# **Chapter 3: Design**

# Introduction to design:

The third phase of software development is design which answer the question “how” in software development. It displays the ideas how the system will appears. The user requirement will be converted in suitable form with the help of design. Which further helps developer in implementation of software. During development of software various tools and technique are used which makes the design more effective and efficient. It is the method of defining various methods, technique and overall structure of the code which will further satisfy the user’s requirements.

It is an important part of software development because it reduces redundancy and increases reusability. It also allows software developer and owner to allow how the function of software really works. Here, I have used structural and behavioral model to show the work flow of system which is going to be developed. Furthermore, database design and user interface design is also provided to show the overview of the system as how it works or the system backend.

# 3.1 Structural Model

## 3.1.1 Final Class Diagram:

A static view which represent the static view of the system or application is class diagram. It helps to visualize the system and helps to create the executable code of the system. It describes operation of class, its attributes and constraints executed in the system. It is a structural diagram which shows the classes, interface, constraints and associations.

Some of the reasons to use class diagram are mentioned below:

* Systems responsibility is described by it.
* Static view of the application is described by diagram.
* Supports forward and reverse engineering both.

**Notations used:**

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Notation** | **Description** |
|  | Class | It is a templet which is used to create object and provides the initial behavior of the system. |
|  | Attribute | Structure of a class is described and represent the data definition for an instance of class. |
|  | Operation | A function that can be performed by an instance of class or interface is an operation. |
|  | Interface | It is a structure that allows system to enforce some properties in a class. |
|  | Aggregation | Aggregation is a process of compiling data and records from database to put together mixed datasets for facts processing. |
|  | Association | It is used to show the relationship between classifier and instances of that classifiers. |
|  | Generalization | Generalization shows the relationship between one model and another. Which means whether the class receives all of attributes and operations from parent class or not. |
|  | Composition | It is a strong form of binary association in whole part of relationship. |
|  | Dependency | In this relationship one element, the client is dependent to another element, the supplier. |

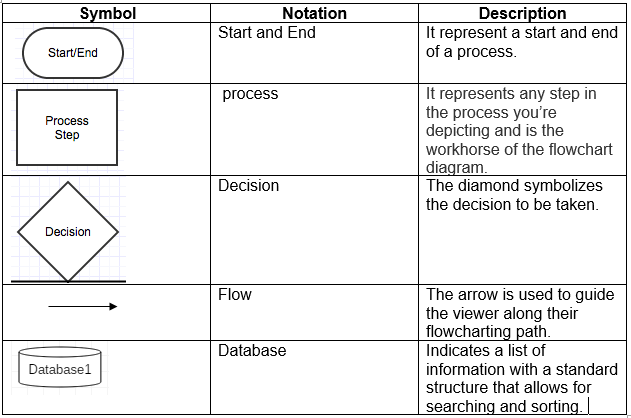
## 3.1.2 Flowchart:

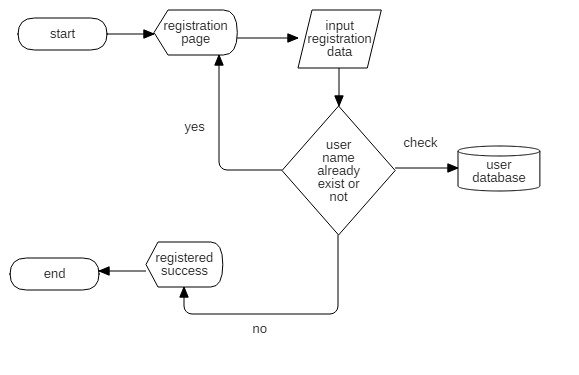
A flow chart is a diagram that represent the sequence of steps and decisions needed to perform a workflow. It is also a diagrammatic representation of algorithm. It gives a solution model to a given problem. Also it is used in analyzing, managing or documenting a program in different field.

Some benefits of flowchart are mentioned below:

* Visual clarity
* Instant communication
* Effective coordination
* Efficiency increases
* Effective analysis
* Problem solving

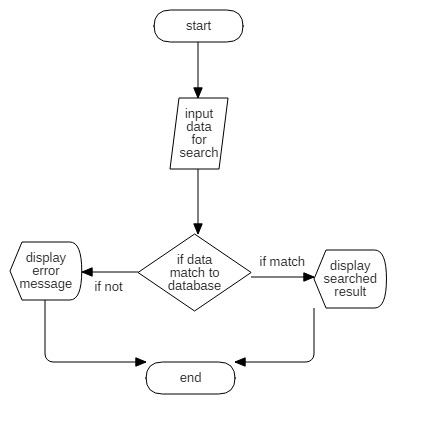
**Notation:**

****



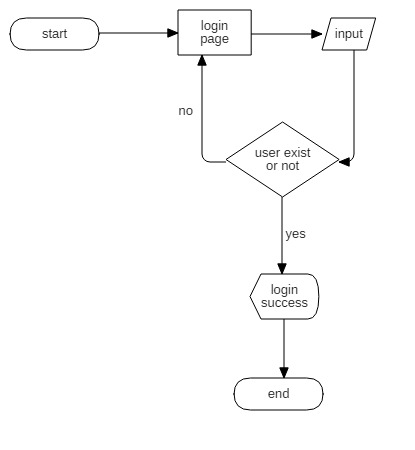
**Figure 1: Register flowchart.**

Here, the above diagram show the registration process that a user do in a system. User will move on in registration page and fill up the form. If the inserted data will exist then it will move back to registration page and if not exit then the process will be succeed.



