# Chapter 3: Designing

## 3.1 Introduction to design

Design phase plays an important role to build a system. Here we can get the answer of “how” questions, arises during the development of system. GUI part and database design of the system development covered by this phase. To demonstrate the structure and workflow of the system, that is being developed I have used different models like structural model and behavioral model. It gives us an overview of the frontend and backend of the system. Design phase involves different diagrams so that non- technical user can know more about the system. The main purpose of the design phase is to transfer the user requirements into some acceptable form.

# 3.2 Structural Modelling

## 3.2.1 Final Class Diagram

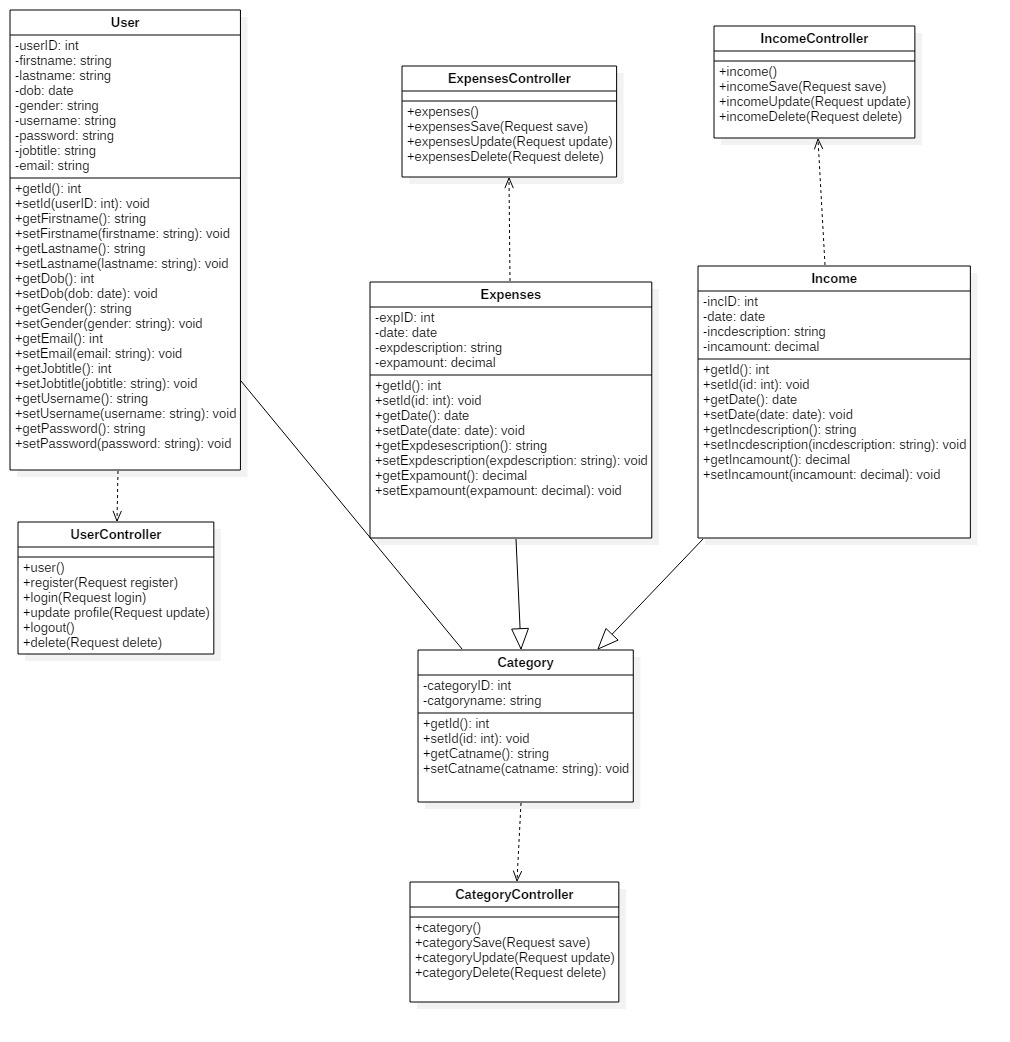
In UML modelling, class diagram shows the relationship between classes, their attributes and operations. Class diagram is the backbone of an OO modelling and referred as blueprint of the system. It is the exception in UML Modelling that class diagram directly mapped with any object-oriented programming languages.

**Justification:**

I have chosen class diagram in comparison to other structural diagram because class diagram visualize, describe and document the different aspects of system. Class diagram is the only UML diagram that can directly mapped with object-oriented languages and it can be widely used at the time of construction.

