# Chapter 3: Design

After completion of the analysis, the design phase takes place. Basically, in design, the architecture of the project is built. In this phase, the requirements will be further broken down to estimate the needed effort and amount of resources. The design focuses on identifying specific workflows and designs for the application. It includes the design of an application, databases, network architecture, user interfaces, and system interfaces. This phase sets a standard to stick to it. It allows helping remove possible flaws.

Various tools are used to create design models. For this project, various structural and behavioral models are created to show the structure and workflow of the system. Database design is created to show the backend overview, architectural model made to show network structure for the system and user interface design prototype are made to show a frontend overview.

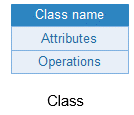
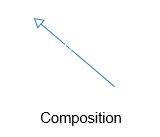
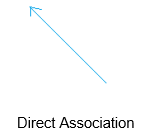
## 3.1 Structural Modeling

The structural modeling represents the static features of a system. This model represents the framework for the system. It never represents dynamic behavior of the system. The Class diagram is mostly used structural diagram.

## 3.1.1 Final Class Diagram

The class diagram is a static structure view of an application. It represents all the classes their properties and methods and the relationships between the objects. Class diagrams are widely used for object-oriented systems.

**Notation used:**

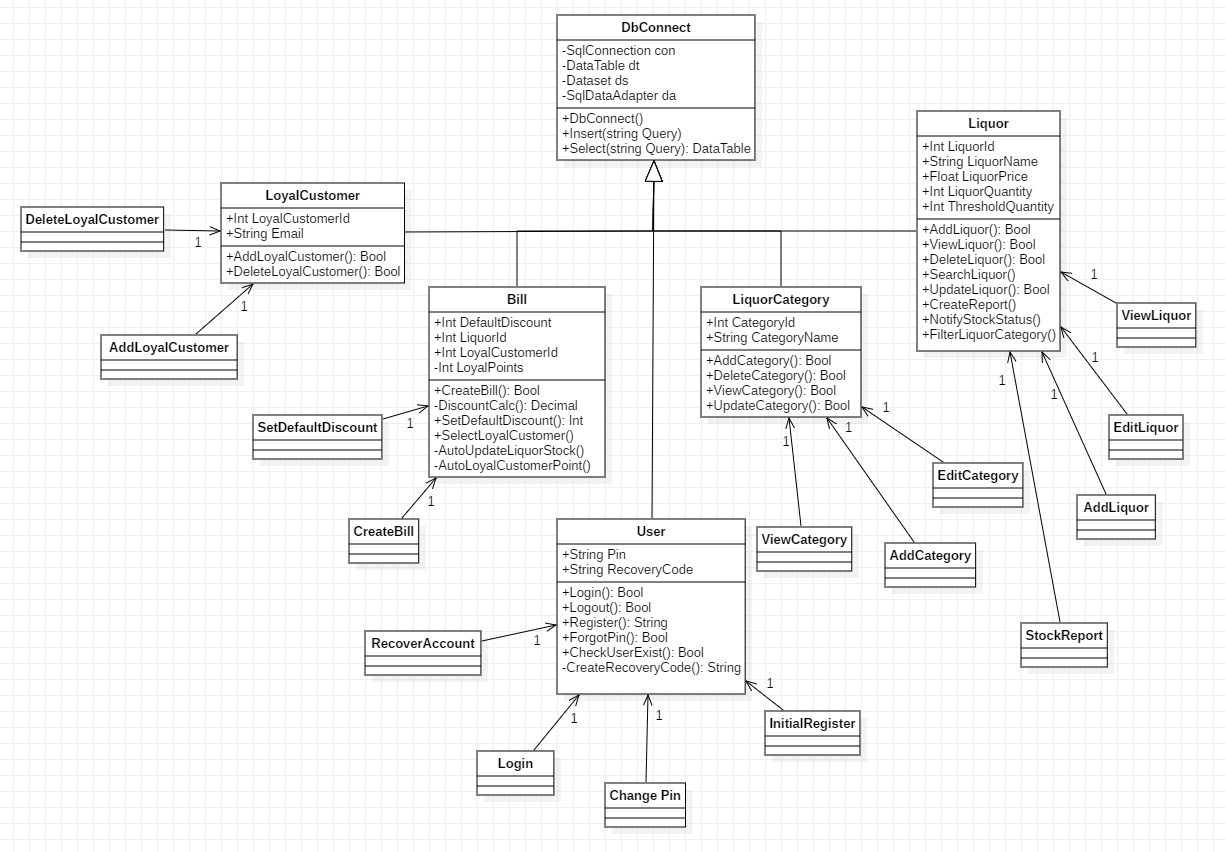


Figure 1: Final Class Diagram

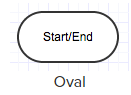
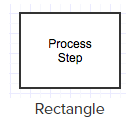
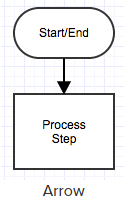
The above class diagram is for this project. The project will be developed using C# .net framework. The class diagram follows the Model View Controller design pattern. The diagram includes 5 controllers and 15 views.

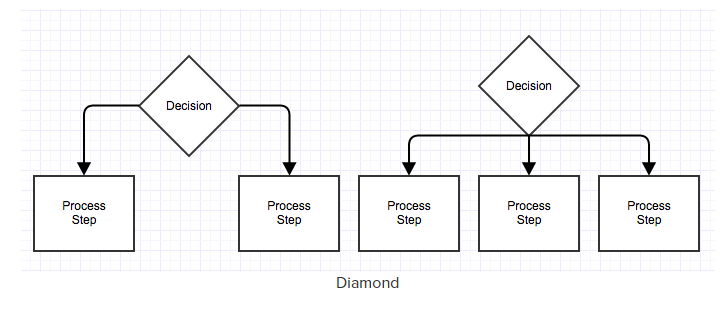
DbConnect acts as the model. It performs connection of database and all the query execution for the system. LoyalCustomer, Bill, User, LiquorCategory, Liquor classes act as the controller and other are views. The User classes perform all user verifications and registration tasks and other classes perform various functions according to the criteria. The windows forms act as the view for the system.

