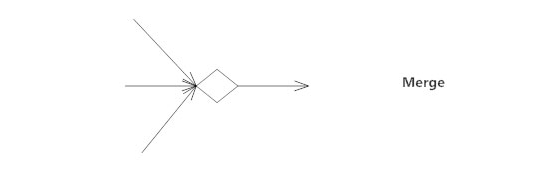
1. Decisions and branching

During the flow of the program if the system needs to perform some action which drives them to different actions on the basis of results. Decision in UML is shown using diamond and arrows from the diamonds which leads them to different directions. Decision is shown in image below



1. Merge Event

Merge event is the condition where two parallel process meet with one another and they have the flow of activity from one to another. Merge event is represented by different action flow meeting at a point. The image of merge event is shown below:-

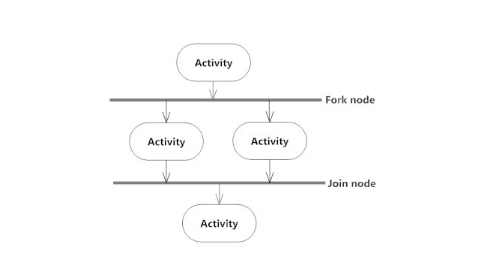


1. Join and Fork Node

Fork node splits one activity into many flows to different directions whereas join node assembles different activities to one single activity.

Join and fork nodes are also called synchronization because nodes assembles and disassembles single activity to many or vice versa.

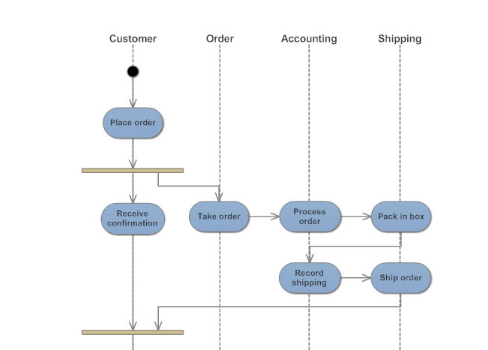
Join and fork nodes are shown in the image below:-



1. Swim Lanes

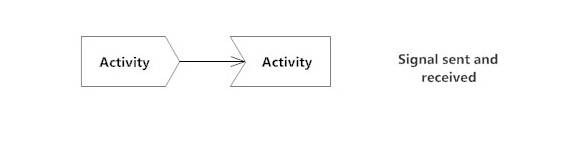
Swim lanes shows the job sharing and responsibilities for sub process of a business process. Swim lanes show the accessibility of one class with the other where messages are send and received from one to another.

The figure of swim lane is shown below



1. Send and receive signals

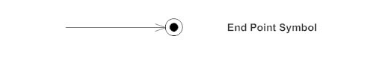
Send and receive signals are used to show message sharing functionality within the application. The signal must be received in order to move to the next activity. Symbols used to show message flowing from one to another are shown as below:-



1. Final State of end point

Final state is the end of the activity which is shown with a circle inside another circle filled.

