# Chapter 3 - Design

Design, this is the third and important phase while designing any application or a software which is intended to accomplish certain goals. Simply, Design is the process of transforming user’s requirement into some suitable form, which helps the programmer in software coding and implementation. (Anon., 2019) Designing phase helps to give us the answer of “**HOW**” question. We user various tools and techniques even models to develop the logical solutions which is later implemented in the system.

Some of the importance of design while developing my project are listed below:

1. It helps to understand the requirement in better way.
2. It helps to increase reusability and prevent redundancy.
3. It helps to allow both admin and users to realize how the application is going to function.
4. It helps to create class and objects obtained during analysis, which will eventually help to make the work of programmer easy.

For doing this phase, I have used **Star UML**, a software which is used to make different class, object, use case diagram, component diagram, etc. **Visual Paradigm,** a UML CASE tool for making ER Diagram. **Belsamiq Mockup 3,** a user interface designing tool for developing Paper Prototypes.

## 3.1) Structural Modeling

Models that help to show the different component along with their relationship that are organized in a system is said to be structural modelling. Structural Modeling helps to show the static relationship of different components which are present in a system.

### 3.1.1) Final Class Diagram

Class is a blueprint of an object that defines what object can do. Class Diagram is a diagram that gives static view of a system. It helps describe the type of object and its relationship that exist among the system. By the help of this diagram almost all method can run with Object Oriented Models. (TutorialsPoint, 2017)

Purposes of class diagram are listed below:

1. Analysis and design of system in static view.
2. Helps to describe the responsibility of the system.
3. Acts as base component diagram and also helps in forward and reverse engineering.

### 3.1.2) Flow Chart

Flow chart is a formalized graphic representation of a diagram containing different symbols having its own information about steps or a sequence of works. Each of the works are linked with arrows to show the flow of the process.

Importance of creating flow chart for my project is listed below:

* Communication: It helps to explain the logic of a system to all the users, whether he/she is skilled or not.
* Proper Documentation: It serves as a good program documentation, which helps to make things more effective and efficient.
* Efficient Coding: Flow chart helps as a guide during the development of an application.
* Proper Debugging: As it helps in good communication for explaining the logic of a system, debugging process becomes fast.

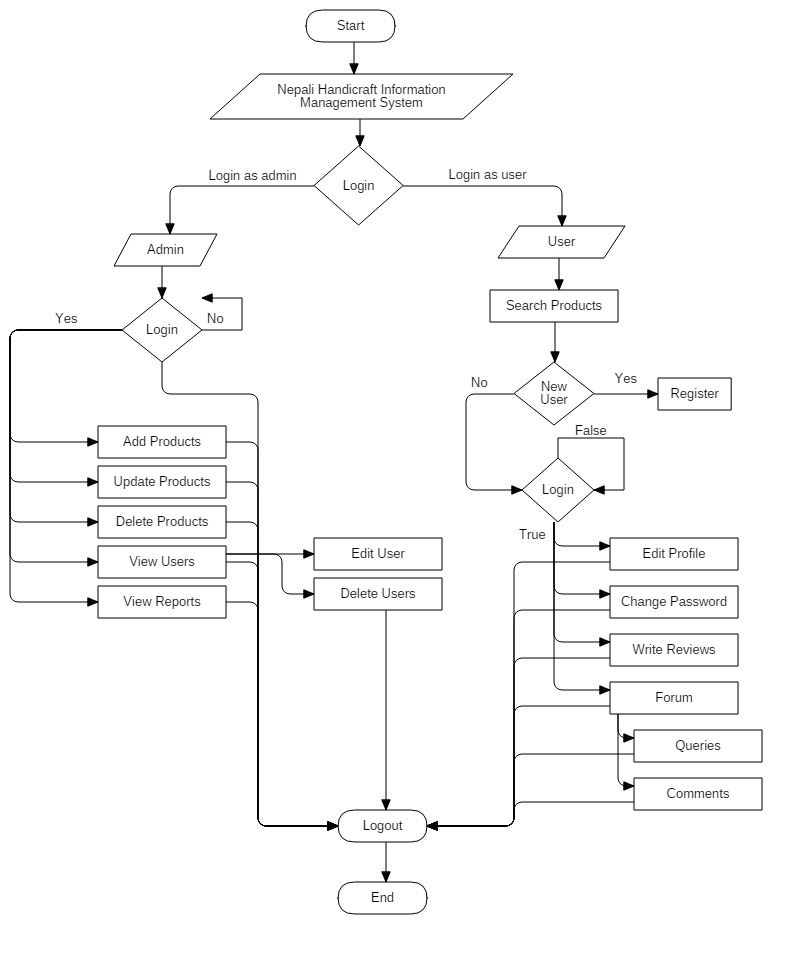


Figure 1: Flow Chart of a System

In above diagram, Admin can easily login to dashboard. Before going into the dashboard, Admin need to correctly type username and password. Once it is correct then admin is directed to the homepage else login is failed. After redirecting to the homepage of Admin, S/He can manipulate products, edit and delete any registered user from the database. For Users, one can easily search the products after or before login but cannot buy a product once user logs out. After logging into the system, user can edit their profile, change password, write a reviews and even ask any queries and comment in the forum section.

## 3.2) Behavioral Modeling

The type of modeling that models the dynamic behavior of the system to produce a specific behavior often designated through a use case is said to be behavioral modeling. It shows how objects engage with each other to provide a specific behavior.

### 3.2.1) Activity Diagram

Activity Diagram is a representation of the flow of activity from one to another, which helps to describe the dynamic aspects of the system.

Importance of activity diagram for my project is given below:

* It shows different steps involve in the UML use case.
* It helps to represent multiple conditionals activities easily.
* It can be easily understood by any people or the end users.