## 3.1.2 Flowchart

A flowchart is a graphical representation of the computer algorithm. It is a step by step approach to solve a task or show the process or workflow.

**Notation used:**



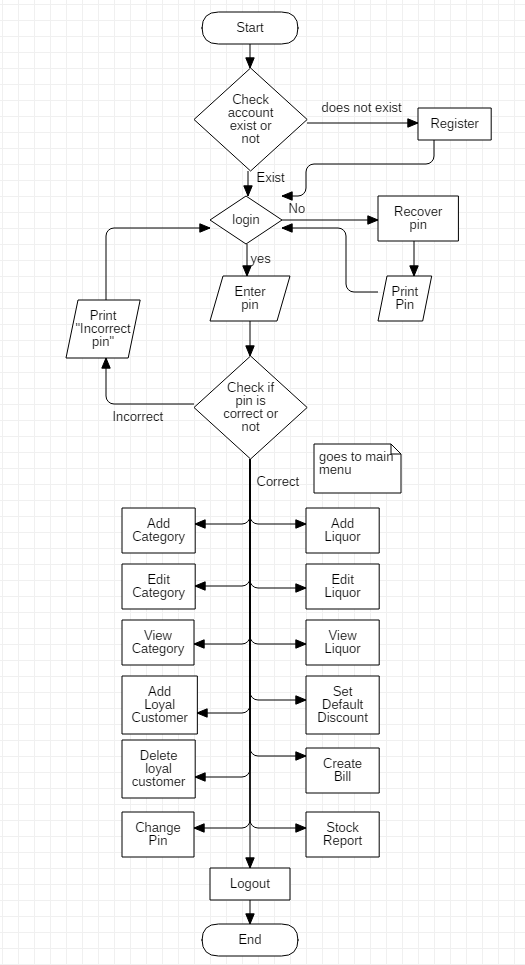


Figure 2: Flowchart

The flowchart shows the application’s initial registration and login process. If an account exists then it will straight proceed to open Login form. User can either login using an authenticated registered pin or recover pin using recovery code that is provided after registration. After logging in user can perform all the tasks as listed in the rectangle box. The flowchart ends when user logs out from the system.

## 3.2 Behavior Modeling

Behavior model represents the interaction among the structural diagrams in the system. It shows the dynamic sequence flow of the system.

## 3.2.1 Activity Diagram

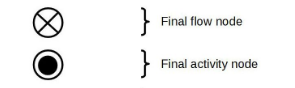
An activity diagram represents the flow or sequences of activities in a process. It includes sequential, parallel activities and decision that are made in the process. It represents the dynamic workflow behavior of a system.

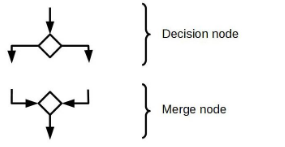
**Notation Used:**

****

****

****

****

****

****

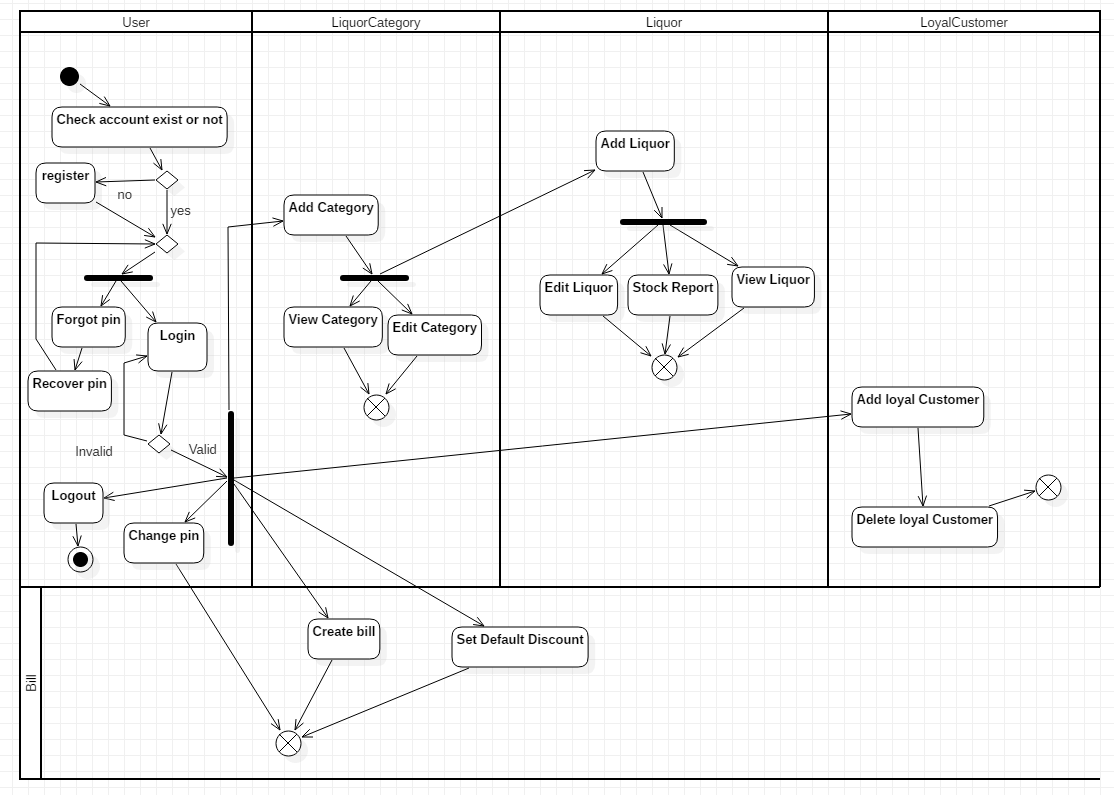


Figure 3: System Overview Activity Diagram

Above activity diagram is the full system overview of how system registration, login and its function occurs. The system checks if there is an account or not. If an account exists it will not allow to register again. The user is able to login after registration of the system. The user gets recovery code after registration which can be used to recover pin when the user forgets the pin code. After logging in, the user can add liquor to stock if the liquor category exists. Liquor category is essential for adding liquor. User can create a bill, set default discount percent per 100 points, add and delete loyal customers, change the pin and log out to end the flow.