**Notation used to draw Class Diagram:**

|  |  |  |
| --- | --- | --- |
| Notation Used | Name | Description |
|  | * Top row represents the class. * Second row represents the attributes. * Third row represents the operations. | We have given the name of class in first row, their attributes in second row and operations in third row. |
|  | Generalization | It shows the parent and child relationship called inheritance relationship in object oriented. |
|  | Association | It shows the static relationship between classes. |
|  | Dependencies | It shows the relationship between super class and sub-class. |



**Explanation:**

In design phase, final class diagram of expenses management system has created according to the MVC pattern. In MVC pattern, model is a non-visual object that holds all the data and behavior other than utilized for the User Interface (UI) design. Display of the model (i.e. data or information and behavior) in the User Interface represents View. Changes made to the data or information managed by third member called controller. Above class diagram shows the specific operations with their controller and view.

## 3.2.2 Context Diagram

Context diagram is the part of structural modelling which shows the overview of the complete system. The main purpose of this diagram is to show the expecting input and output values from the system, to and from various external entities. In the context diagram there is only one process to represents for entire system.

**Justification:**

I have chosen context diagram in comparison to other structural diagram because categories of context diagram is unique to other, describe the input, and output process of the system.

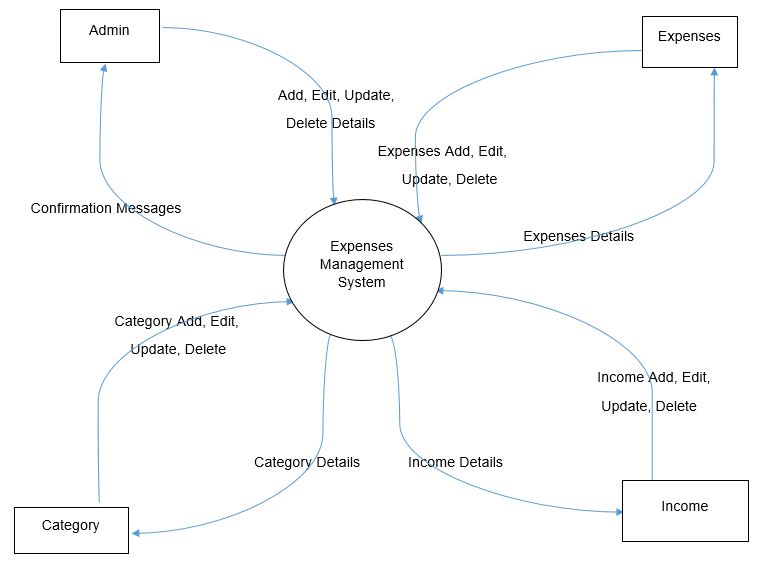
Categories of the context diagram include following points:

* Active: Dynamic to accomplish some objectives or reason
* Passive: represents external entities and interaction of those entities with the system
* Cooperative: to release some ideal result with the use of some external entities by the system
* Autonomous (Independent): Autonomous represents those external entities that are being separate from the system and indirectly affect the system by means of forced requirements.

**Notation used to draw Context Diagram:**

|  |  |  |
| --- | --- | --- |
| Notation Used | Name | Description |
|  | External Entity | Input data or retrieve data into and from the information system. |
|  | Processes | Action takes place on data and turning it into information. |
|  | Flow Line | Represents the flow of data (i.e. data is being sent or retrieved) |

**Context Diagram:**



**Explanation:**

Above context diagram represents the overview of the entire system. Admin can easily access into the user account and manipulate their data and returning confirmation message by providing what type of action take place. In case of expenses, income and category we can add, edit, update and delete data of those entities provide returning details message.

## 3.3 Behavioral Model

## 3.3.1 Activity Diagram

In Unified Modelling Language, activity diagram is another important diagram. It describes the dynamic behavior of the system by showing the workflow from one activity to another and also known as operation of the system.

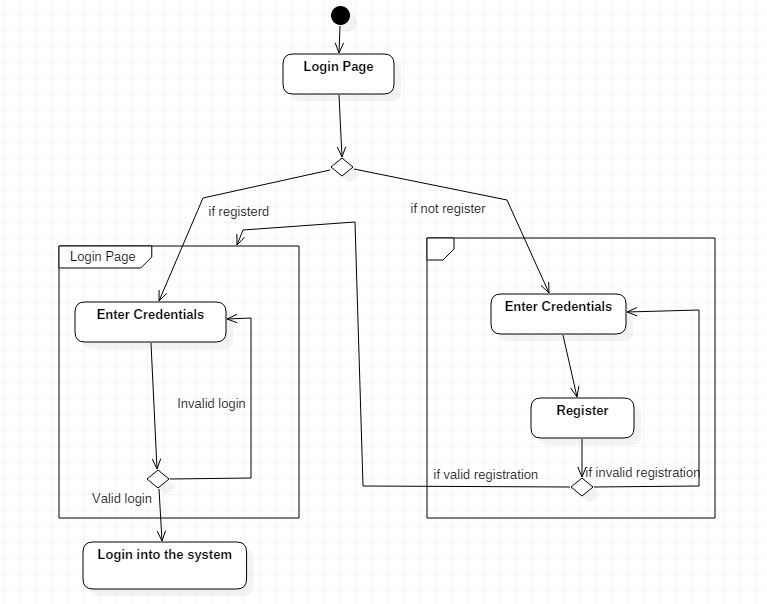
**Justification:**

I have chosen activity diagram in comparison to other behavioral diagram because it shows the flow of activity and describes the sequence of activity from one to another. The main purpose of this diagram is to show the flow of system i.e. parallel, branched and concurrent.