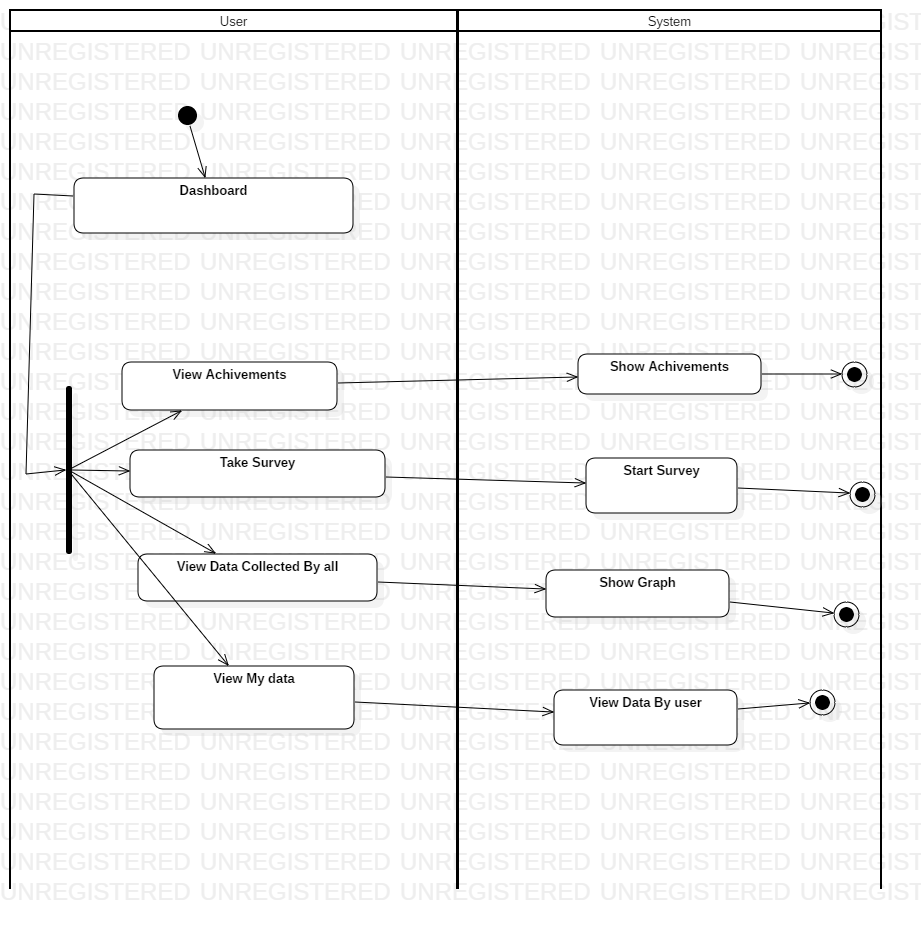
The final state is shown in the image below.



The actual activity diagram for the application is shown and explained below







## 3.2.2 Sequence diagram

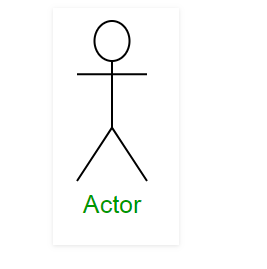
Sequence diagram is behavioral diagram of the application which shows the interaction between the objects in sequential order in which the program flows. The order in which objects interact with each other to fulfill a functionality that is being integrated in the application is a sequence or order of execution. Sequence diagram describes the order of execution in the application.

Different symbols are used in the diagram to show flow of program. Some of the symbols used in the sequence diagram are described below: -

1. Actor

Actors are the users of the application who gives instructions to the system to perform the business activity. Actor interact with objects and interface of the system. Actors are outside the scope of application. Many users can be added in the sequence diagram to represent different level of users.

Actor is a stickman in the sequence diagram. The symbol of actor is shown below

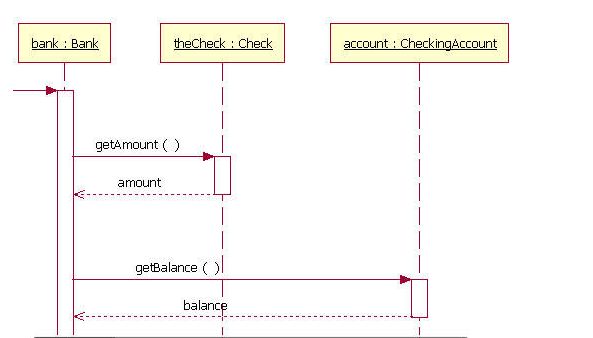


1. Lifeline

Lifeline are drawn as a box with dashed line from the center of class which represents either role or object instantiation that participates in the sequence being modelled.

Lifeline shows the accessibility of class and message sharing from one class to the other.