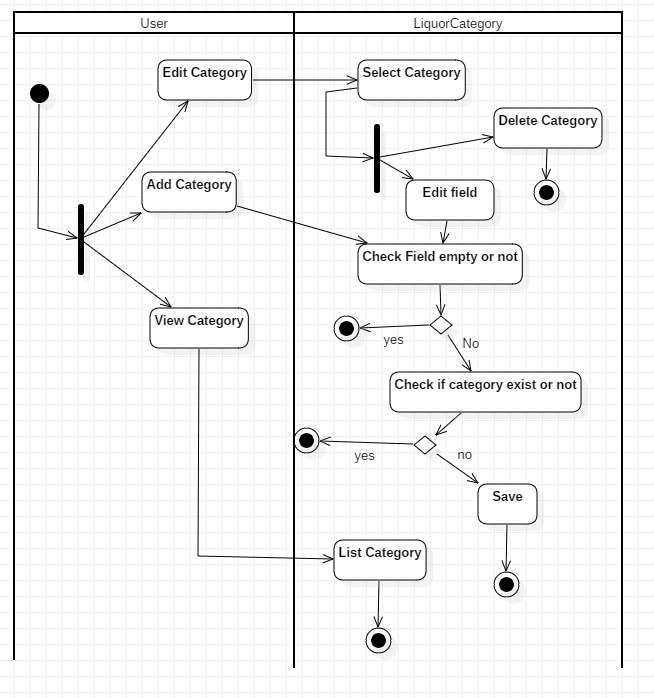


Figure 4: Liquor Category Activity Diagram

This activity diagram shows how the user add, edit, delete and view the category. While adding new category or editing existing category it checks if null values are entered or not. It also does not allow to enter the same category twice.

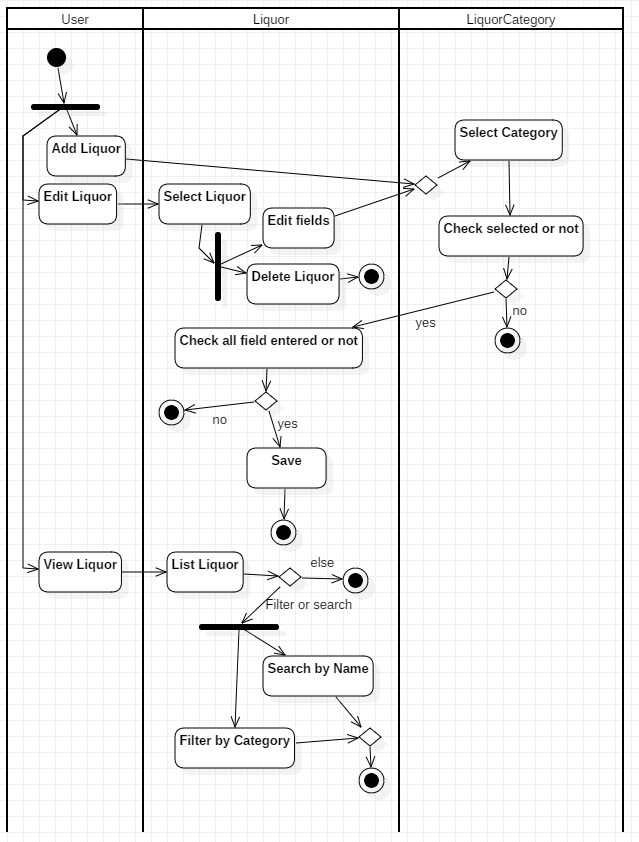


Figure 5: Liquor Activity Diagram

This activity diagram shows how the user add, edit, delete and view the liquors. Similar to the liquor category it does not allow any null value while adding or editing the liquors. User can search the liquor by its name and filter the liquor according to its category while viewing the liquors.