**Notation used to draw Activity Diagram:**

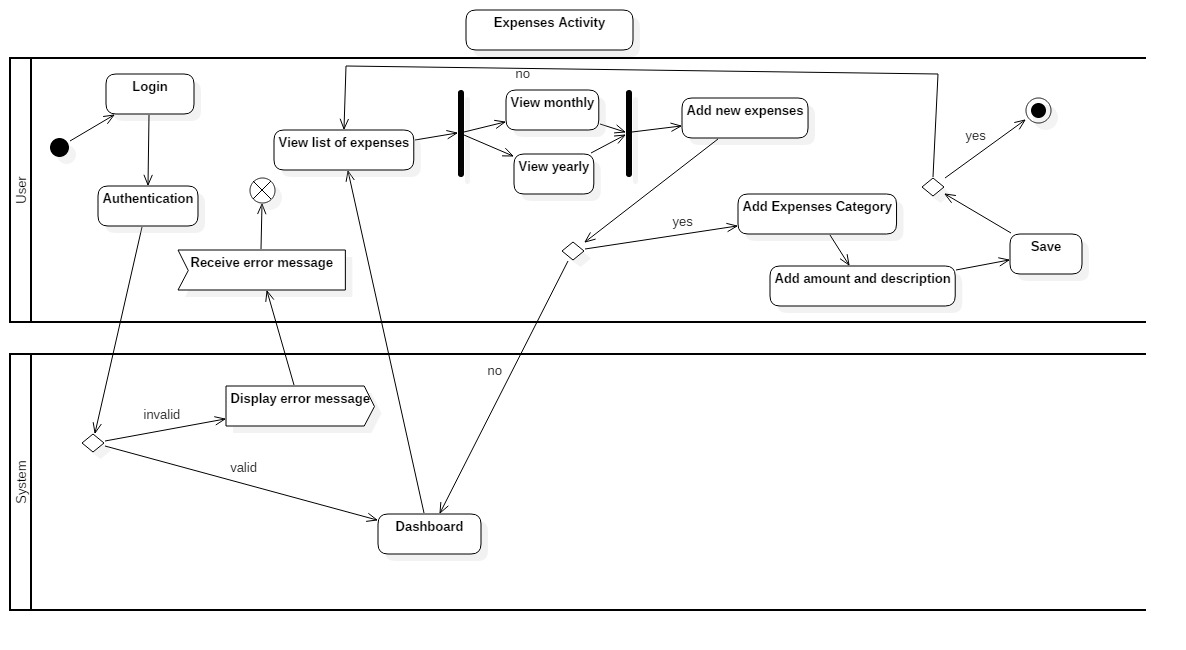
|  |  |  |
| --- | --- | --- |
| Notation Used | Name | Description |
|  | Initial Node | Represents the initial action state. |
|  | Final Node | Represents the final action state or system. |
|  | Activity or Action State | It shows the activity i.e. actions of object. |
|  | Action Flow | It shows the path from one activity to another. |
|  | Decision and Branching | It represents the decision with alternate path. |
|  | Fork Node | It represents the split of single incoming flow into multiple flows parallely. |
|  | Join Node | It helps to join the multiple flows and back them into single flow. |
|  | Sent and Received Signals | Here signal must sent by sent signals and accepts by accepting signals of activity. They usually occurs in pair. |
|  | Flow Final Node | It represents the termination of a flow. |