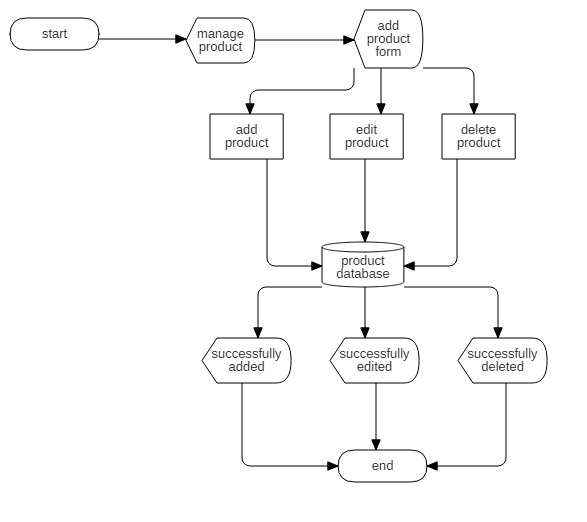
**Figure 2: Search flowchart.**

Here, user search the product he/she want. If input detail will match to database than it will display the result and if not the process will end and system will display error message.



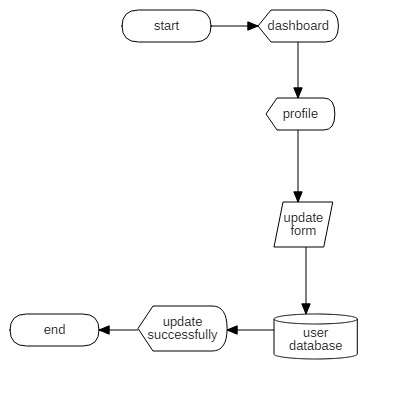
**Figure 3: login flowchart.**

Here, user enter their username and password. If the username and password match or existed in system the login attempt will be successful otherwise the process will stop or move to login page.



**Figure 4: Admin flowchart.**

The above diagram illustrates that admin can manages the product and can add product form. Here, the admin can edit, add and delete the products. Every steps that the admin follows will be checked in database. Then all the products are managed.



**Figure 5: update flowchart.**

In the above diagram admin can update the profile using update form. After login to the dashboard the admin can change some information which is also checked by the database and will be updated to the system.

# 3.2 Behavioral Model:

Behavioral model represents the dynamic behavior of the system. It show what is supposed to happen to a system when it respond to stimulus from it environment. It also motivate the aim of obtaining a general framework for system. It does not distinguish between input and output variables which is a most important aspect of it.

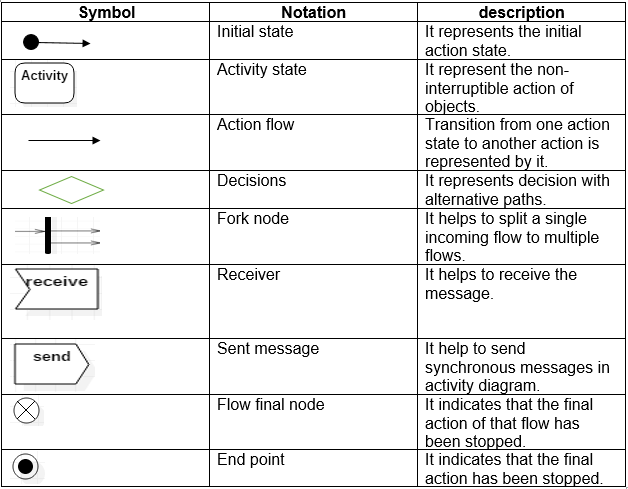
## 3.2.1 Activity Diagram:

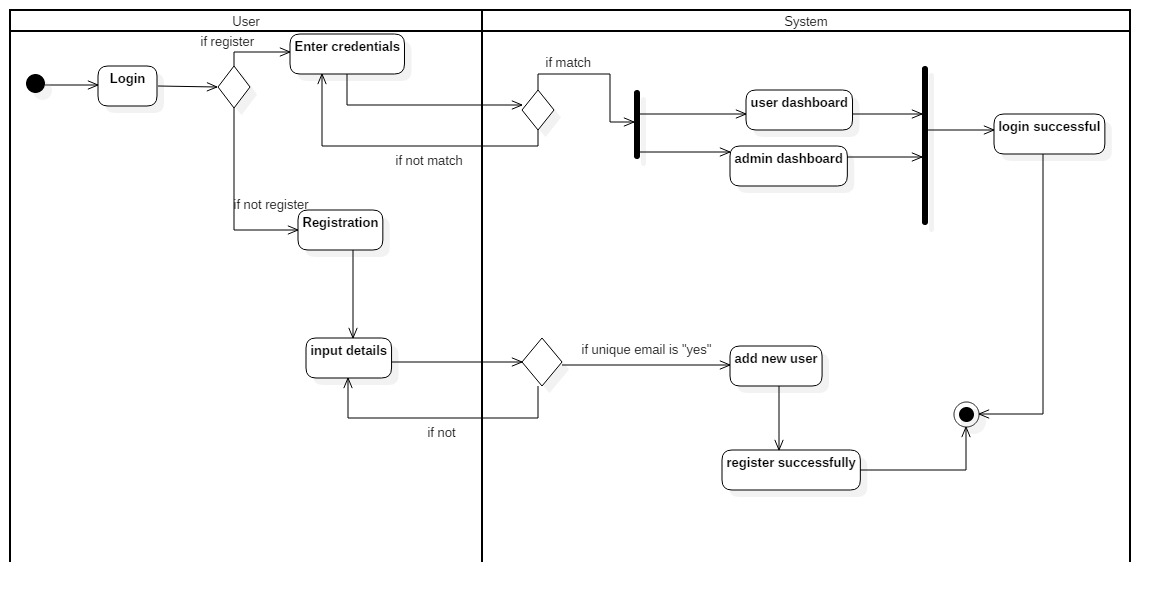
The dynamic aspect of system is described by activity diagram. Dynamic behavior is also captured by it. It shows the message flow of one activity to another activity. It uses reverse and forward engineering show that to create the executable system.