**Notations used for making Activity Diagram**

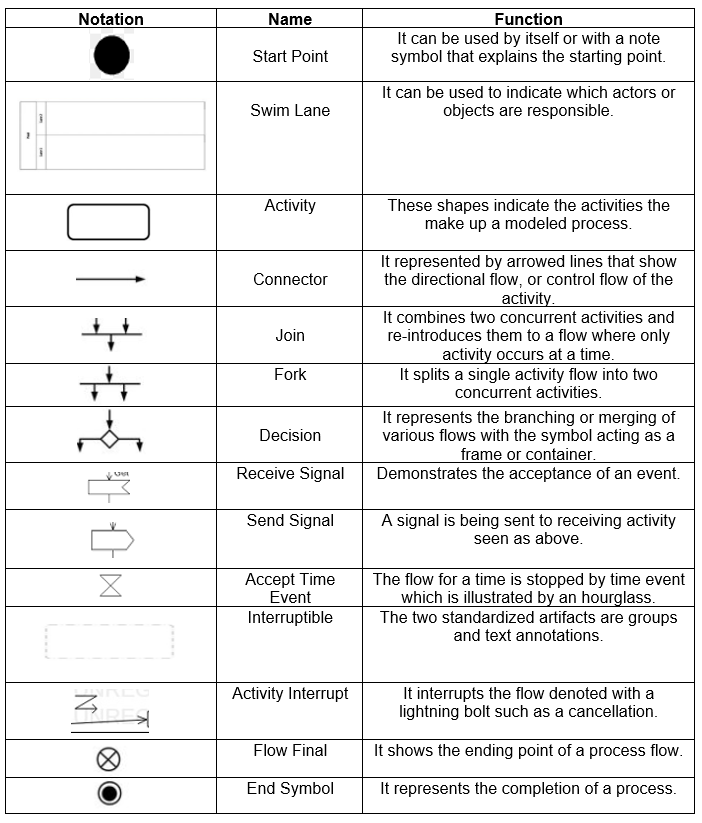


Figure 2: Notation used for Activity Diagram

**Login and Registration Activity Diagram**

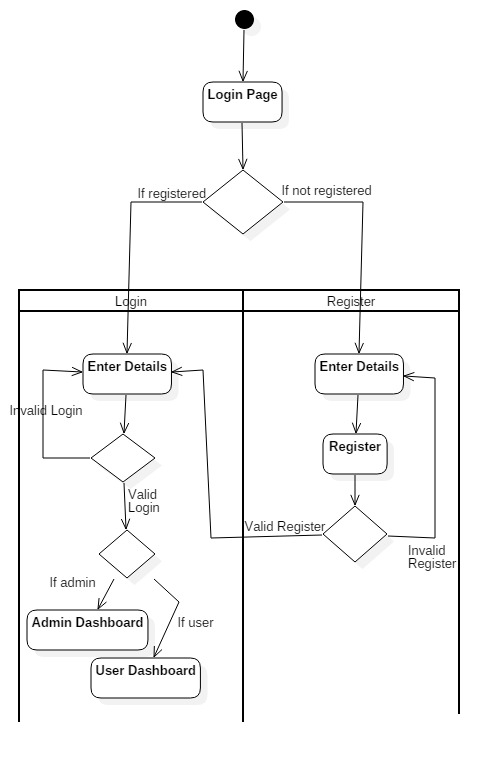


Figure 3: Login and Registration Activity Diagram

In above diagram, the activity of user starts from Login page. Then decision notation will help to take a decision whether a user is registered or not. If a user is registered, one will open login form else register form. For not registered user, Once register form is opened he/she has to enter their details in the form and proceed their process to register. After that decision is taken, if user provides valid information then user is successfully registered else warnings is shown. After registering, user is redirected to the login form and user will be asked to fill the login form. In case of incorrect email or password, error message is displayed or it will be redirected to the corresponding dashboard i.e., Admin Dashboard or User Dashboard.

**Activity Diagram of Admin**

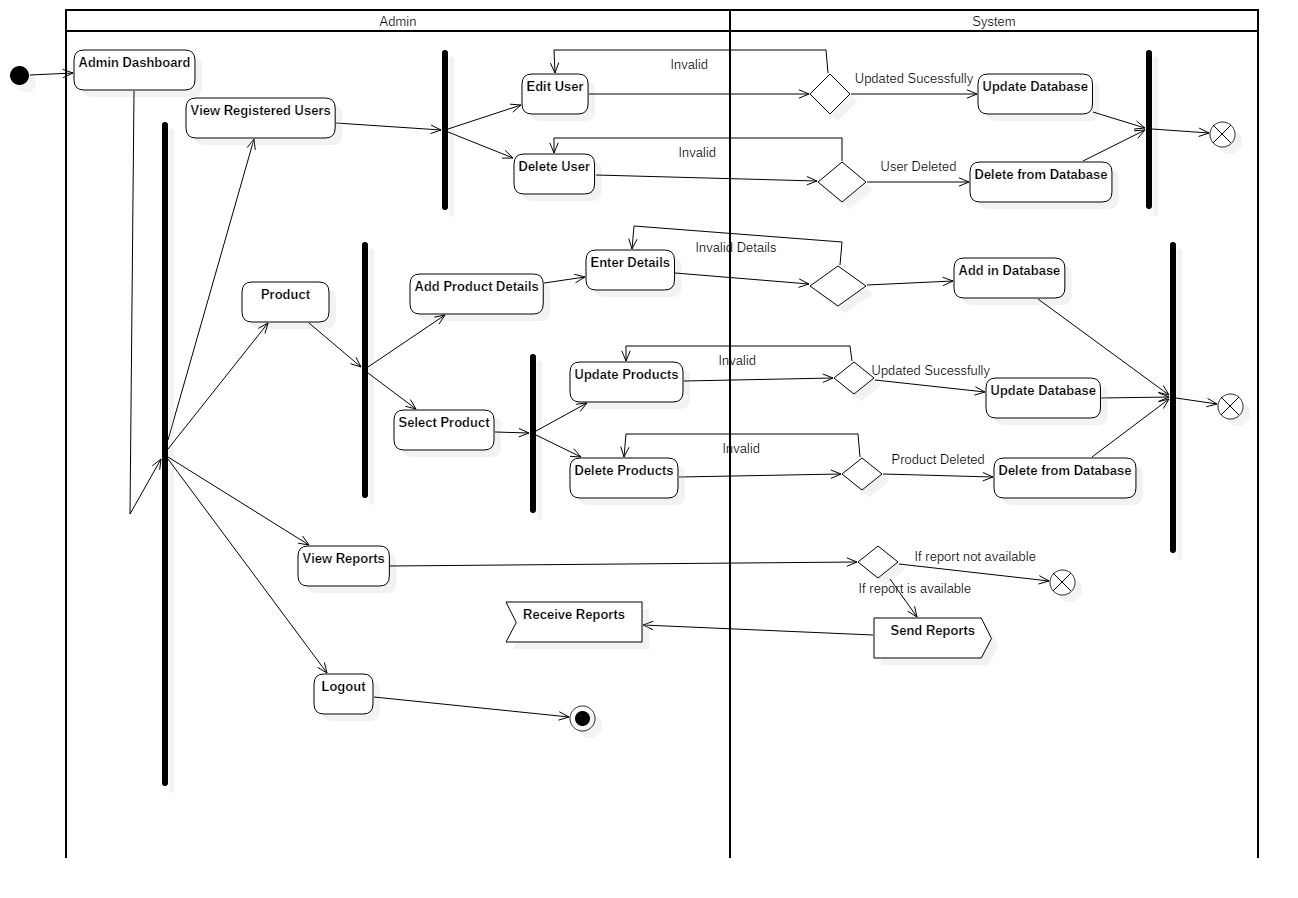


Figure 4: Activity Diagram of admin

Above given diagram explains all the activities that can be performed by Admin of the system. Once admin successfully logins to the system, admin can perform following tasks as shown by Fork Notation. That is, Admin can view registered users, Manipulate Products, View Reports and Get out of the system. Further, Admin can edit user and delete users in the database with the permission granted from the system. Admin can view reports in database with permission from the system, if reports are not available then process is ended else message is send from system to admin for receiving a message.