**Registration Activity Diagram:**

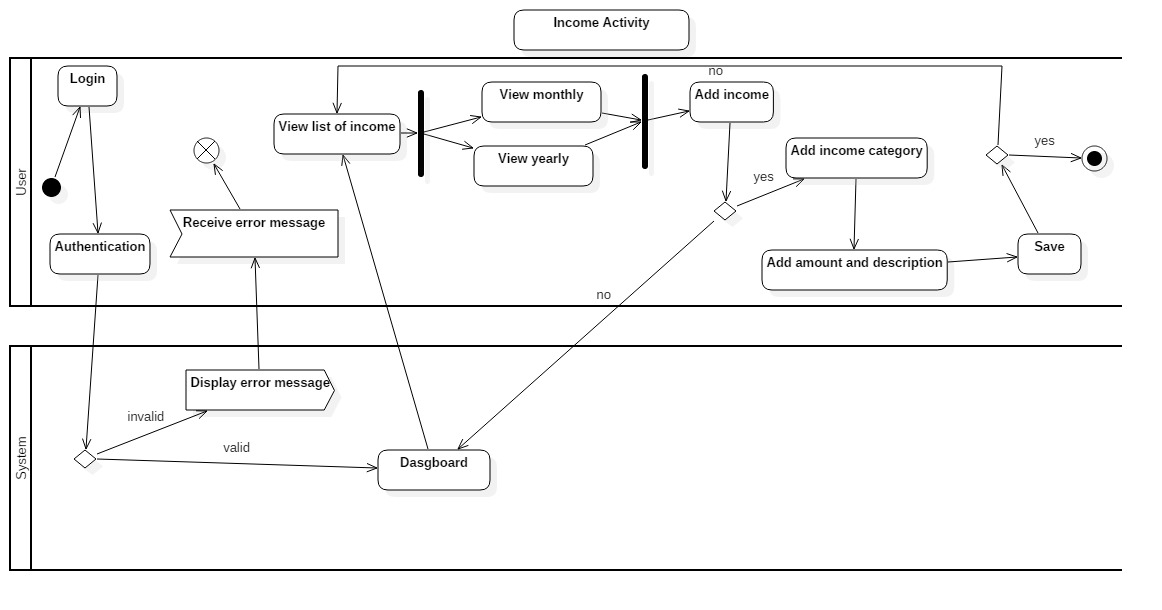


Above diagram shows the activity diagram of registration and shows how to login into the system. User must be registered to use the system i.e. sign in into the system otherwise user cannot get access int the system.

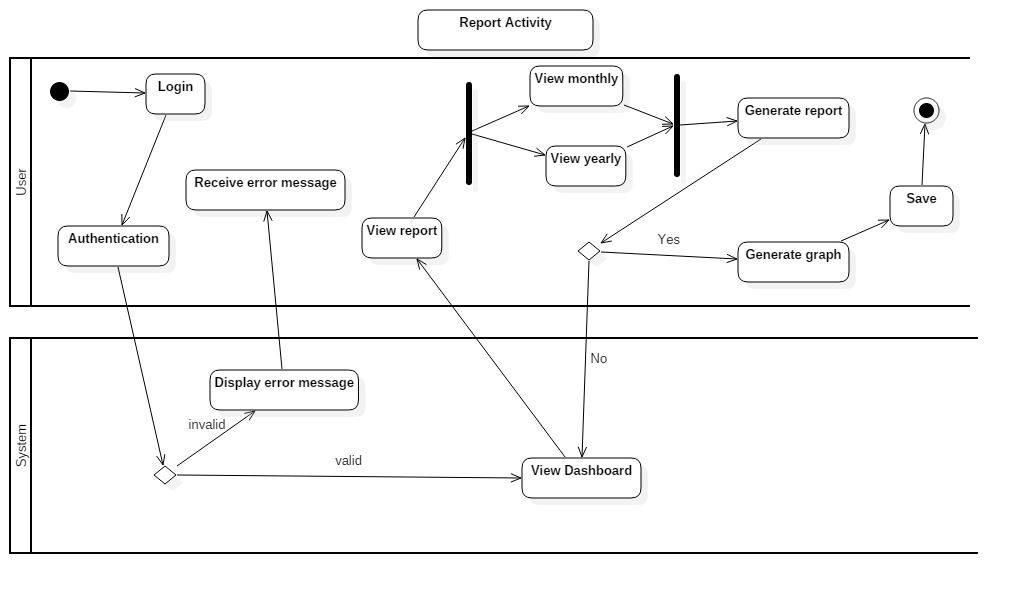
**Expenses Activity Diagram:**



**Income Activity Diagram:**



**Report Activity Diagram:**



**Explanation of the System Activity Diagram:**

Above diagram (i.e. expenses, income and report activity diagram) shows, the complete flow of activity occurs in a system. After user get registered user can easily access into the system. Then expenses activity of user shows the list of expenses, add expenses, category of expenses and income activity shows the list of income, add income, category of income. Report generated after the activity of expenses and income. Here we can generate monthly and yearly report.

## 3.3.2 Sequence Diagram

Sequence diagram is the special type of interaction diagram, which shows the message communication between various object. The dynamic behavior of the system has captured by sequence diagram.