Propose of activity diagram is described below:

* It show the flow of activity that takes place in a system.
* It describe the sequence of one activity to another activity.
* Parallel, branched and concurrent flow of the system is described.

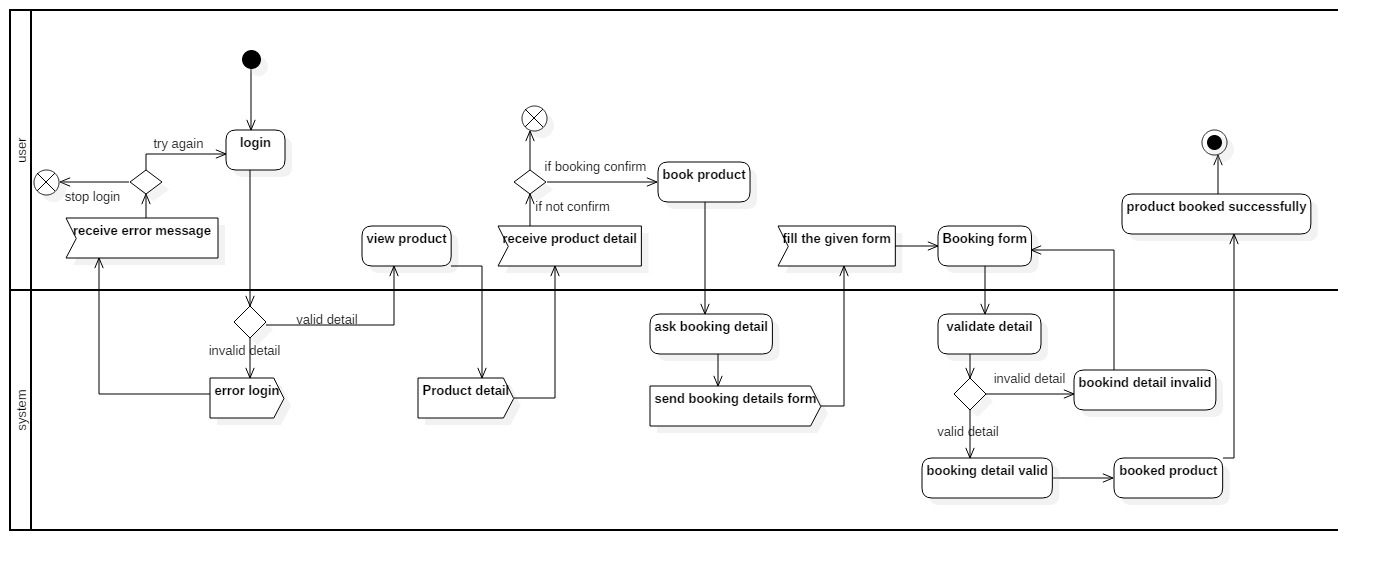
**Notation used:**

****

****

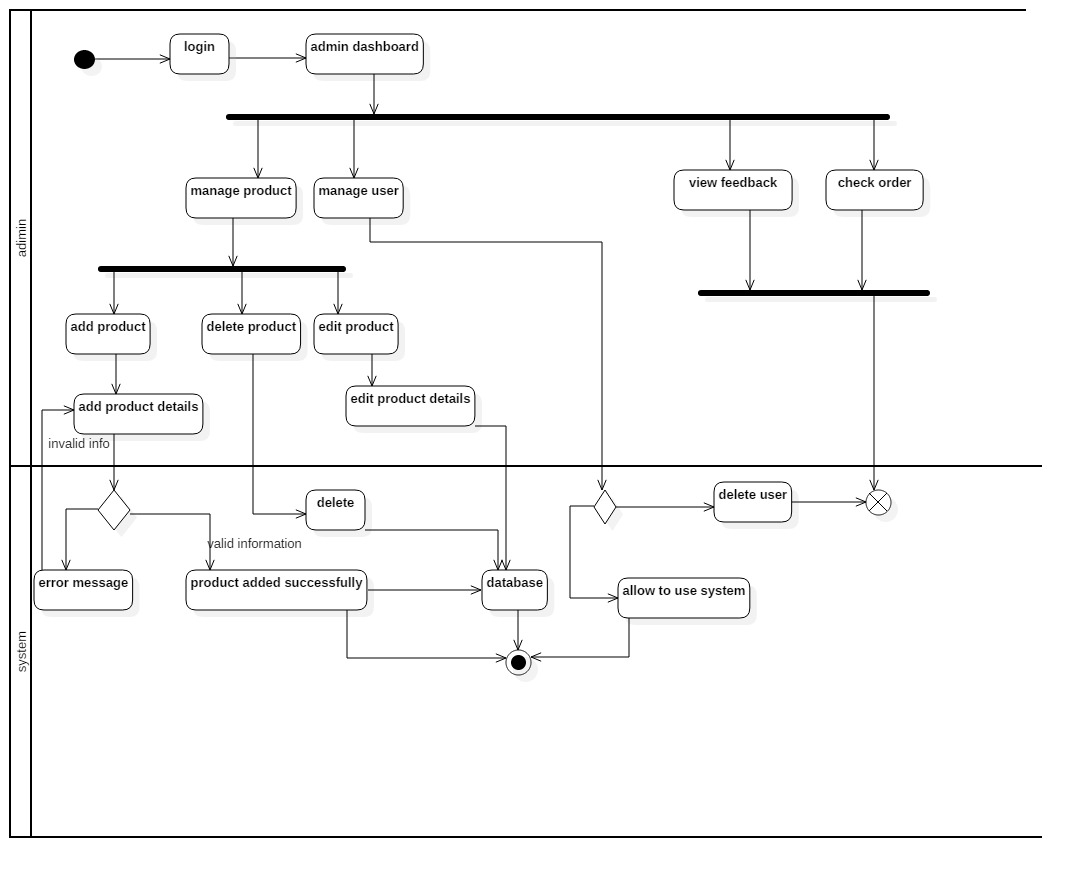
**Figure 6: user register and login activity diagram.**

Above mentioned diagram is user registration and login diagram. Here the new user can register by entering credentials. If the input information is matched according to the system validation the new user will be added to the system. And, the already registered user will directly login to the system using their username and password.

****

**Figure 7: user activity diagram.**

Above diagram mention the user activity in a system. The user must login first and then will have access to function of system. After that, user can view all the products and their details. Users can also book the product