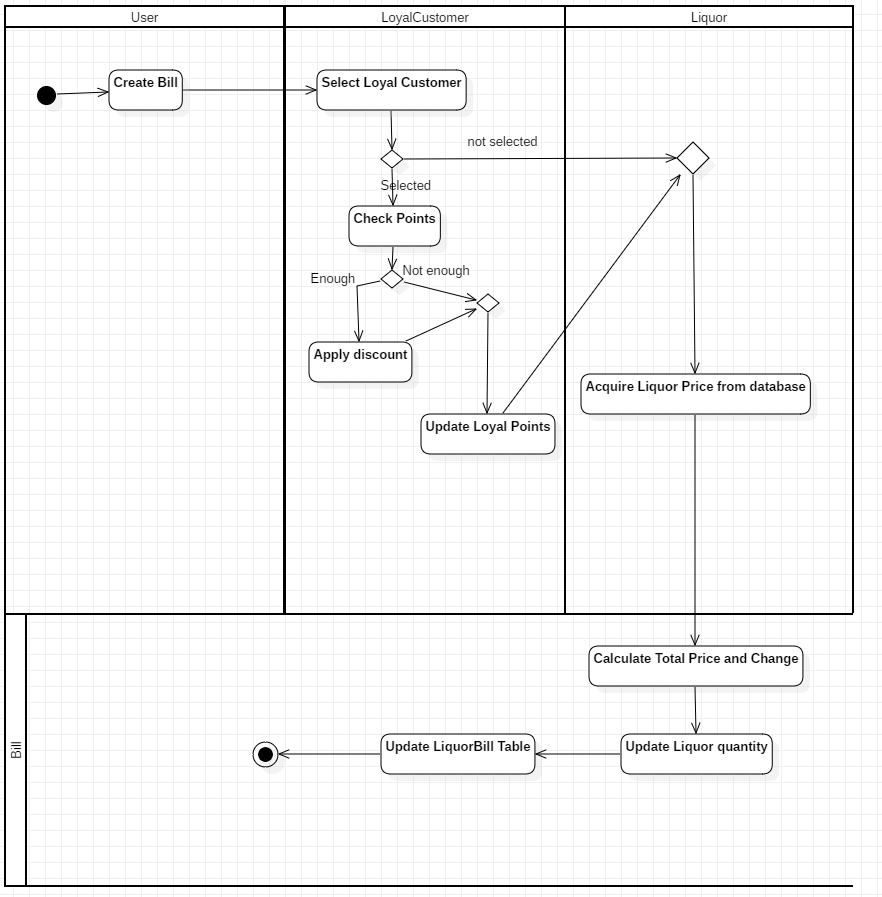


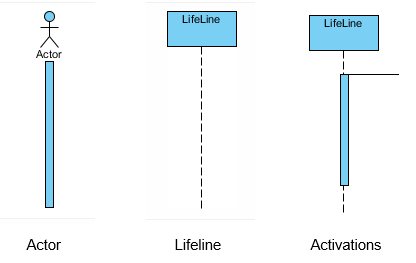
Figure 6: Bill Activity Diagram

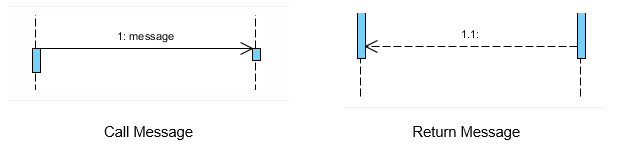
This activity diagram shows how the discount is applied for a loyal customer when they have enough points while creating the bill. After calculating the total price the bill updates the liquor stock quantity and stores bill information separately.

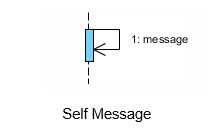
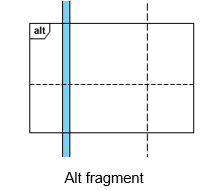
## 3.2.2 Sequence Diagram

A sequence diagram is used to show the interaction between objects in a sequential order in which it takes place. It describes how and in what order the objects in a system functions. It allows to show the logic of relationships between the objects.

**Notation Used:**

****

****

**** ****

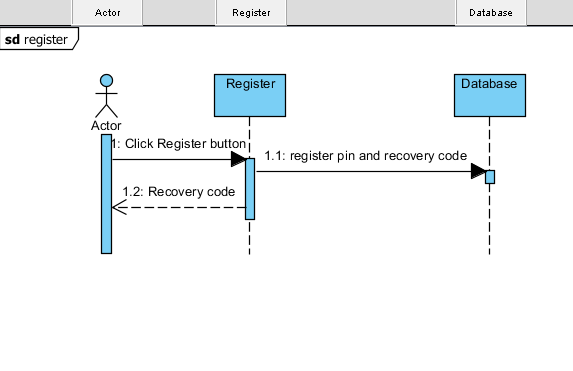


Figure 7: Initial Register Sequence Diagram

This Sequence diagram shows the initial registration process of the application. After entering the pin the register sends to store in database as well as provides recovery code to the actor.

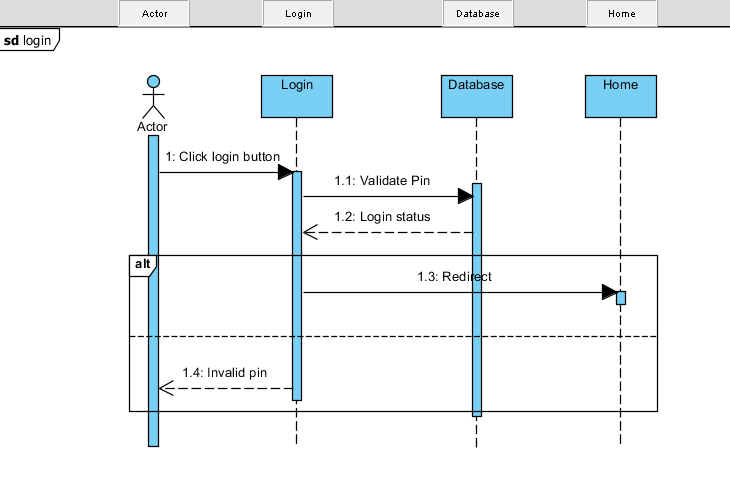


Figure 8: Login Sequence Diagram

This sequence diagram shows the login process takes place in sequential order. Validation takes place in the database and if it is authentic then the actor is redirected to home else shows actor the pin is invalid.

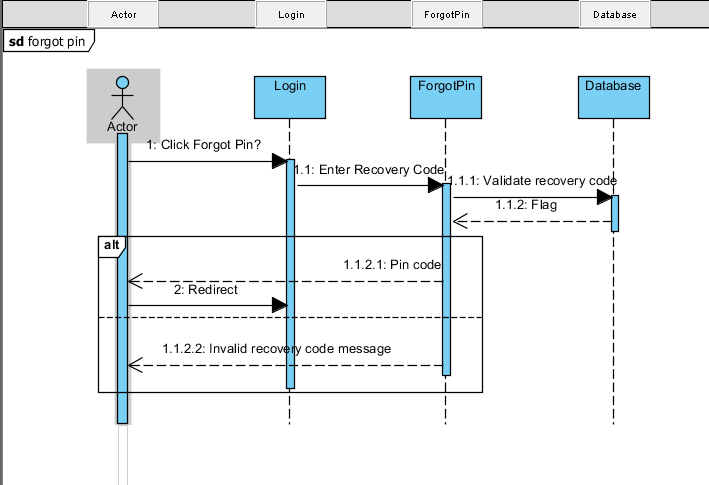


Figure 9: Forgot Pin Sequence Diagram

This sequence diagram shows the pin recovery process. After selecting forgot pin and entering recovery code through login, ForgotPin validates the recovery code in the database. If the code is valid then it returns pin code else it throws invalid recovery code message.