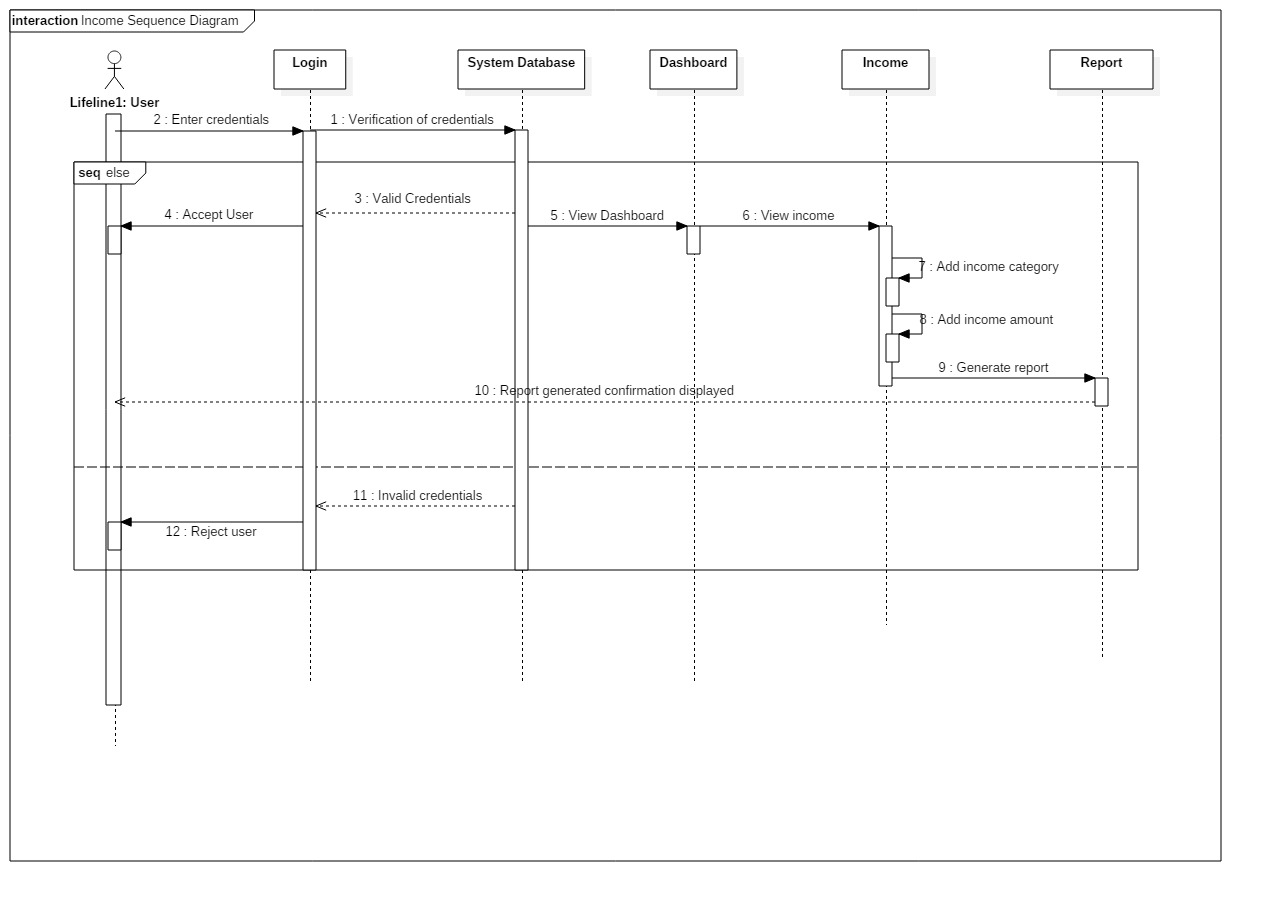
**Justification:**

I have chosen sequence diagram in comparison to other behavioral diagram because this diagram helps to describe message flow in the system, structural organization of the objects and interaction among objects.