if they want to buy it later; by filling the booking detail form. Moreover, user will get a “booked successfully” message after booking the products.

****

**Figure 8: admin activity diagram.**

Above diagram mention the admin activity diagram. Here, admin will manage all the products, users, feedback and orders. For managing products, admin will add, edit and delete the product which will be saved in database. For managing users, admin will view all the registered users and allow them to use system functionality if their provided detail is valid. Moreover, admin will view feedback and orders given by users.

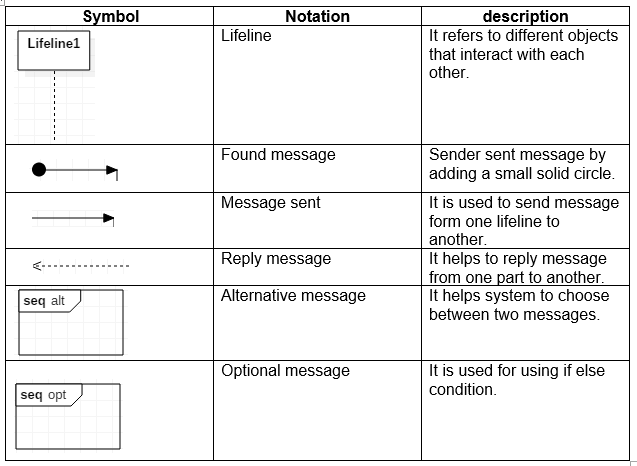
## 3.2.2 Sequence Diagram:

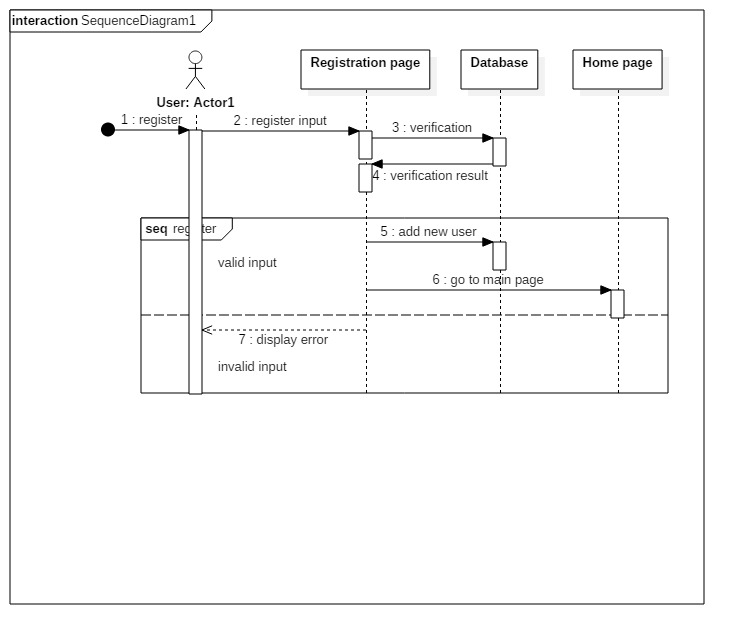
A sequence diagram shows that how the operations are carried out in a system. This type of diagram is time focused and shows the order of action in horizontal or vertical way. It is the way of visualizing aim of system in a standard way. It shows the iterative behavior of a system.