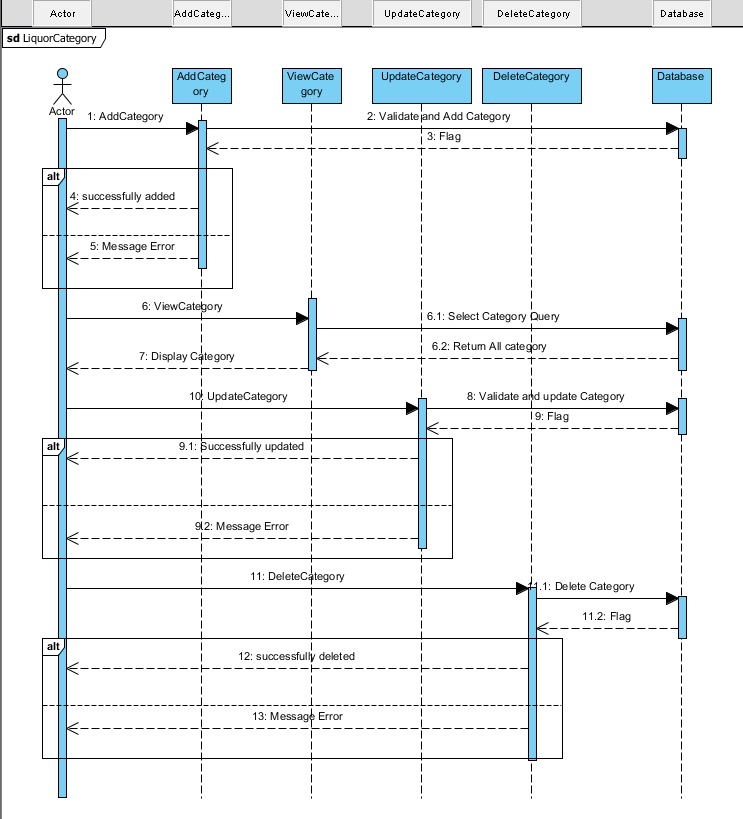


Figure 10: Liquor Category Sequence Diagram

This sequence diagram shows add, update, delete, and view process of liquor category. While adding, updating the category, the database validates and creates a flag telling whether the process is successful or unsuccessful. While viewing category the query is executed in DataBase and returns a list of the liquor category to display. While deleting if there are any exception it throws an error message.

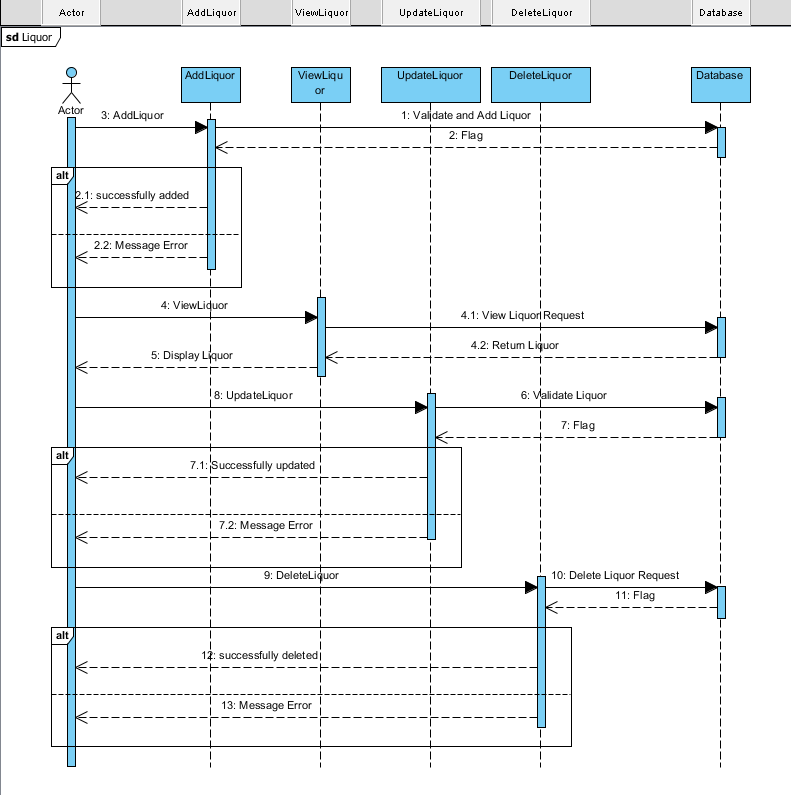


Figure 11: Liquor Sequence Diagram

This sequence diagram shows add, update, delete, and view process of the liquor. While adding, updating the liquor, the database validates and creates a flag telling whether the process is successful or unsuccessful. While viewing liquor the query is executed in DataBase and returns a list of the liquor to display. While deleting if there are any exception it throws an error message.

