# **Design**

## **3.1 Introduction**

In design phase the SDLC process continues to move from the “what” question of the analysis to the how. We use various tools and techniques, models to develop logical solution that might be present in the form of paper or design tools which later on is implemented in the real system. It helps to describe the systematic sequence of steps to produce a project. It lets us be fixated, crystal clear while developing the project. Design process are charted in order to avoid any blunders of task or functions. The structure of the project is already scheduled/planned and how the project can accomplish target.

Design is important because:

* It provides the framework to ensure that all the aspects of the project are properly and consistently defined, planned and communicated.
* It makes it easier to evaluate each part of software development.
* It established architecture that describes a plan to implement the requirements.

I used open software Star UML which is fast, flexible. I used Visual paradigm to develop ER diagram.

## 3.1) Structural Modeling

### 3.1.1) Final Class Diagram

### 3.1.2) Flow Chart

Below is the data flow diagram showing how user interact with the system:

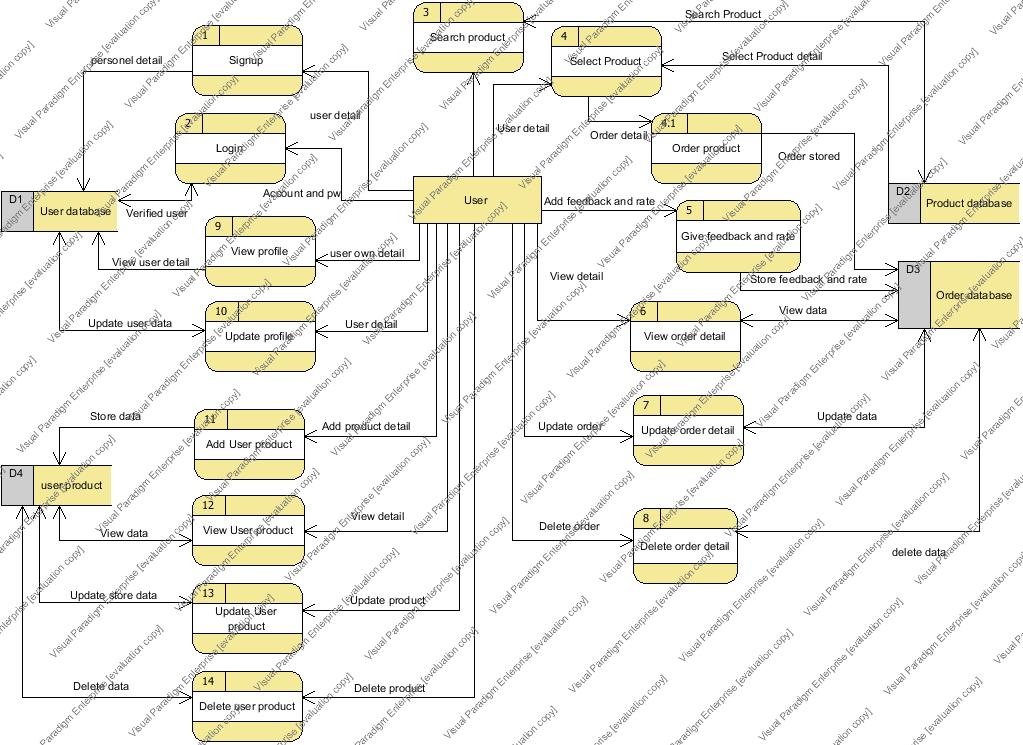


Figure 1**:** DFD of user data flow

Below is the data flow diagram showing how Admin interact with the system:

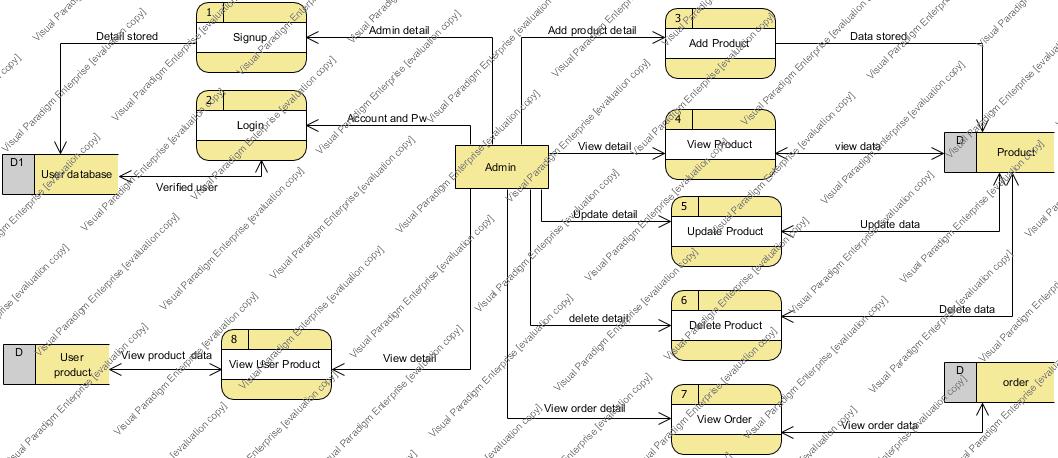


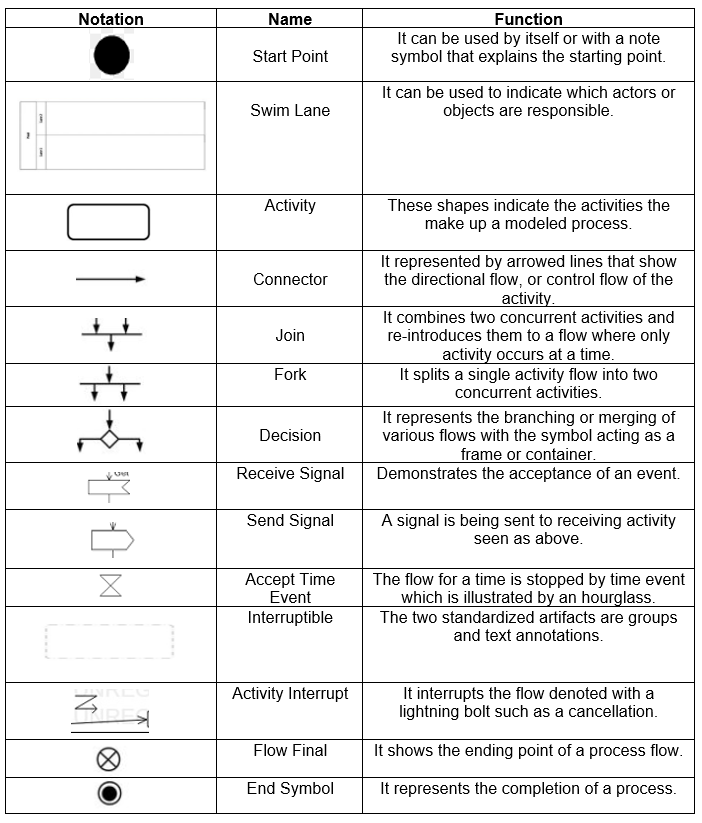
Figure 2**:** DFD of admin data flow

## 3.2) Behavioral Modeling

Behavioral describe internal logic of system. This model deals with communication between objects. It denote dynamic behavior of system. It helps to show CRUD operation of the system through diagram. It helps us to know about guidelines, process and strategies of system. There are many behavioral model, among which I choose: Sequence and activity diagram.