Some propose of sequence diagram is mentioned below:

* High level of interaction between the active object is maintained by it.
* Shows each and every detail of UML in use case.
* Show the interaction between the object and component of use case.
* Each and every details is represented.

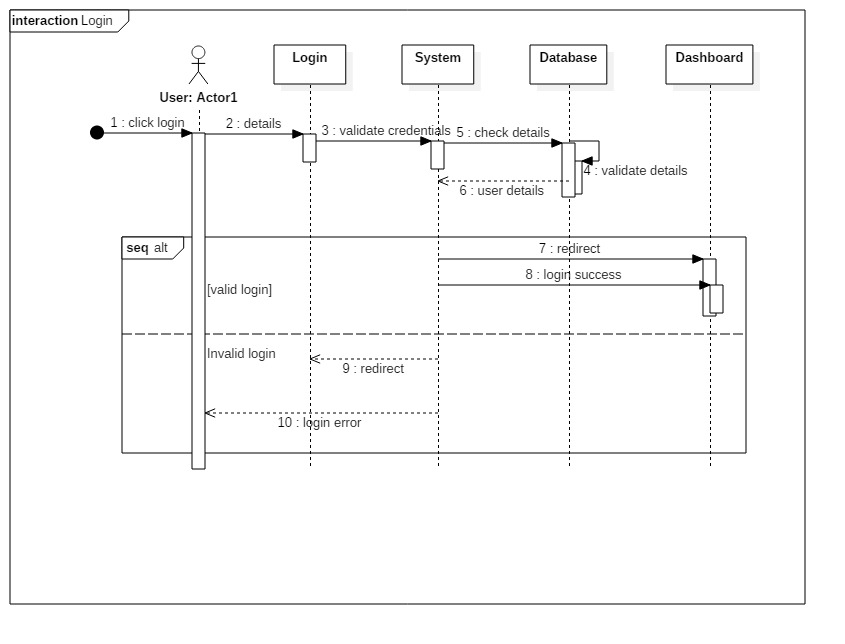
**Notation:**





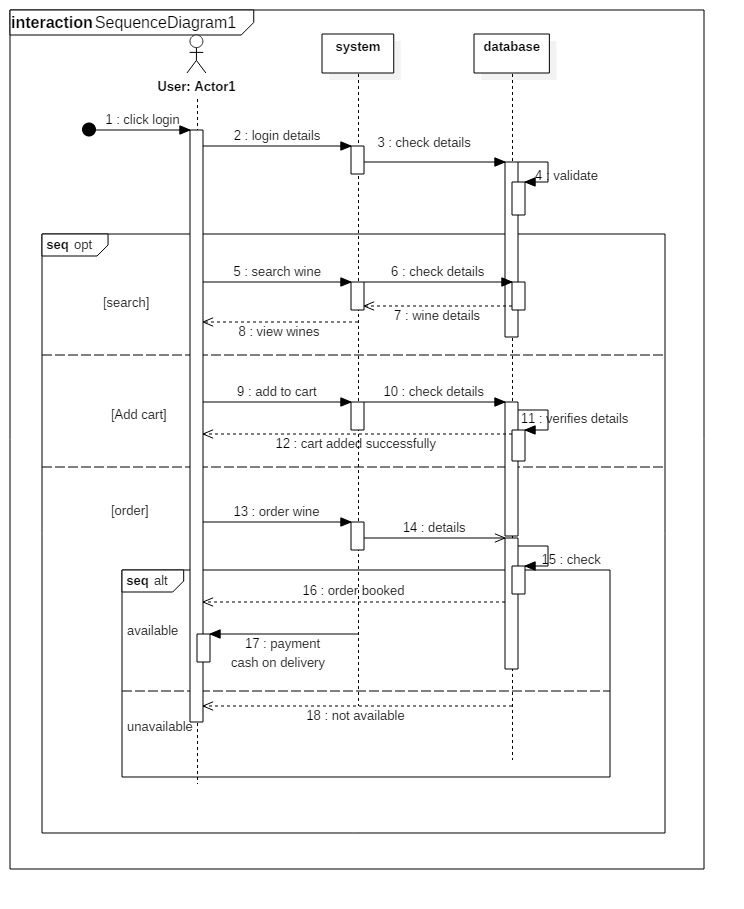
**Figure 9: sequence diagram of registration.**

Above diagram mention the sequence diagram of registration of user. Here the user input the information which is verified by database. Then the new user is added and if the information will be valid then it move forward to home page otherwise it displays the error. Which means it return back to registration page.