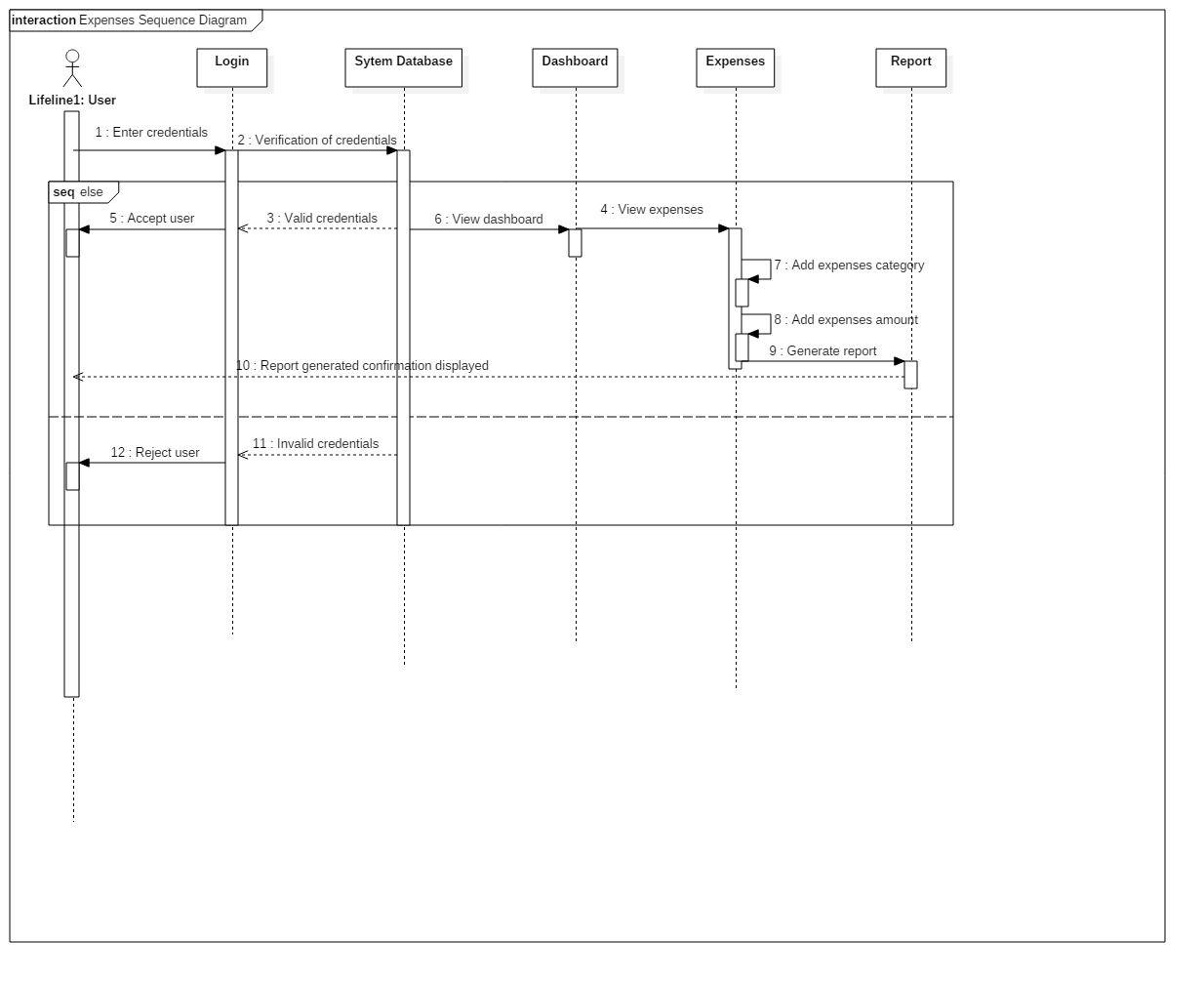
**Notation used to draw sequence diagram:**

|  |  |  |
| --- | --- | --- |
| Notation Used | Name | Description |
|  | Object | It illustrates class roles but don’t list object attributes |
|  | Activation Boxes | It represents the time of an object needs to complete a task |
|  | Lifeline | It is the vertical dashed lines that denotes the object’s presence over time |
|  | Messages | * Message are arrows that represents communication between objects. * A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline |
|  | Self-Message | A message an object send to itself, usually shown as a U shaped arrow pointing back to itself. It represents recursion. |