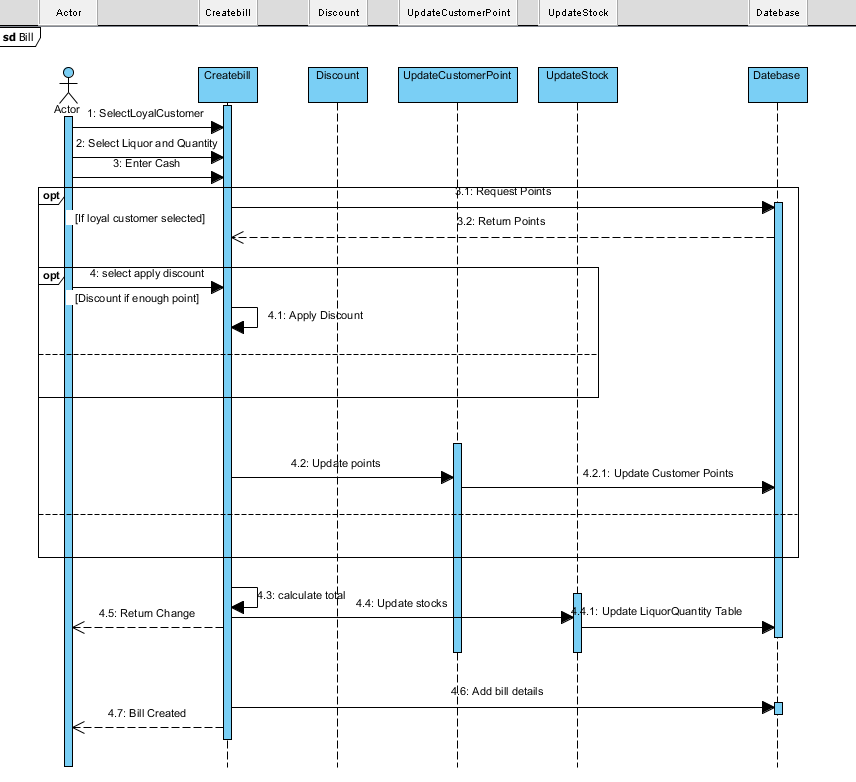


Figure 12: Create Bill Sequence Diagram

This sequence diagram shows the process of creating a bill. It allows to enter liquor, its quantity and paid cash. It shows if the loyal customer is selected then their points are updated. If the loyal customer has enough points they can apply the discount. The total amount is calculated and the liquor quantity stocks are updated. It returns change cash and adds bill details in the database.

## 3.3 Database Modeling

Database modeling is the process of designing a database model that helps while developing and implementing the system. It shows the relationship between entities, normalized table, metadata, and fields.

## 3.3.1 Data Dictionary

The Data dictionary is a set of files that contains metadata of a database. In the following data dictionary data type, length, null or not, key and constraints are included.

Table: User

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** | **Constraints** |
| Pin | varchar | 25 | No | Primary Key | Pk\_Pin |
| RecoveryCode | varchar | 25 | No | - | - |

Table: LiquorCategory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** | **Constraints** |
| CategoryId | int | - | No | Primary Key | Pk\_CategoryId |
| CategoryName | varchar | 255 | No | - | - |

Table: Liquor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** | **Constraints** |
| LiquorId | int | - | No | Primary Key | Pk\_LiquorId |
| LiquorName | varchar | 255 | No | - | - |
| LiquorPrice | decimal | - | No | - | - |
| CategoryId | int | - | No | Foreign Key | Fk\_CategoryId |

Table: LiquorQuantity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** | **Constraints** |
| LiquorId | int | - | No | Foreign Key | Fk\_LiquorId |
| Quantity | int | - | No | - | - |
| Threshold | int | - | No | - | - |

Table: Bill

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** | **Constraints** |
| BillId | int | - | No | Primary Key | Pk\_BillId |
| TotalAmount | decimal | - | No | - | - |
| CustomerId | int | - | Yes | - | - |

Table: LoyalCustomer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** | **Constraints** |
| LoyalCustomerId | int | - | No | Primary Key | Pk\_LoyalCustomerId |
| Email | varchar | 255 | No | - | - |
| Points | int | - | No | - | - |

Table: LiquorBill

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Name** | **Type** | **Length** | **Null** | **Key** | **Constraints** |
| BillId | int | - | No | Foreign Key | Fk\_BillId |
| LiquorId | int | - | No | Foreign Key | Fk\_LiquorId2 |
| LiquorAmount | decimal | - | No | - |  |

## 3.3.2 ER diagram

An ER diagram is the structural diagram for use in database design. It shows the major entities, attributes and their relationship among them.

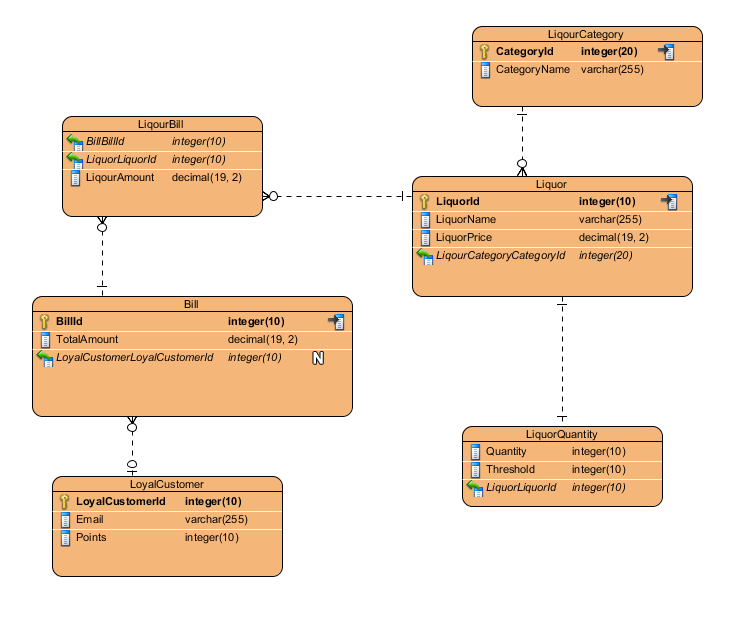


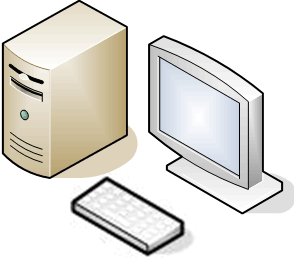
Figure 13: Entity-Relationship Diagram

The Above ER diagram is built based upon the data dictionary in 3.3.1. In this diagram, the entities are shown with their attributes and their respected relationships with each other.