The lifeline symbol is shown in image below

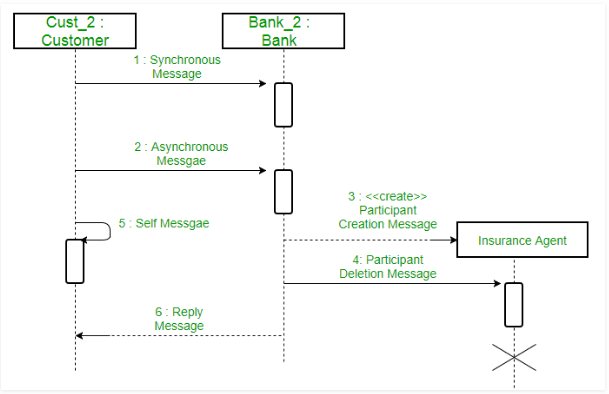


1. Messages

The application consist of different classes where they have to share message internally to fulfill the functionality. The messages appear in sequential order on the lifeline. Messages are communication protocols between two or more classes. Some types of messages are

* Synchronous Message

Synchronous messages are those type of messages where reply must be got from other classes before executing another activity. The symbol used in synchronous message is shown in image below:-

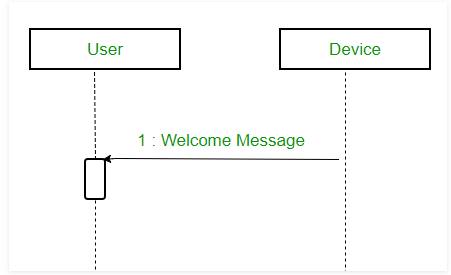


* Asynchronous Message

Asynchronous message is the type of messages shared between two classes where the reply is not to be waited before executing another activity.

Lined arrowhead is used to show this type of message in sequence diagram.

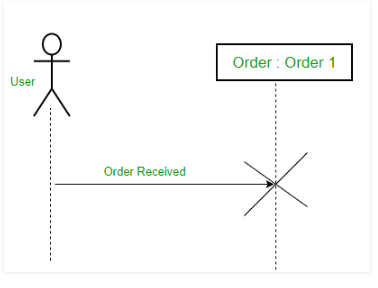
Following picture shows asynchronous message



* Delete Message

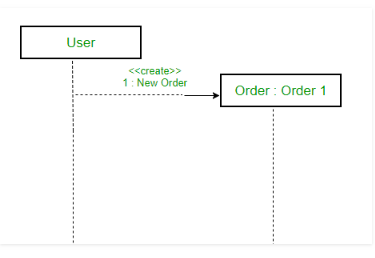
Delete messages are sent to delete the objects instantiated during the execution of the program. Delete message deletes the objects in the memory if occurrence exists.

Delete message Is shown using dashed line terminating at ‘x’. Delete message is shown in the figure below-



* Create Message

Create message is used to instantiate the object of class in the program. Create message allocates space in the memory for the instantiated object. Create message is shown in the sequence diagram with dashed line with arrowhead and create labelled above the line. The symbol of create message is shown below:-



* Reply Messages

Reply messages are the messages sent from the class to another in response of initial message sent. Reply messages are shown using the dotted line with arrowhead from one swim lane to another.

The symbol of reply message is shown below



* Lost Messages

A lost message is used on such cases where the recipient is not known to the system. It is represented using an arrow directed towards an end point from lifeline.

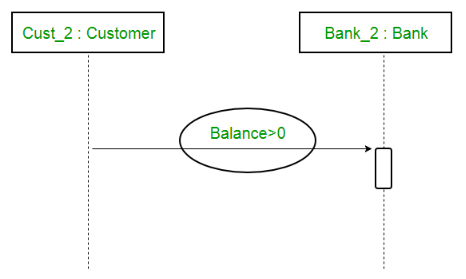
The symbol is used to represent lost message is shown below in the diagram