**Activity Diagram of User**

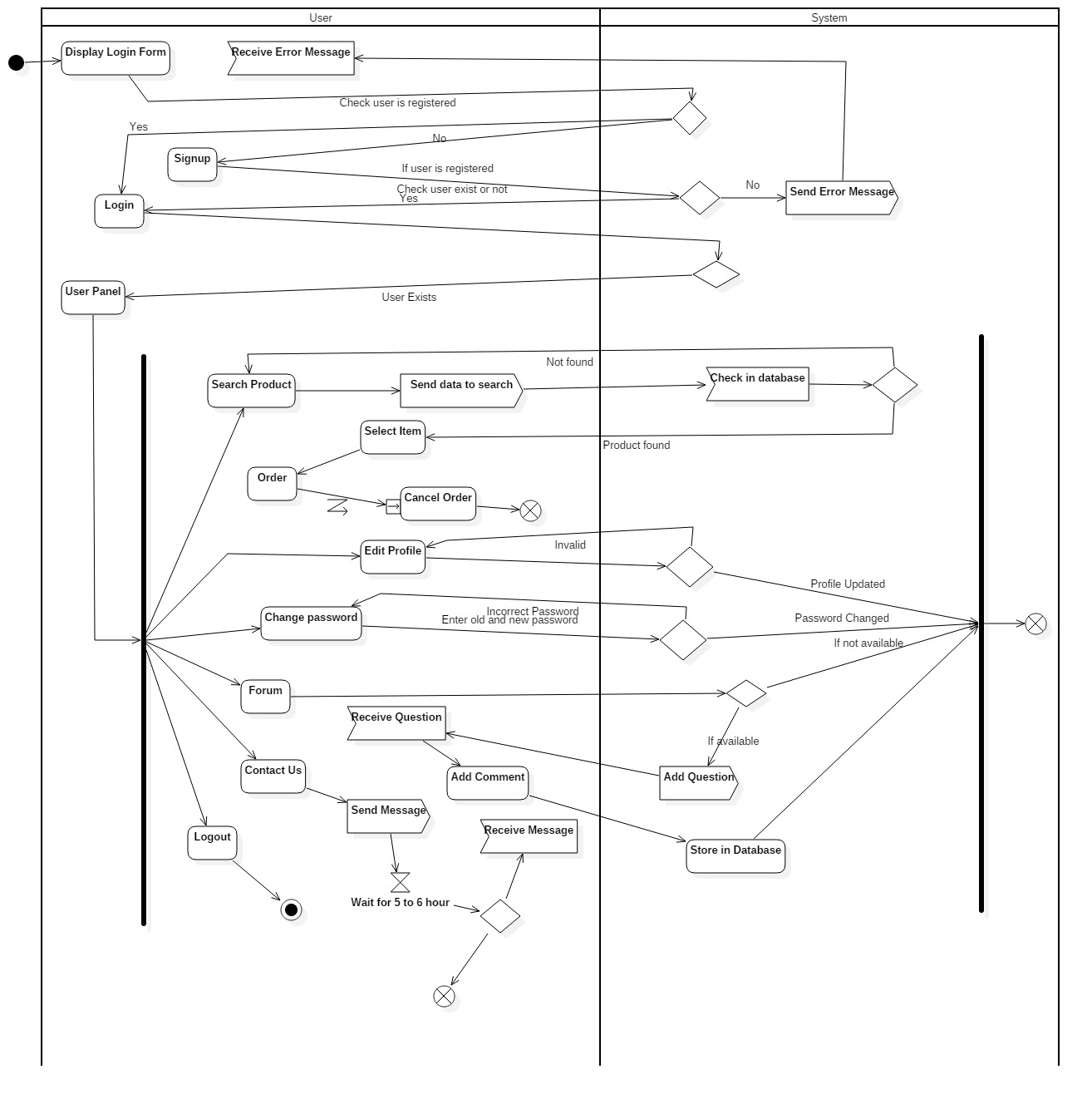


Figure 5: Activity Diagram of User

In the diagram above, user will be displayed with a login form, through decision notation whether users are registered or not it is checked. If user is registered, then login form is opened and if user is not registered error message is displayed from the system and that message is received by the user. Once user is successfully registered, user will be navigated to their panel. Users can do the following activities denoted by fork notation in the diagram. User can search product, after searching product user can select item they require and even cancel the order. User can change their password, edit profile, contact admin for any queries and get out of the system whenever they like. In forum section, user can ask a question about products and also reply as a comment if they want.

### 3.2.2) Sequence Diagram

Sequence diagram is a diagram that helps to represent the interaction between any objects in a sequential order. It helps to describe how and what order the objects in a system function. It shows the logical flow of the system is a good way.

Importance of this diagram for my project are listed below:

* Helps to understand the detailed function of the system.
* Helps to model and visualize the logic behind advance function or operation.

**Notations used for making Sequence Diagram**

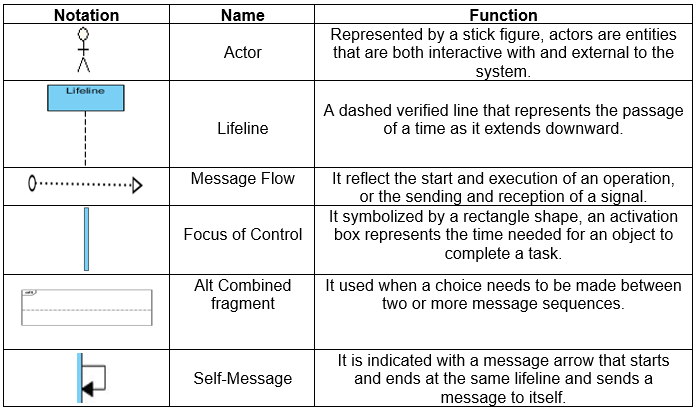


Figure 6: Notation used for making Sequence Diagram

**Registration Sequence Diagram**

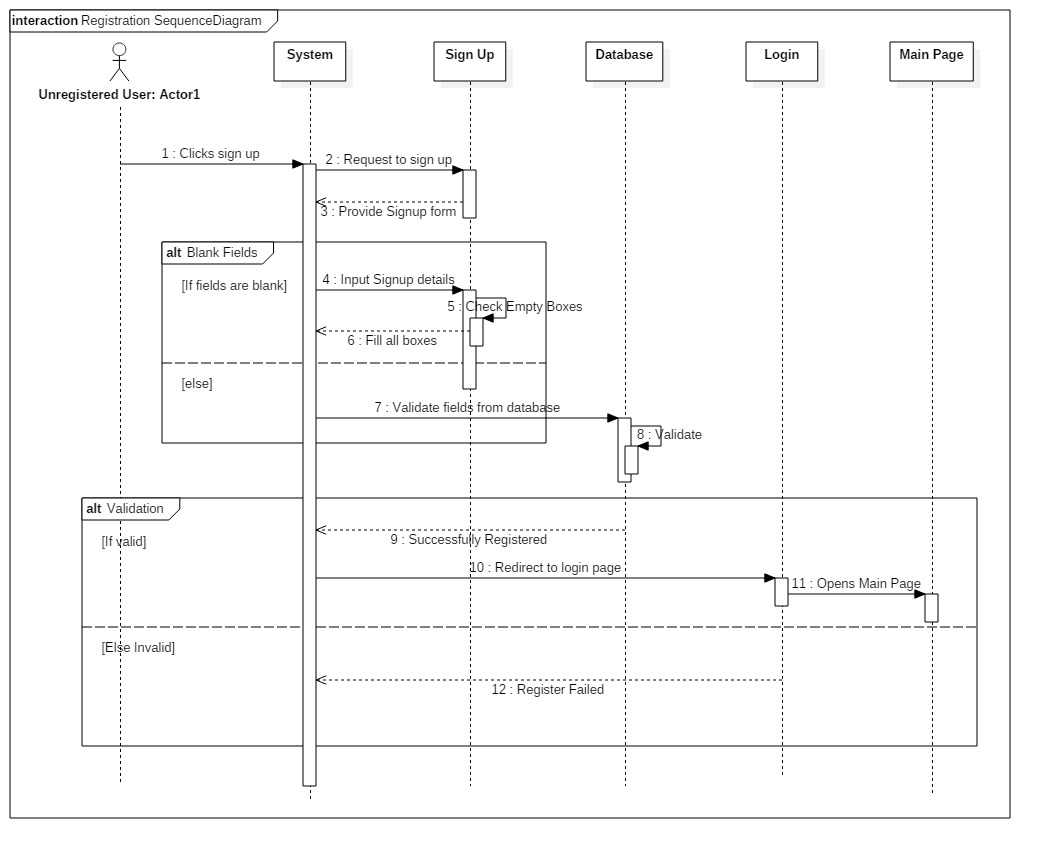


Figure 7: Registration Sequence Diagram

From the diagram above, at first user will click on sign up button, system will request for a signup form. Now user will be asked to input their details, during this process if user lefts any empty field user will see error message and if all the fields are filled properly satisfying the validation rules then user is created and stored into the database. After successfully signup user is now redirected to the login page for further actions.