There are total 8 tables for this project. The User entity is not included in the above ER diagram because it is an independent entity as the system is standalone application thus does not requires any relationship with other tables.

## 3.4: Architectural Model

The system is being built for the standalone computer. Basically, it is a computer system that can run a local application on its own without the need of LAN or WAN connection.



The system that is being built is for a small/ medium size liquor store where there traditional hardcopy way of storing stock information and billing system. The main objective is to make an automated stock and billing system for a single store. The standalone architecture main benefit is it does not require any external connection for the application to run.

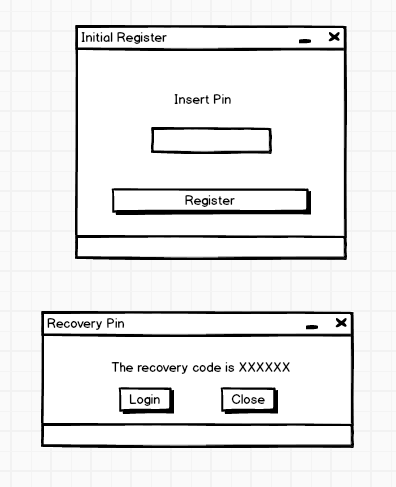
The database is store in the singular computer where the application is installed. This allows access of database virtually faster than any other architecture i.e. client-server, peer to peer. The standalone application is never kept online, and remote access of data is practically impossible. This allows for high security of the stored data.

The computer resources are fully available for the application whereas one user might waste resources in the client-server architecture. It allows monitoring and accessing application to be more controlled. Whenever the application crashes it can be reinstalled without any problem of accessing the network.

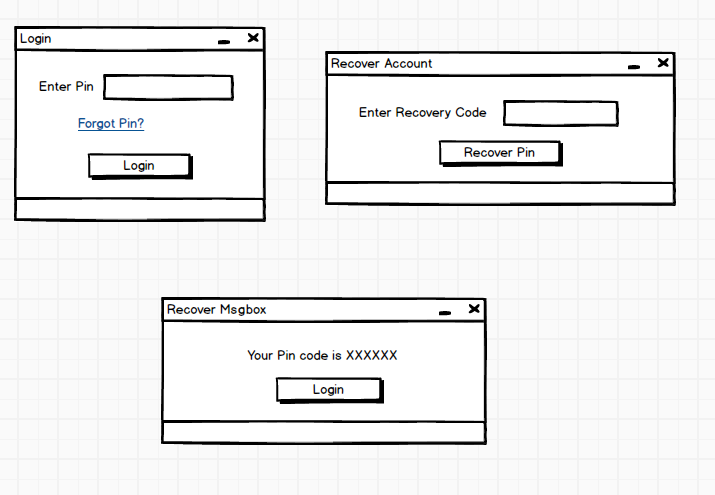
The standalone computer is the best to use for having the support of more hardware such as barcode reader, printer, biometric devices, etc. The stationary printer is still relevant and can be better according to preference. The feature for printing bills, using the barcode to scan liquor can be implemented.

## 3.5: Prototype UI Design

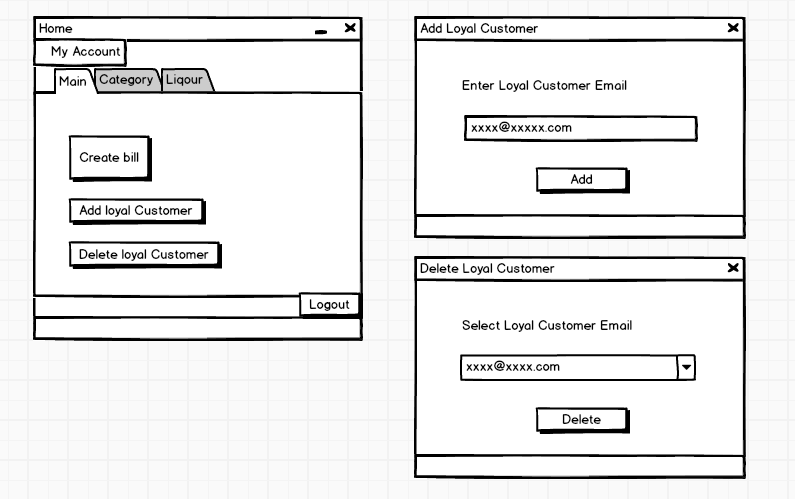
The graphical representation of an application is called the user interface. The prototype UI design is the early model of the application’s user interface. Here is the prototype The UI design of this project.