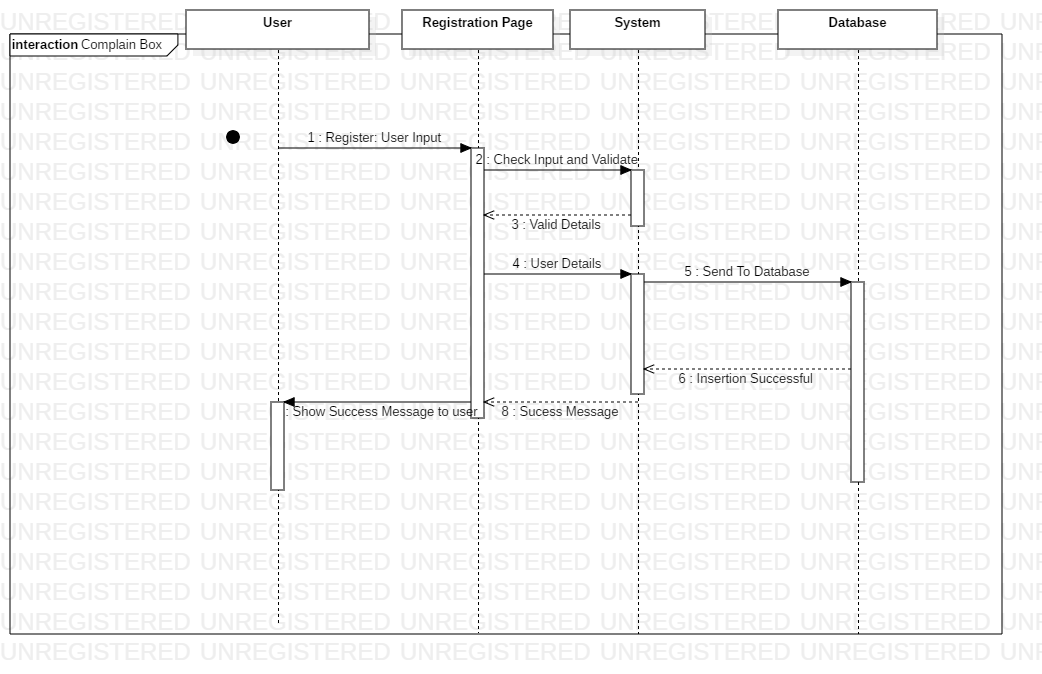
1. Guards

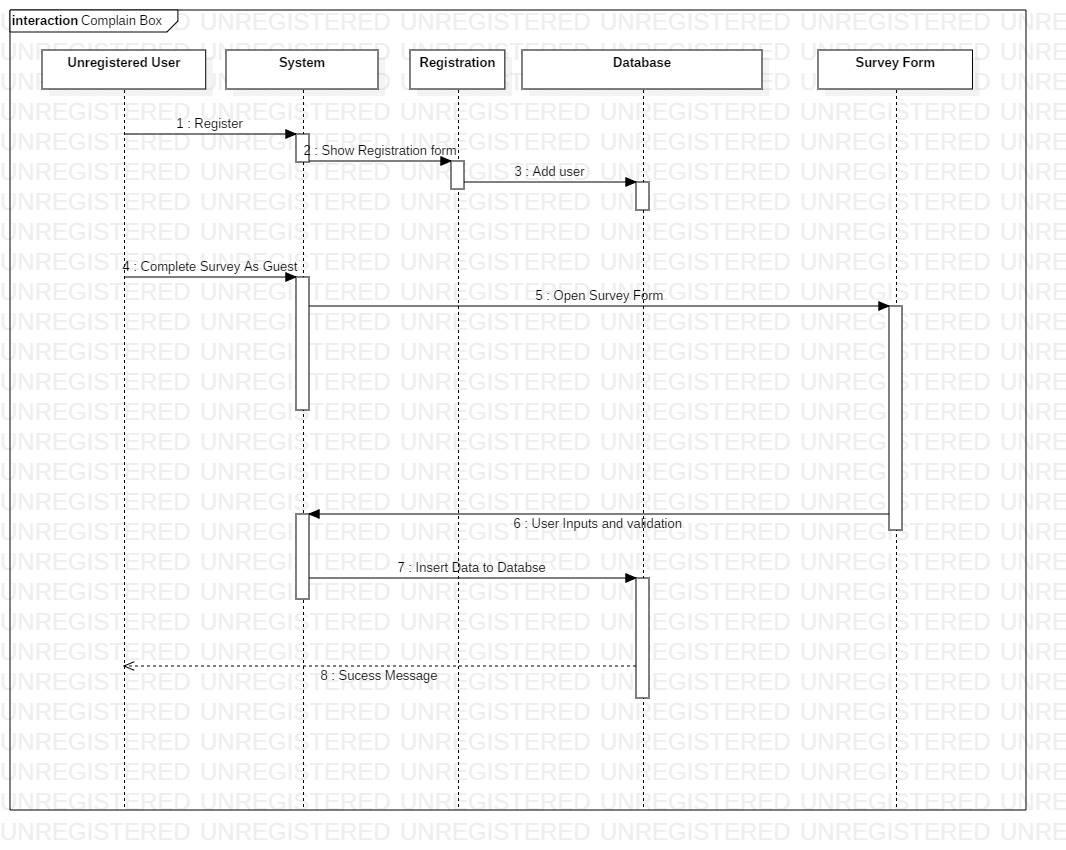
In order to model conditions guards are used in the sequence diagram. Guards play an important role in letting software developers know the constrains attached to the system or particular activity.