**Login Sequence Diagram**

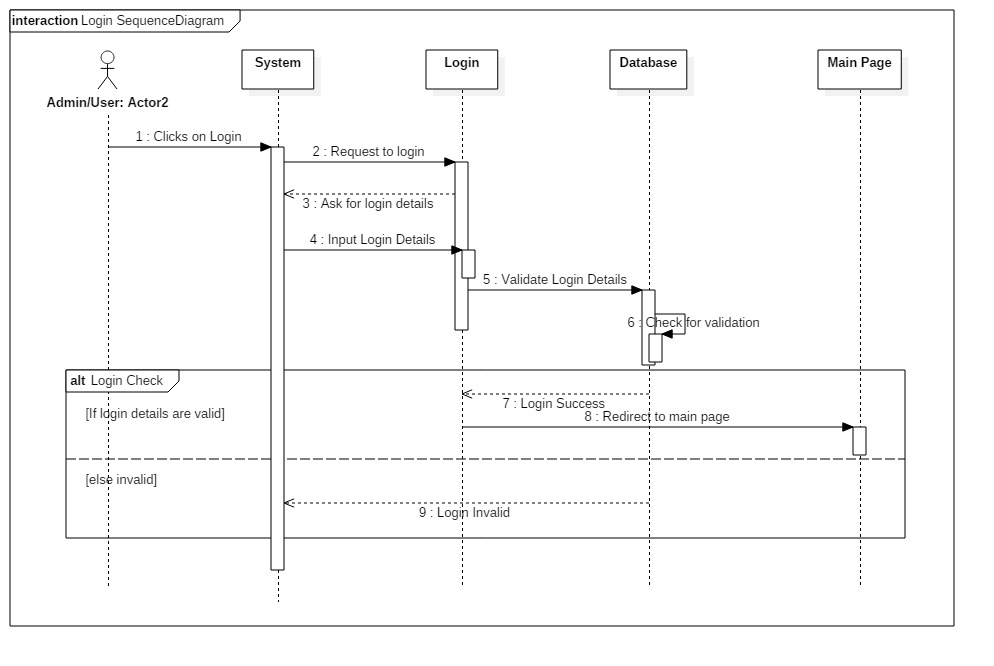


Figure 8: Login Sequence Diagram

After successfully registration of a user, user can now login. For that user can click on Login and system will request for a login form and then user will enter their required credentials. After entering the credentials, is validate from database. If the required credentials matches, user is redirected to main page or user dashboard else invalid login message is displayed.

**Admin Privilege to user**

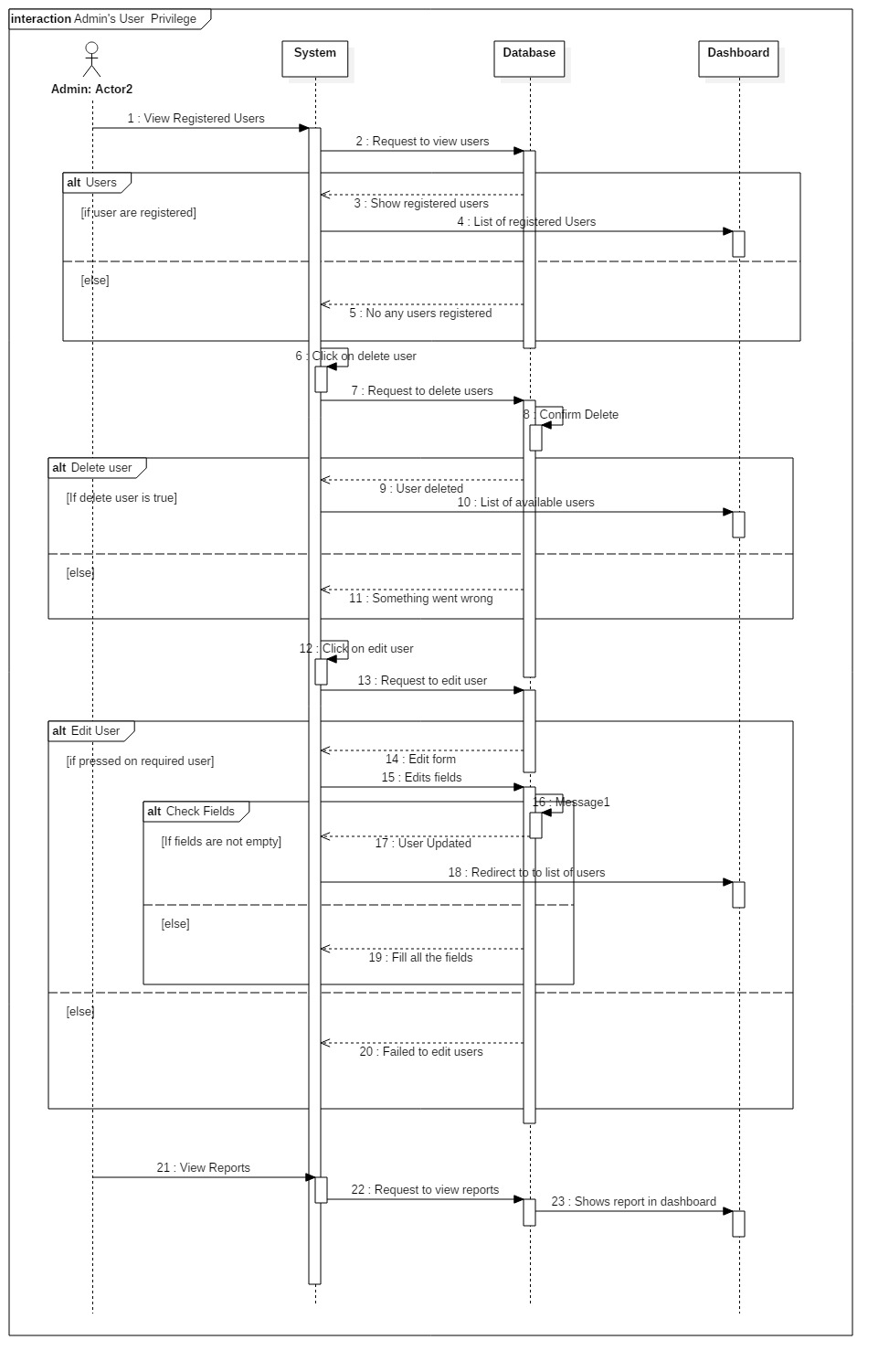


Figure 9: Admin Privilege for user

**Sequence Diagram of Products**

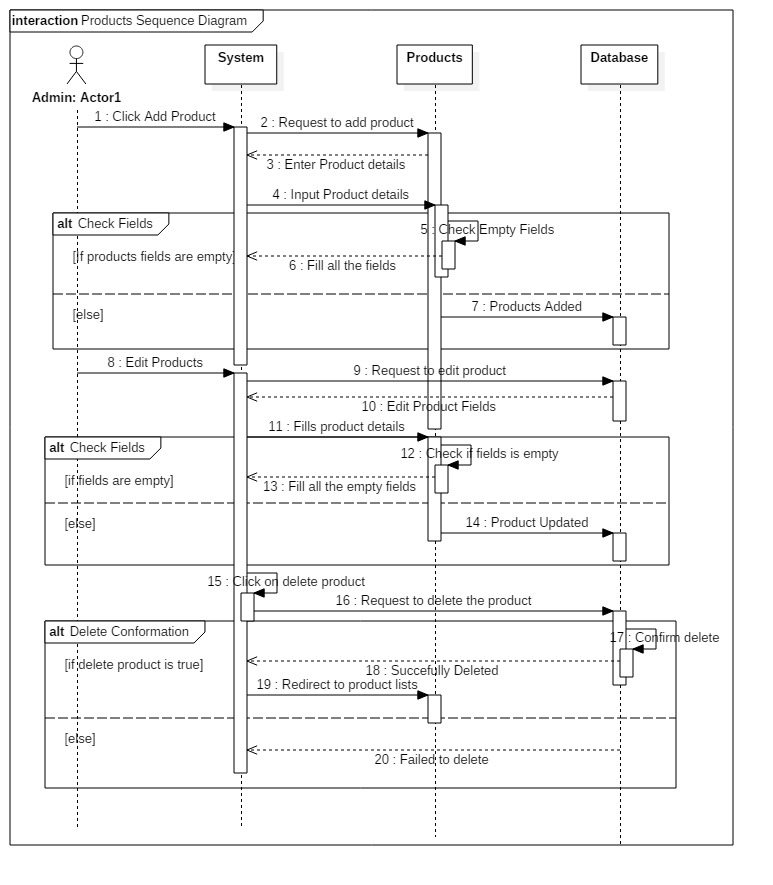


Figure 10: Sequence Diagram of Products

From Figure 9 and 10, These two diagram represent all the actions or the functions which is performed by admin. Admin request to view registered users from the system, system request from database if user are registered in the database then lists of all the registered users are seen else no any users are seen. If admin wish to delete any registered user from the system then, S/He will click on delete user, after clicking system will request its permission and if permissions and conditions are okay then selected user is deleted from the database else error message is displayed.