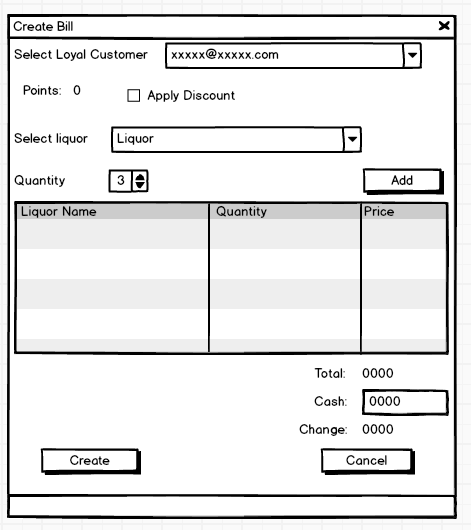
Screenshot 1: Initial Register form



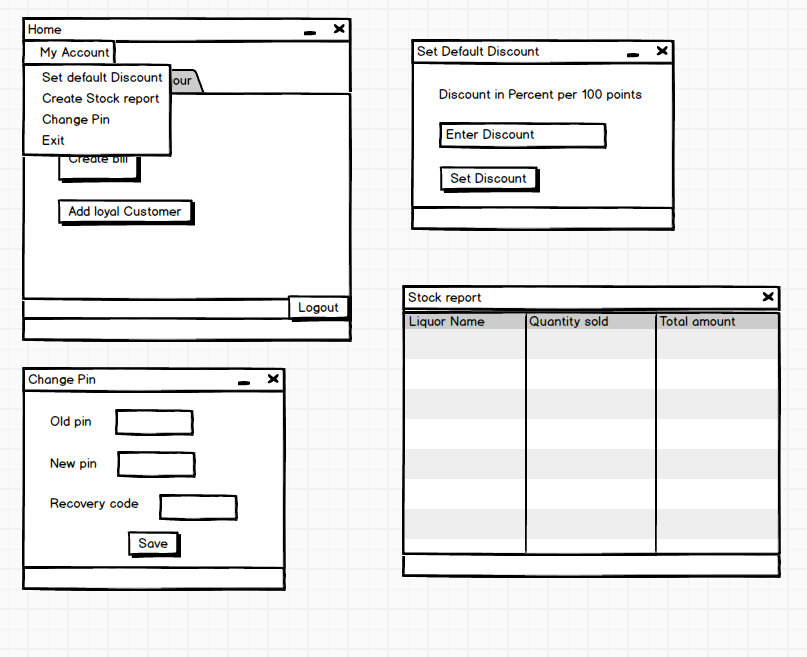
Screenshot 2: Login and recover account form



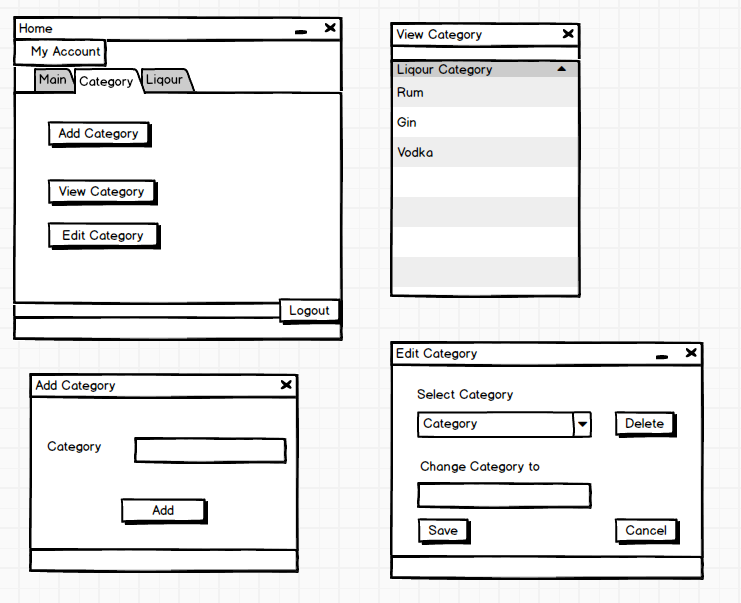
Screenshot 3: Home, Add loyal customer, Delete Customer form



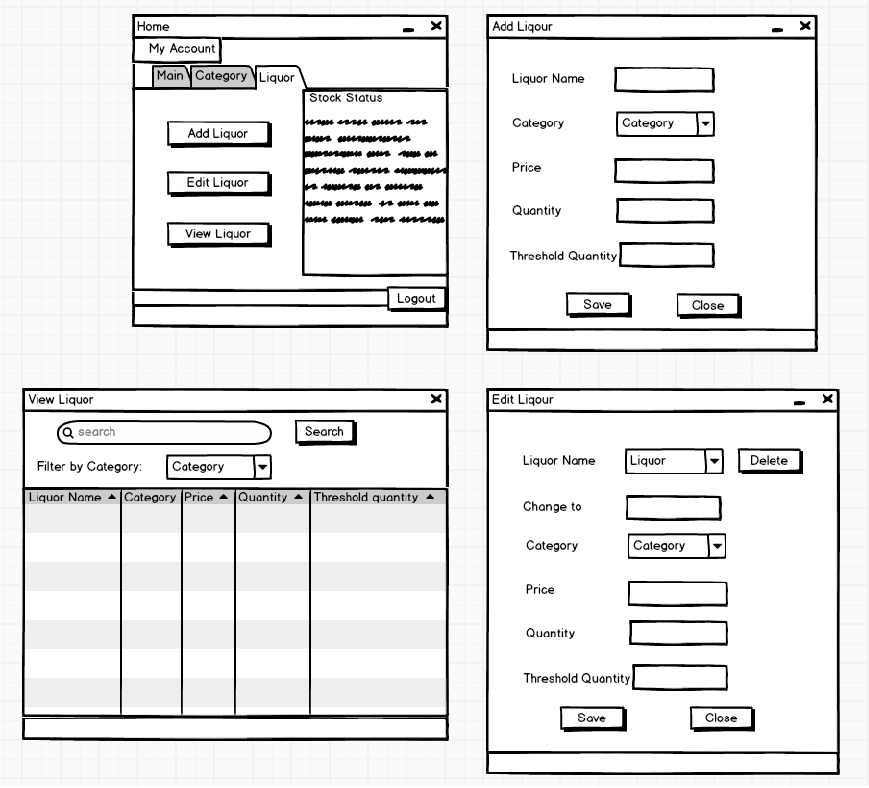
Screenshot 4: Create Bill form



Screenshot 5: Menu bar and its forms



Screenshot 6: Liquor Category forms



Screenshot 7: Liquor forms