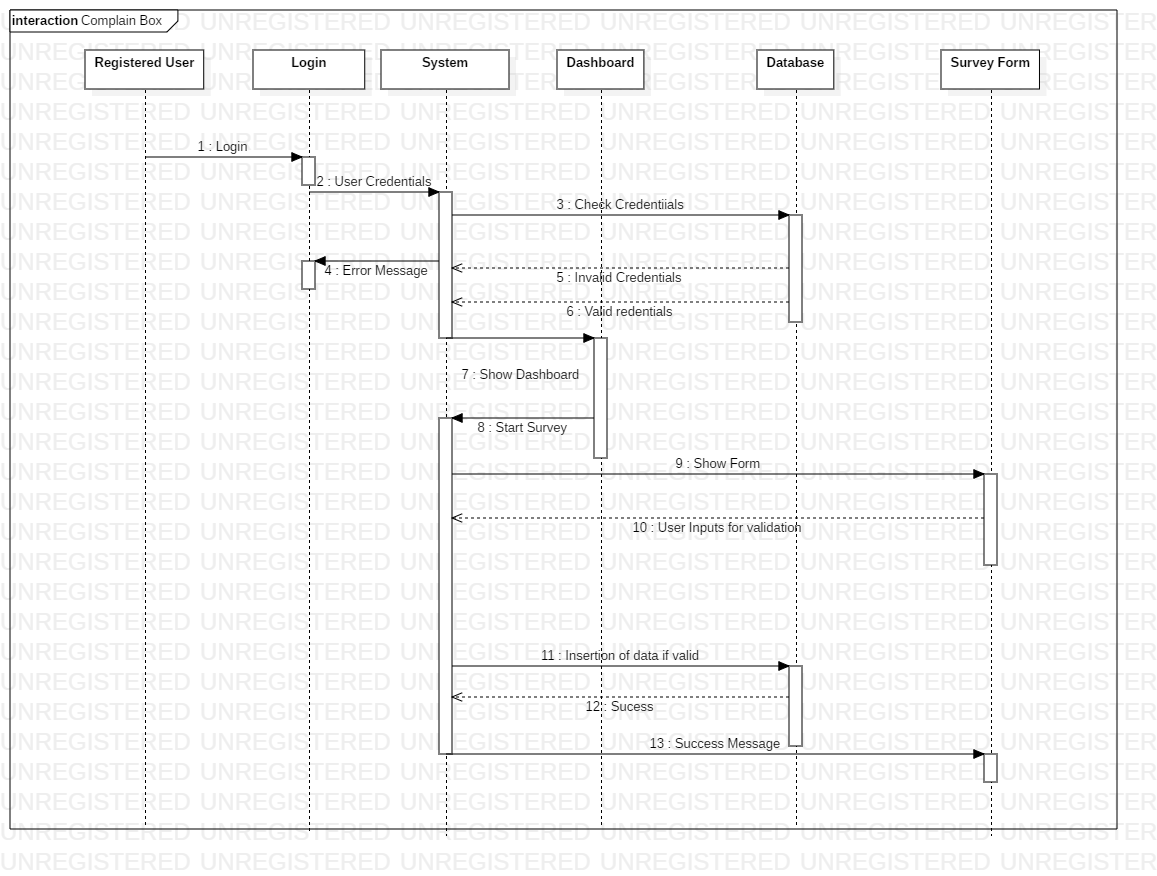
The guards are shown in image below:-



The sequence diagram for the system is shown and explained below:-







# Database Modelling

Database modelling is a technique of modelling the database that determines logical structure of database and fundamentally determines in which manner data can be store organized and manipulated.

The database for this application is modelled using object-oriented manner so that objects can be managed easily and data are stored in the database.

The object-oriented database model is best known post-relational database model since it incorporates tables but is not limited to tables.

## 3.3.1 Data Dictionary

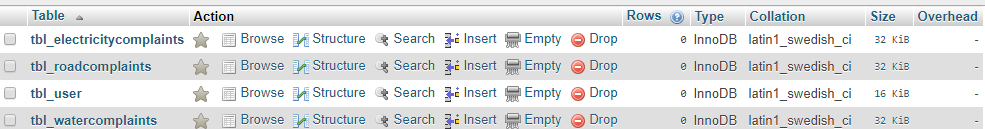