Admin willing to add product, can click on add product and system requests for a product to add. Then admin will receive an add product form, after filling all the products details empty fields are checked. If found admin will be displayed error else product is successfully added in the database. Also for deleting product admin has to select the required product from the product list and if the confirmation goes as planned then product is deleted else error message is displayed to the system. For updating product, system request to select product from database, product edit form is opened for admin to make changes to the product. If any fields are left empty, error message is displayed else product is successfully updated and stored into the database.

**User Sequence Diagram**

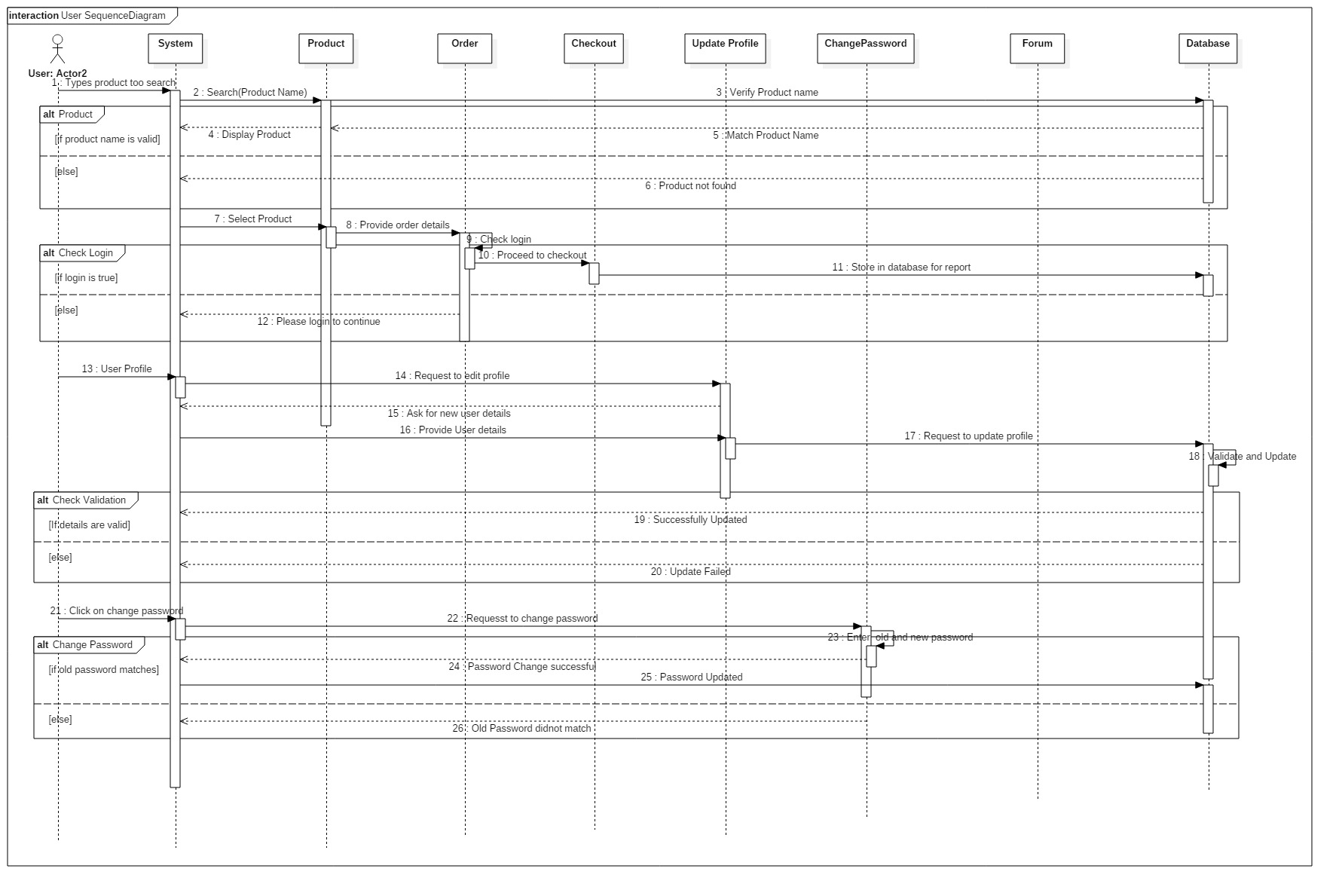


Figure 11: User Sequence Diagram

From the diagram above, User can easily search product, system requests the typed keywords of user from the database. If the keyword matches the product which is in database, then it is displayed else not. If displayed is selected user login is checked. User can only proceed to checkout only if user has logged in to the system else user should first login to the system and only continue the process. For editing the user’s profile, one need to login into the system and system will request to edit profile. Then update profile form is displayed from the system, user need to enter the boxes without leaving any fields. If user fails to fill all the boxes, error message is displayed else profile is successfully updated. User can change their password too, for doing that user need to click on change password. After doing that system will request to change a password, system is replied with change password form having current password, new password and re-password in it. Here, only if old password matches then password is successfully changed or the error message is displayed.

## 3.3) Database Modeling

It is the type of data model that helps to determine all the necessary and important logical structure of a database. It fundamentally determines in which way a data should be stored, organized and manipulated.

### 3.3.1) Data Dictionary

Simply, Data Dictionary means data of a data. It is the collection of information describing about the contents, format and structure of a database.

Importance of Data Dictionary in my project are listed below:

* It holds ownership of a data.
* It provides detail information about data elements.
* It helps to figure out the basic data structures required for the system.
* It helps to prevent data duplication; each entry field is identified for faster performance.

**User Register Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| u\_id | int(11) | Yes | No | No |
| fname | varchar(50) | No | No | Yes |
| lname | varchar(50) | No | No | Yes |
| e\_mail | varchar(50) | No | No | Yes |
| telephone | bigint(50) | No | No | Yes |
| address | varchar(50) | No | No | Yes |
| city | varchar(50) | No | No | Yes |
| country | enum | No | No | Yes |
| region | enum | No | No | Yes |
| gender | enum | No | No | Yes |
| password | varchar(255) | No | No | Yes |
| confirm\_passowrd | varchar(255) | No | No | Yes |

**Admin Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| a\_id | int(11) | Yes | No | No |
| username | varchar(255) | No | No | Yes |
| password | varchar(255) | No | No | Yes |

**Product Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| p\_id | int(11) | Yes | No | No |
| a\_id | int(11) | No | Yes | No |
| pname | varchar(50) | No | No | Yes |
| description | varchar(50) | No | No | Yes |
| image | varchar(255) | No | No | Yes |
| original\_price | double | No | No | Yes |
| product\_price | double | No | No | Yes |
| date | date | No | No | Yes |

**Order Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| o\_id | int(11) | Yes | No | No |
| u\_id | int(11) | No | Yes | No |
| p\_id | int(11) | No | Yes | No |
| date | date | No | No | Yes |

**Forum Table Question**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| fq\_id | int(11) | Yes | No | No |
| u\_id | int(11) | No | Yes | No |
| question | text | No | No | Yes |
| date | date | No | No | Yes |

**Forum Table Comment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| fc\_id | int(11) | Yes | No | No |
| u\_id | int(11) | No | Yes | No |
| fq\_id | int(11) | No | Yes | No |
| comment | text | No | No | Yes |
| date | date | No | No | Yes |

**Forum Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| f\_id | int(11) | Yes | No | No |
| fq\_id | int(11) | No | Yes | No |
| fc\_id | int(11) | No | Yes | No |

### 3.3.2) ER Diagram

Entity Relationship Diagram (ERD) is a data modeling technique that graphically represents a system’s entities and relationship. It is a backbone for developing a database design.

Importance of making ERD in my project are listed below:

* Base for designing a database design.
* Database will be well documented.
* Helps to know the relationship including their cardinality.