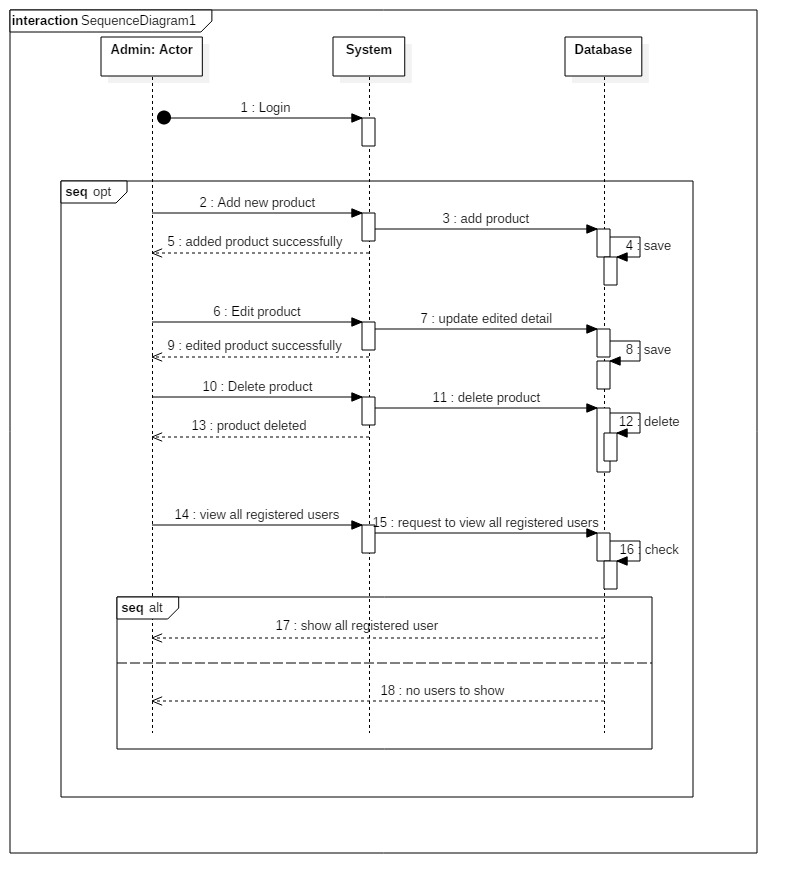
**Figure 10: login sequence diagram.**

Above sequence diagram show the user login system. Here, the user inputs the valid detail which is checked and validated by database. If the data match then login success message is popup. But if the input data mismatch to the data inside the database then login attempt will stop and displays the error message. And then it automatically return to login page.



**Figure 11: user sequence diagram.**

Here, after login to the system by user they can move on to the system and can search verities of wines. Also they can add the product to cart and can order the product that have been successfully added to the cart. If the information will be valid then the product will be booked but if the data mismatch or incorrect then error message will displays.



**Figure 12: Admin sequence diagram.**

Above diagram mention the admin dashboard panel and the activities that is done by admin. Here the admin add, edit and delete the product. Before performing those activities all are checked in database and replied message by database to system. Also user can check the registered user and booking details. Order placed by user are also checked by admin.

# 3.3 Database Model:

A model that represents the consistence flow of logical structure in database and which shows the way in which data is stored, manipulated and organized is database model.

## 3.3.1 Data Dictionary:

Data dictionary is a collection of name, attributes, datatype and constraints that are being used in database. It explains the purposes of each and every elements within the context of system. It also provide metadata of data elements.

Purpose of using data dictionary are described below:

* It helps in avoiding data inconsistencies throughout a project.
* Makes easy to analyze data.
* The use of data standard is implemented.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Userid | integer(10) | - | yes |  |
| 2 | Firstname | varchar(20) | - | yes |  |
| 3 | Middlename | varchar(20) | - | yes |  |
| 4 | Lastname | varchar(20) | - | yes |  |
| 5 | Mobileno | varchar(20) | - | yes |  |
| 6 | Dob | varchar(20) | - | yes |  |
| 7 | Address | varchar(50) | - | yes |  |
| 8 | Email | varchar(50) | PK | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Ordered | integer(10) | PK | No |  |
| 2 | Customername | varchar(30) | - | yes |  |
| 3 | Address | varchar(30) | - | yes |  |
| 4 | Phone | varchar(20) | - | yes |  |
| 5 | Date | date | - | yes |  |
| 6 | Useremail | varchar(50) | FK |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Productid | integer(10) | PK | No |  |
| 2 | Productname | varchar(30) | - | yes |  |
| 3 | Price | float(10) | - | yes |  |
| 4 | Details | varchar(50) | - | yes |  |
| 5 | Image | varchar(20) | - | yes |  |
| 6 | Producttypeproductid | integer(10) | FK |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Bookingid | integer(10) | PK | No |  |
| 2 | Productname | varchar(30) | - | yes |  |
| 3 | Price | float(10) | - | yes |  |
| 4 | Quantity | varchar(40) | - | yes |  |
| 5 | Useremail | varchar(50) | FK |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Subscribeid | integer(10) | PK | No |  |
| 2 | Email | varchar(50) | - | yes |  |
| 3 | Useremail | varchar(50) | FK |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Productid | integer(10) | PK | No |  |
| 2 | Type | varchar(20) | - | yes |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Id | integer(10) | - | yes |  |
| 2 | Cost | float(10) | - | yes |  |
| 3 | Quantity | Varchar(25) | - | yes |  |
| 4 | Productproductid | Integer(10) | FK |  |  |
| 5 | orderdetailorderid | Integer(10) | FK |  |  |