Data dictionary can be defined as the metadata of the tables in the database. Data dictionary contains the properties of the tables in the database.

Data dictionary provides detailed information about the business data such as standard definitions of data elements their meanings and allowable values.

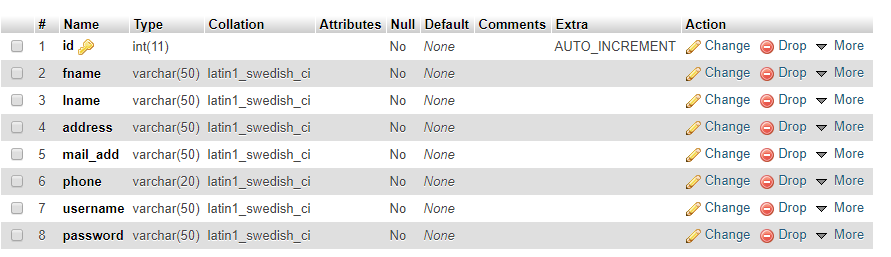
The database of this system is modelled with object-oriented approach so that objects can be managed properly.

The data dictionary of the database is shown in table below

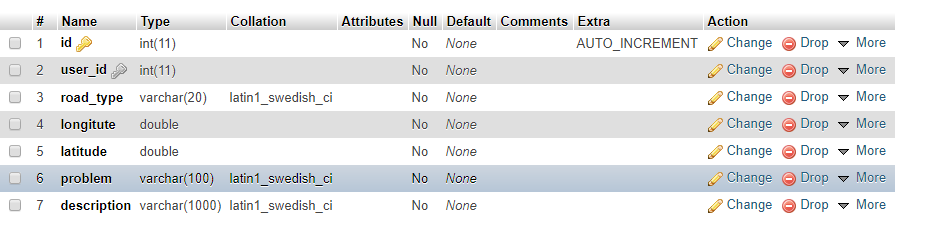
The database contains four tables for the storage of data. The tables in database are shown below



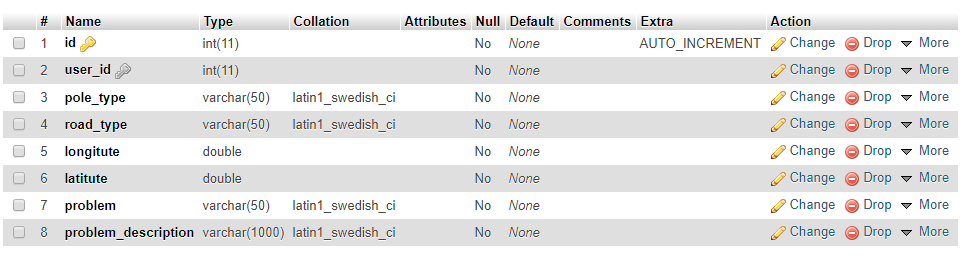
The data dictionary of tbl\_user is shown below in diagram