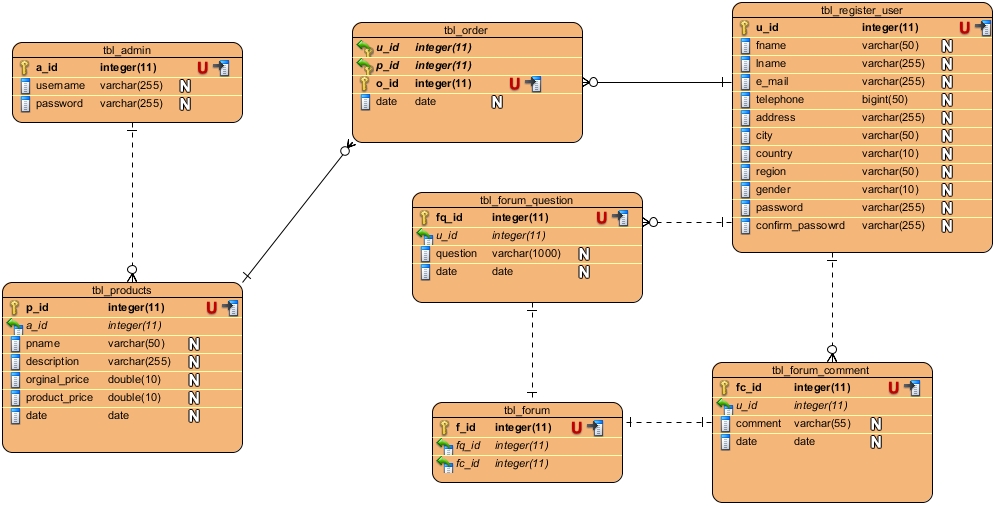


Figure 12: ER Diagram of a System

## 3.4) Architectural Design

**Client-Server Architecture:**

It is an architecture of a computer network, in which the client requests a service from a centralized server and the centralized server gives service to a remote processor or a client-server. Advantages of client server architecture are:

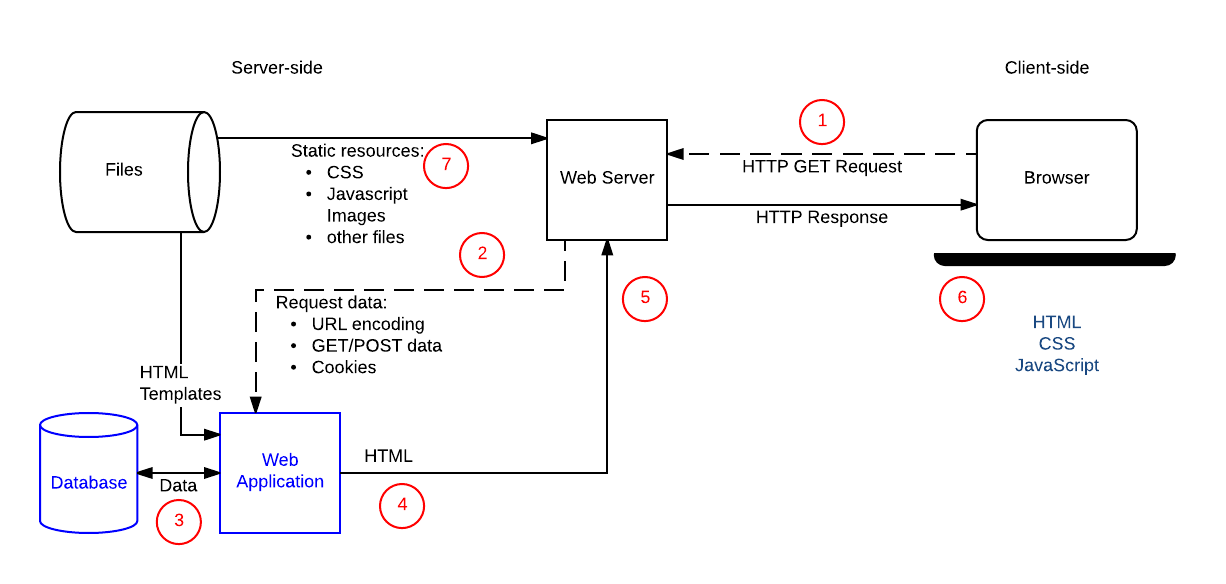
* It helps to share resources among client and servers easily.
* It reduces data of client by storing data on server instead.

Figure: Client Server Architecture

Comparing above figure with my current project, first of all user will send HTTP GET request to the server side through web server. Travelling through web server it requests URL encoding, GET/POST data or cookies from web application and that web application will ask data from database. If it is image files or other files it will be pulled from another part of server side, which is file in above diagram. After requesting files from database by web application, our requested data is transferred to web server again and Web Server will send the HTTP Response back to the browser or client. And Client will be able to see its requested response in a fraction of a time.

## 3.5) Paper Prototyping

Paper Prototyping is one type of prototyping method in which paper models are used to test concept of system. In this project I am using Mockup Prototyping, which is useful for visually conception of the system. Following are the mockups of my project.