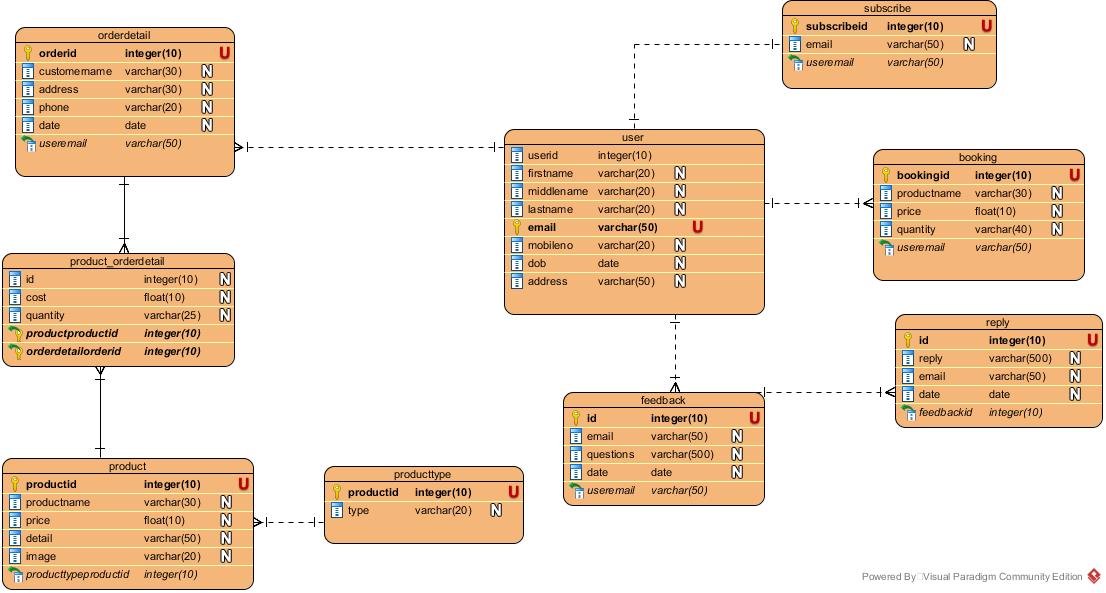
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Name** | **Data type** | **Constraints** | **Null able** | **Documentation** |
| 1 | Id | integer(10) | PK | No |  |
| 2 | Email | varchar(50) | - | yes |  |
| 3 | Questions | Varchar(500) | - | Yes |  |
| 4 | Date | Date | - | yes |  |
| 5 | Useremail | Varchar(50) | FK |  |  |

## 3.3.2 ER Diagram:

An entity relationship diagram is a technique that graphically illustrates the entities and their relationship. It provides the visual starting point to a database design which facilitates in managing the information system of a system. It is a conceptual and representational model that represent entity framework infrastructure.

Some advantages of ER diagrams are discussed below:

* Proper documentation of database design.
* Almost anyone can understand it as it is visually presented.
* Changes to database design could be maintained.
* Database designer and programmer could represent it without any doubt.



**Figure 13: Entity relationship diagram.**

Above entity relationship diagram is based on the data dictionary given above. Here the entities are provided along with the relationship that they have with each other.

## **3.4 Architectural Model:**

A three-tier architecture is a client server architecture where functional process logic, data access, Computer data storage and user interface are developed and maintained. It is a well-established design pattern. User interface uses a graphical interface and implemented on pcs and laptops. Computer data storage is interlinked with relational database management system.



**Figure 14: 3 tier architecture.**

The three-tiers on three tier architecture are described below:

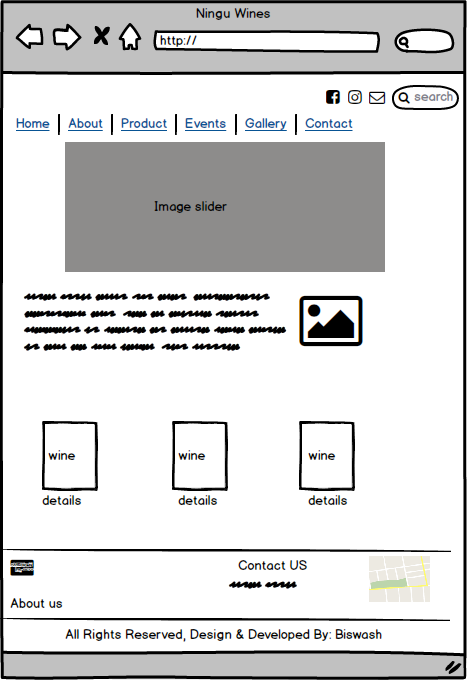
* **Presentation Tier:** It displays the data and information that are provided on websites.
* **Application Tier:** It control the functionality in application by performing detail processing.
* **Data Tier:** Here, the data are stored and are independent of application server.