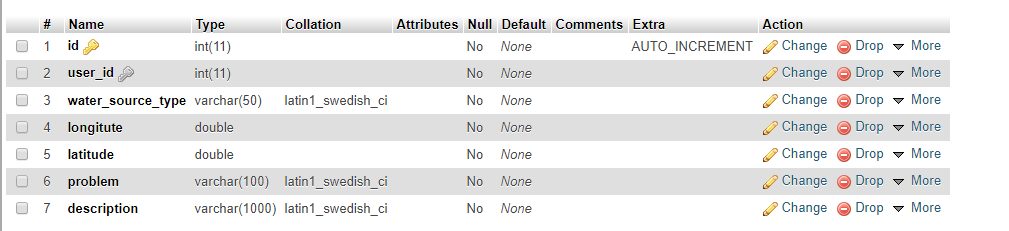
The data dictionary for tbl\_roadcomplaints is shown below in image



The data dictionary of the table tbl\_electricitycomplaints is shown below:-



The data dictionary for tbl\_watercomplaints is shown in table below:-

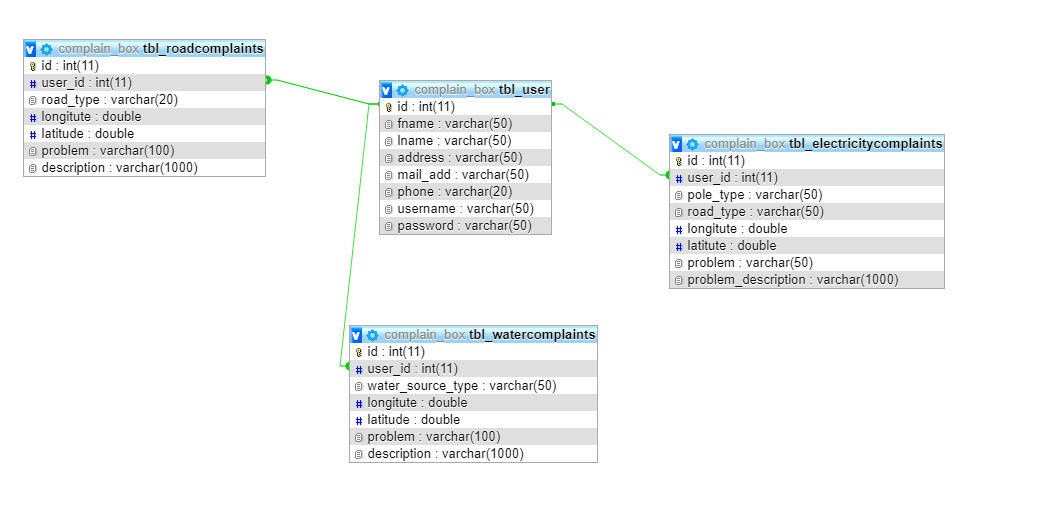


## 3.3.2 ER Diagram

ER diagram can be defined as the structural diagram of the database showing tables and attributes to use in database creation, management and maintenance.

ER diagram helps later in producing the real database for the company or the organization.

ER diagram of the system above is shown in image below:-



# UI Modelling

UI modelling is the technique of designing user interface for the user so that user can get the best easy possible solution to interact with the system. UI plays important role in the usability of the application.

Good interface is a non-functional requirement of the application so that user can have best experience with the system. UI of the system must be kept as easy as possible for maximum usability within the application.

## 3.4.1 Prototyping

Prototyping is the technique for creating the UI of the system being developed on other medium which can be kept as reference for the development of the application.

Prototypes are rough diagrams that can be changed for modelling the application.

The prototypes developed for this application are shown below:-