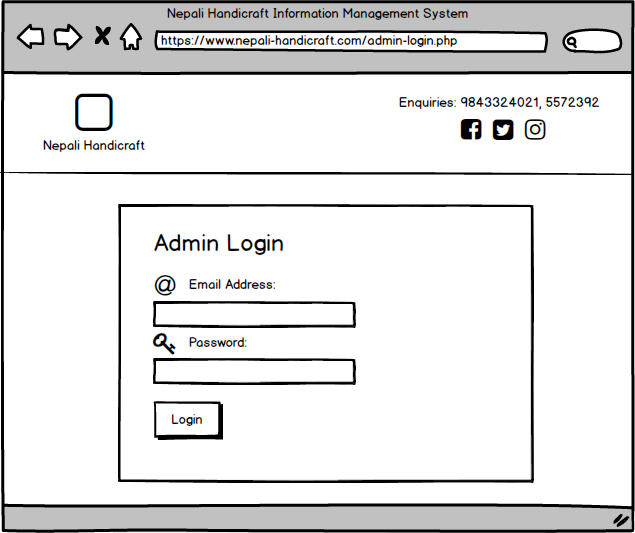


Fig: Admin Login

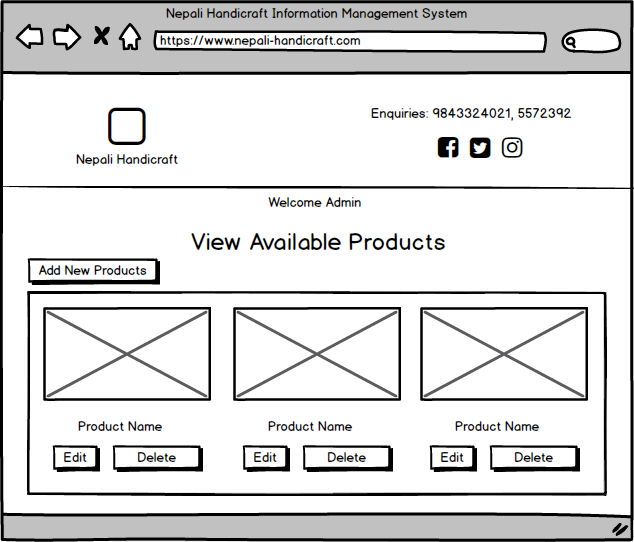


Fig: View Available Products

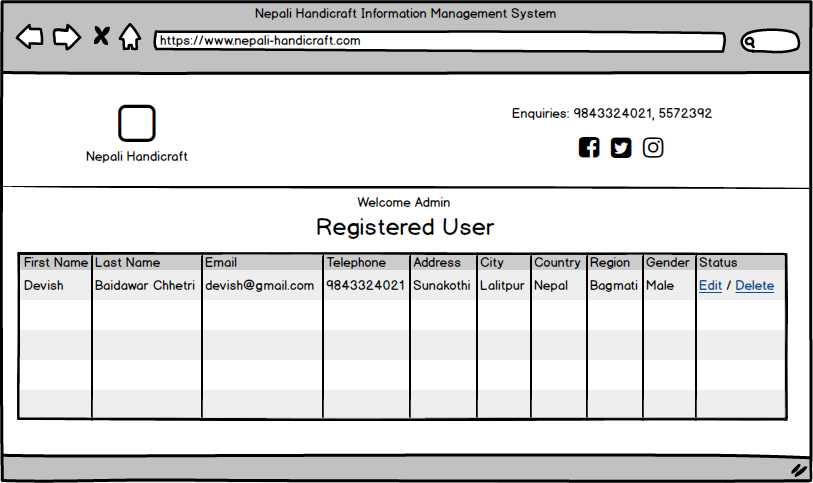


Fig: View Registered Users

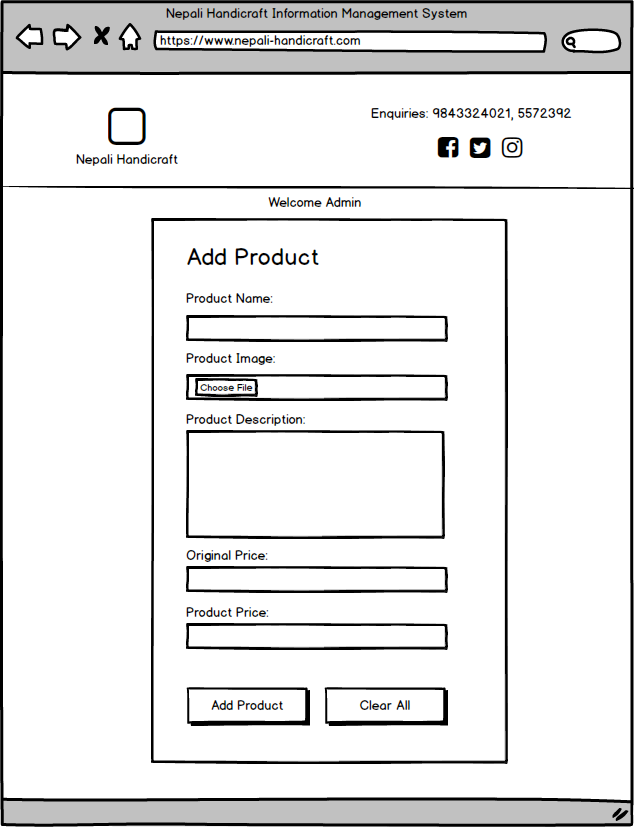


Fig: Add Product

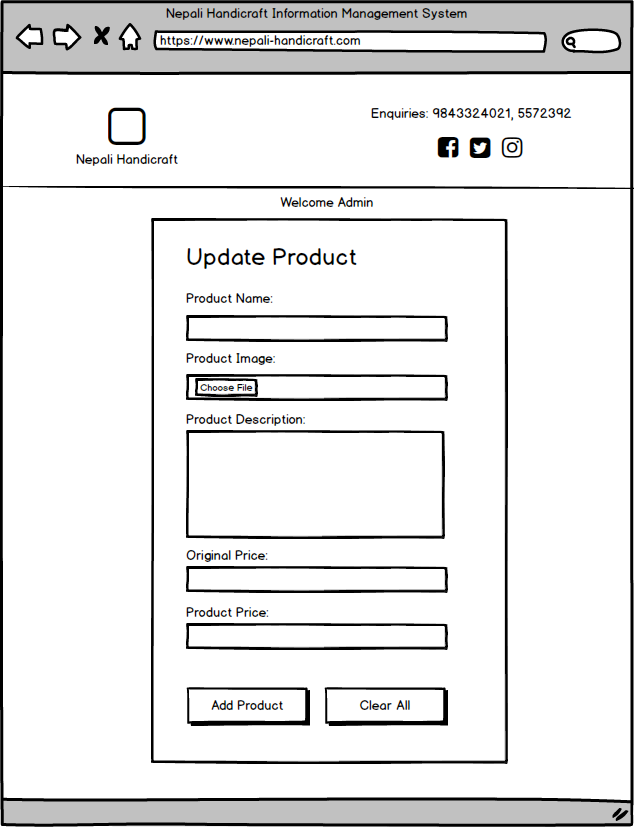


Fig: Update Product

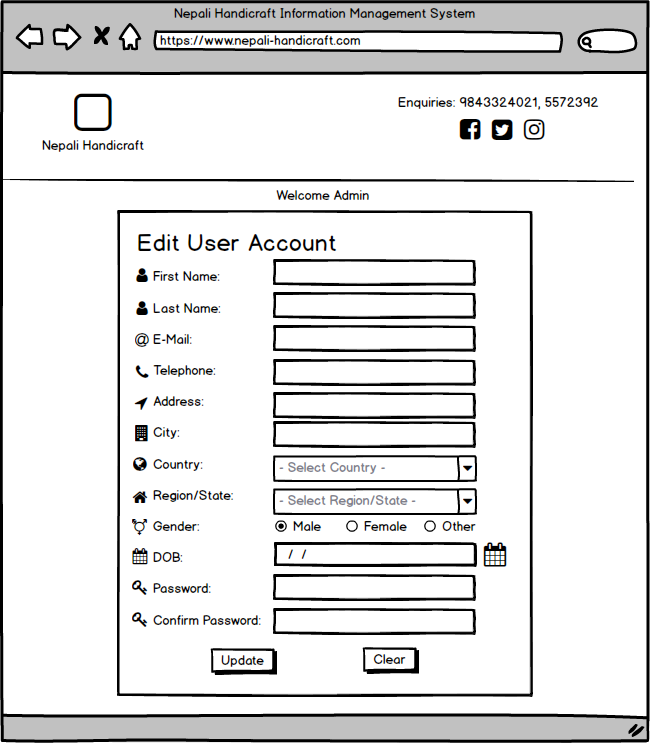


Fig: Admin edit user

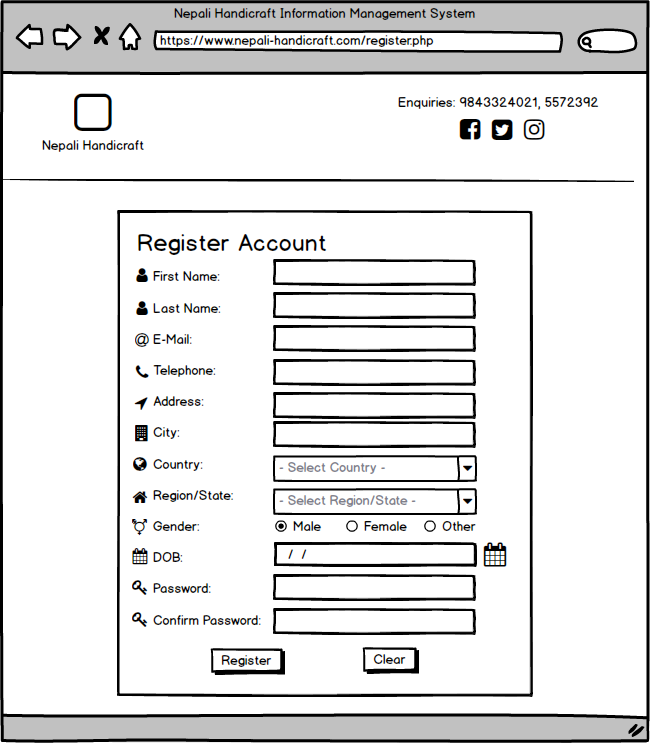


Fig: Register New user

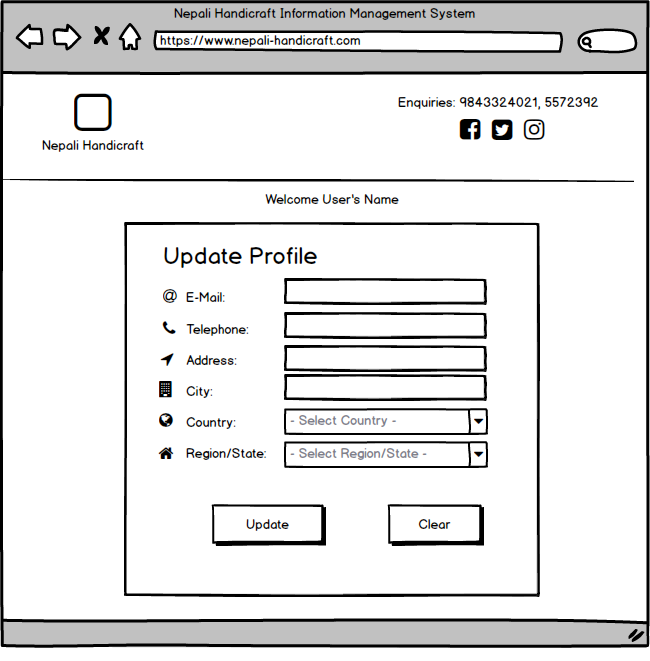


Fig: Update User’s Details

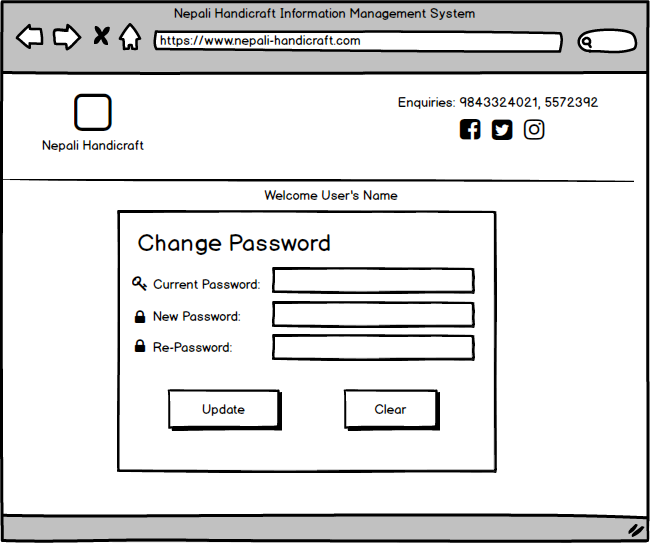


Fig: Change Password