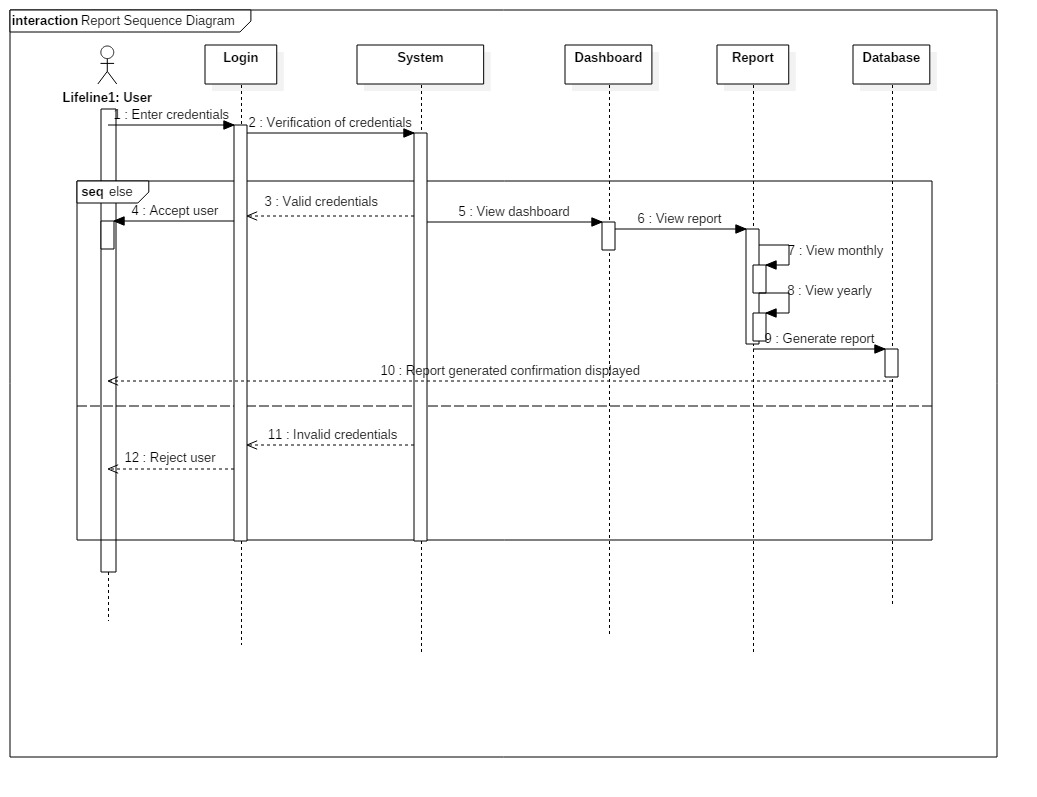
**Income Sequence Diagram:**



**Expense Sequence Diagram:**



**Report Sequence Diagram:**

****

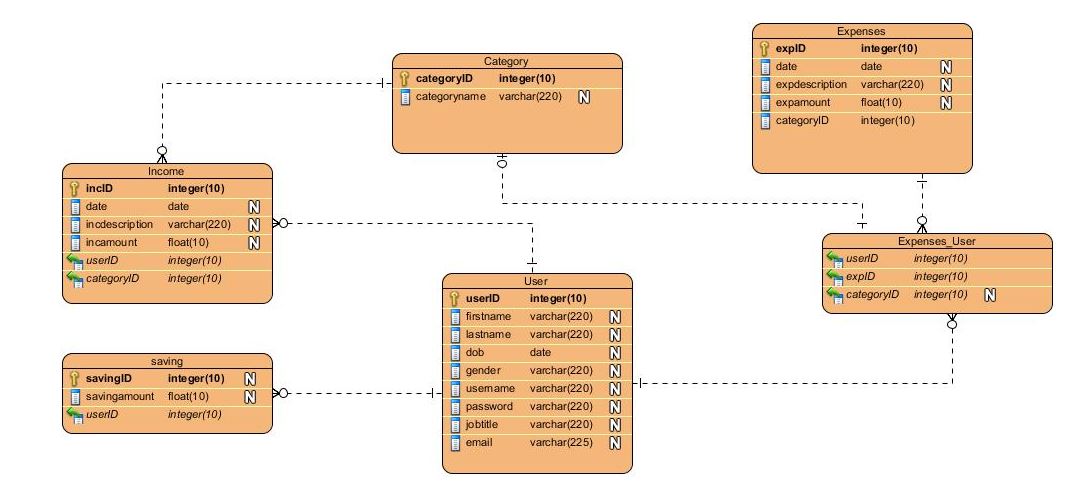
**Explanation:**

Above sequence diagram shows the interaction between user and system and the flow of message between them. If a user enter the valid credentials then user can get login access into the system. User can add their expenses, expenses category, income, income category and generate report according to their needs. If the entered credentials is invalid then user cannot get access into the system.

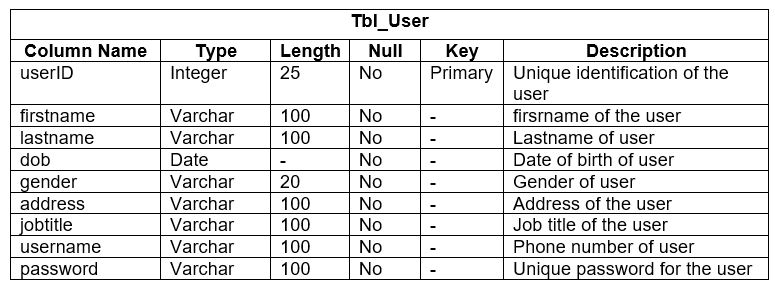
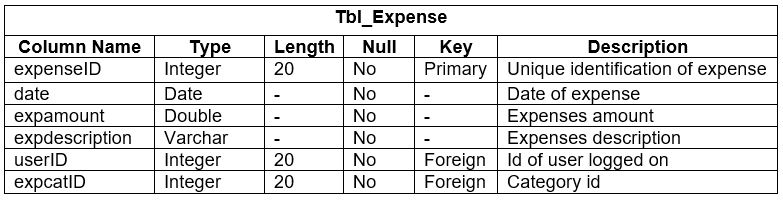
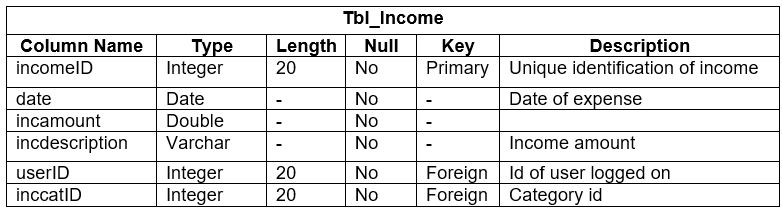
# 3.4 Database Modelling:

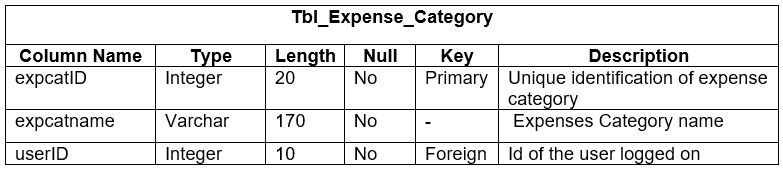
A database model decides the logical structure of database and fundamentally decides that how the data can be stored, manipulated and organized. Database model includes different models like relational models, hierarchical model, flat files model, object oriented models, entity relationship models and network models. Among them, I have chosen relational model to complete my project.

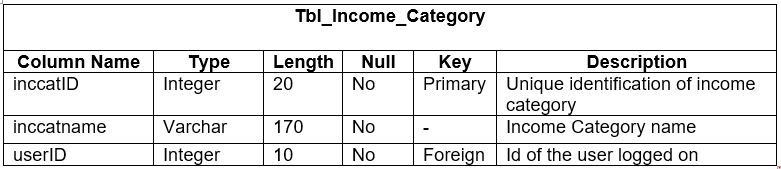
## 3.4.1 Entity Relationship Diagram:

 “Peter Chen” developed Entity Relationship Diagram and based on perception of real world in which this diagram describes the visual representation of data by defining how the data related to each other. This diagram consists of collection of object called entities and showing relationship between those entities and attributes. Below ER-Diagram based on the data dictionary.3.4.2 Data Dictionary:

A structured place to keep details of the contents of data flow process and data store. It is the structured repository of data about data. Data dictionary provides information about Data Flow, Data Structure, Data Elements, Data Stores and Data Processing.

The main purpose of data dictionary is to reduce data redundancy and for developers it helps to make normalization.

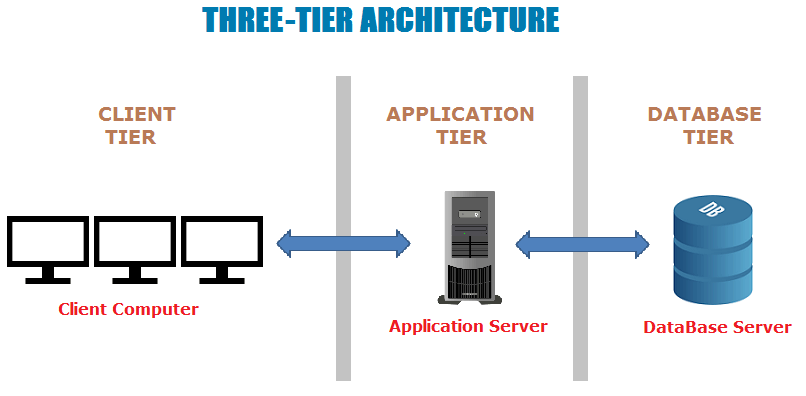




# 3.5 Architectural Model

I have chosen three-tier architecture model in which there are three components:

* Presentation tier
* Application tier
* Data tier

****

Presentation tier:

Presentation tier occupies the top-level and it displays the information related to services on website or application. It communicates with other tier like server and your database and it sends the result to the browser and other tier in network.

Application tier:

Application tier is also known as middle tier, logic tier and business logic. This is pulled from presentation tier and it controls the application functionality by performing and detailed processing. It is the part of program in Computer Software. It encodes the real world business rules and determines how data is going to be created, displayed, stored and changed.

Data tier:

Data tier houses the database server where information is stored and retrieved. The data that is kept here or independent of application server we called it as business logic includes data persistent mechanism of database servers and file servers. It also includes Data Access Layer, which encapsulate the data persistence mechanism and exposes the data and should provide API to Application tier so it exposes the method of managing i.e. store data without exposing/creating dependencies on data storage mechanism. Data Access Layer also helps in avoiding dependencies on storage mechanism and allow update and changes without application tier client and it is aware of change.

# 3.5 Prototype Design

It is an important step in design phase and represents as the draft version of an application where we can explore our ideas.