# 3.5 Prototype Design:

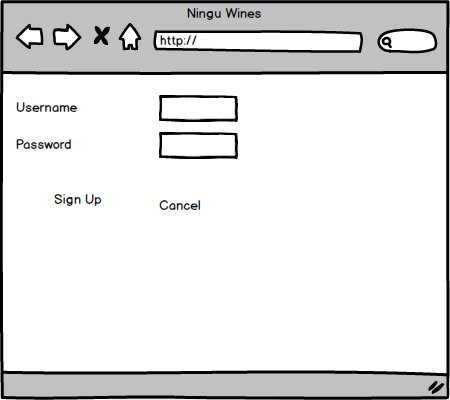
A prototype is a way of presenting an actual system into simpler handmade design. It gives designer an idea and to research in alternatives. It is designed to test and try new design by analyst and users. It gives the overview of system before it is implemented. In some model it is the step between formalization and evaluation of idea and concept of analyst and user.

Some of the benefits of prototype are discussed below:

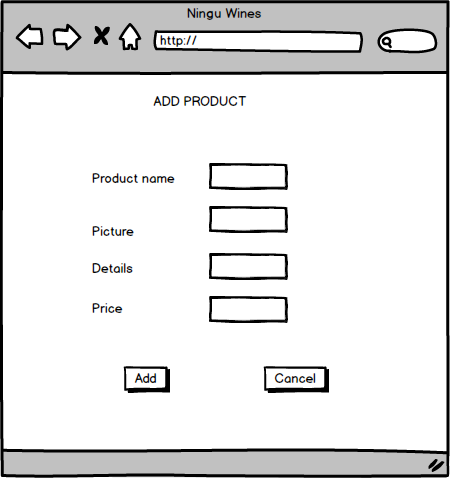
* It helps you to test and refine the overall functionality of your design.
* Almost all aspect of system is tested.
* Facilitates user for more efficient and effective description.
* Encourage analyst and user to take it seriously.



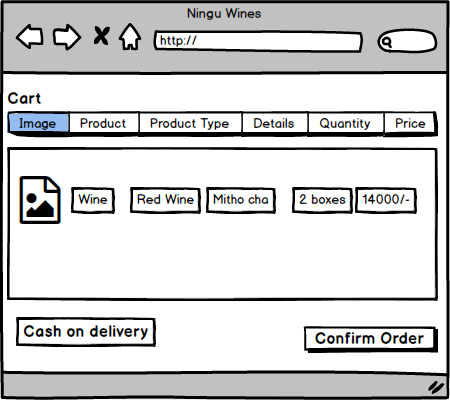
**Figure 15: prototype 1**



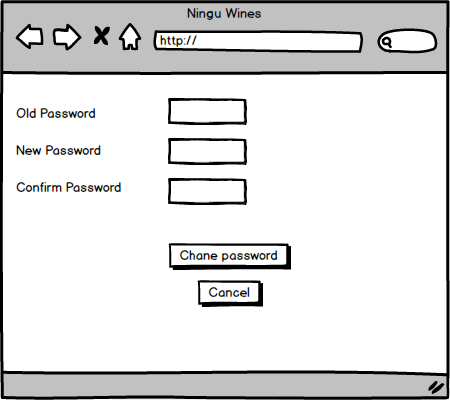
**Figure 16: prototype 2 signup page.**



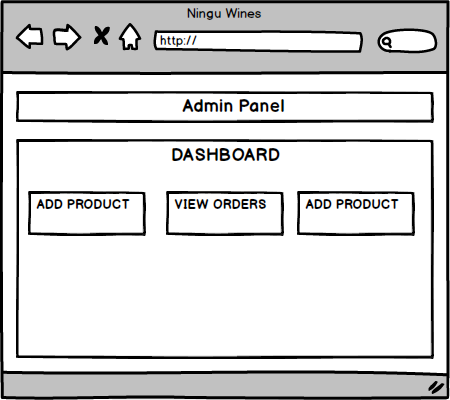
**Figure 17: prototype 3 add product.**



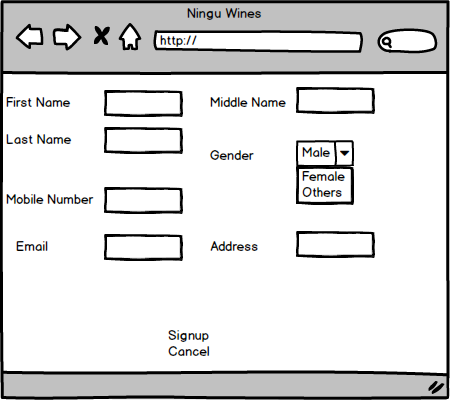
**Figure 18: prototype 4 add to cart.**



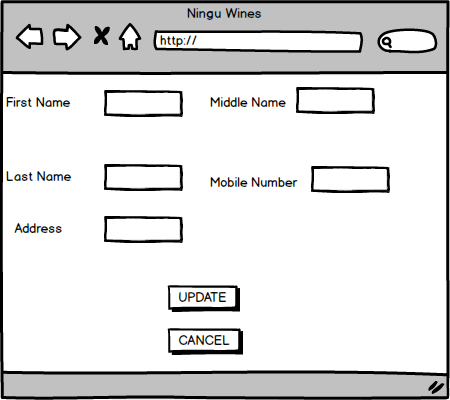
**Figure 19: prototype 5 reset password.**



**Figure 20: prototype 6 admin dashboard.**



**Figure 21: prototype 7 signup page.**



**Figure 22: prototype 8 login page.**