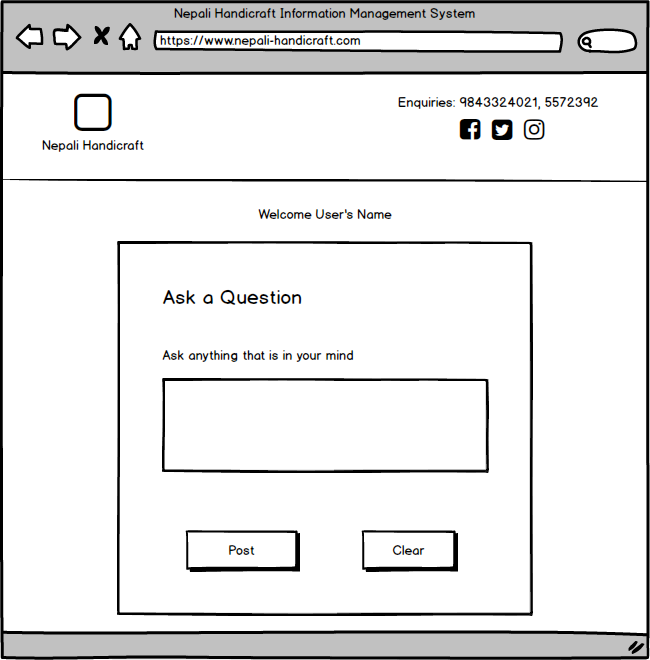


Fig: Forum Ask Question

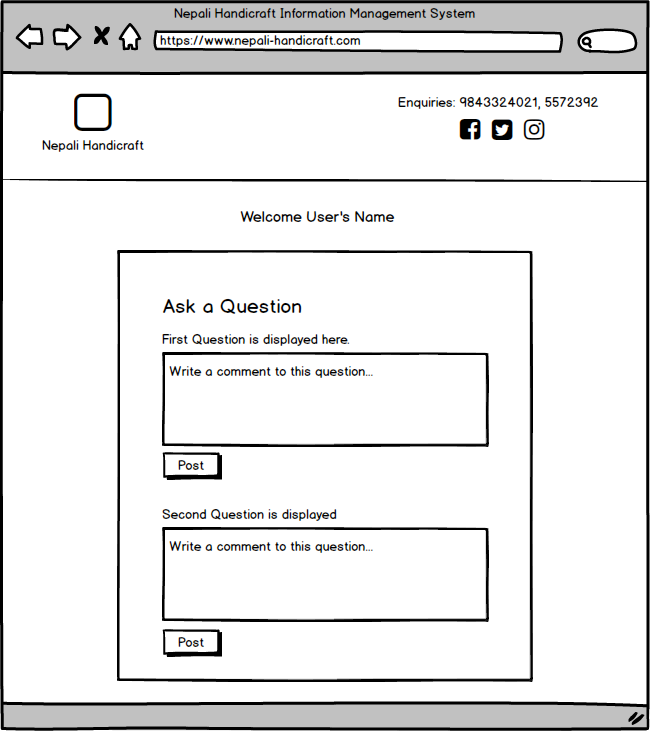


Fig: Forum Comment

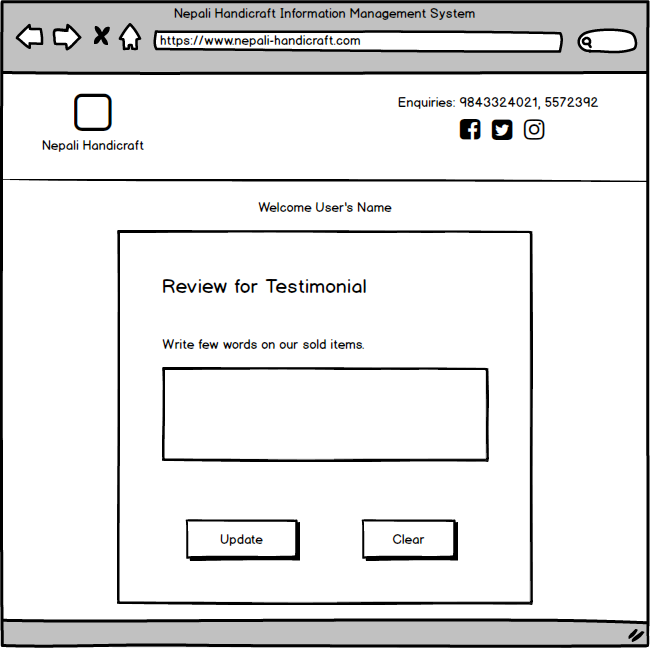


Fig: Review for testimonial

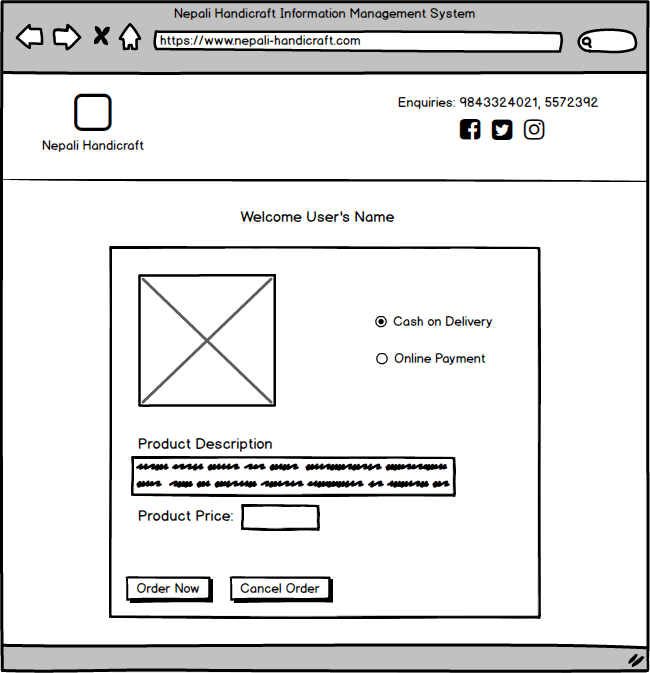


Fig: Order Required Product

# References

Anon., 2019. *Tutorialpoint.* [Online]   
Available at: https://www.tutorialspoint.com/software\_engineering/software\_design\_basics.htm  
[Accessed 15 May 2019].

TutorialsPoint, 2017. *UML Class Diagram.* [Online]   
Available at: https://www.tutorialspoint.com/uml/uml\_class\_diagram.htm  
[Accessed 7 May 2019].