### 3.2.1) Activity Diagram

A UML that graphically represent serial flow of control and several actions in the system is called Activity diagram. It visualize flow of action from one activity to another. I choose activity diagram because it is simple and intuitive illustration of how the system workflow take place. The dynamic aspects of our system can be describe by activity diagram. I can also view the decision paths that exist in system. It describe the functionality that take place in our system. (Software, 2018)



Below is the activity diagram of user interacting with system:

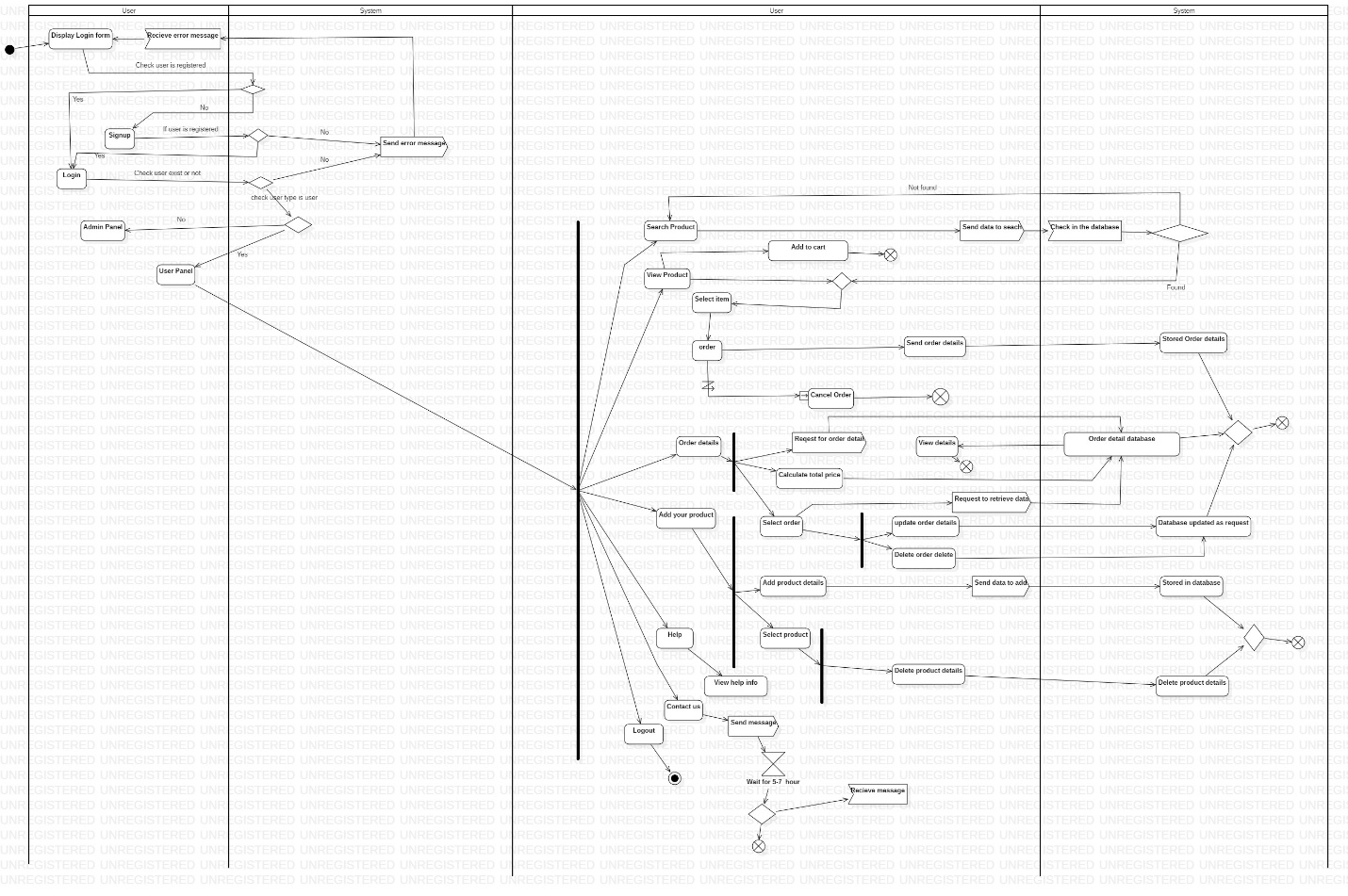


Figure 3: Activity diagram of user action

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.

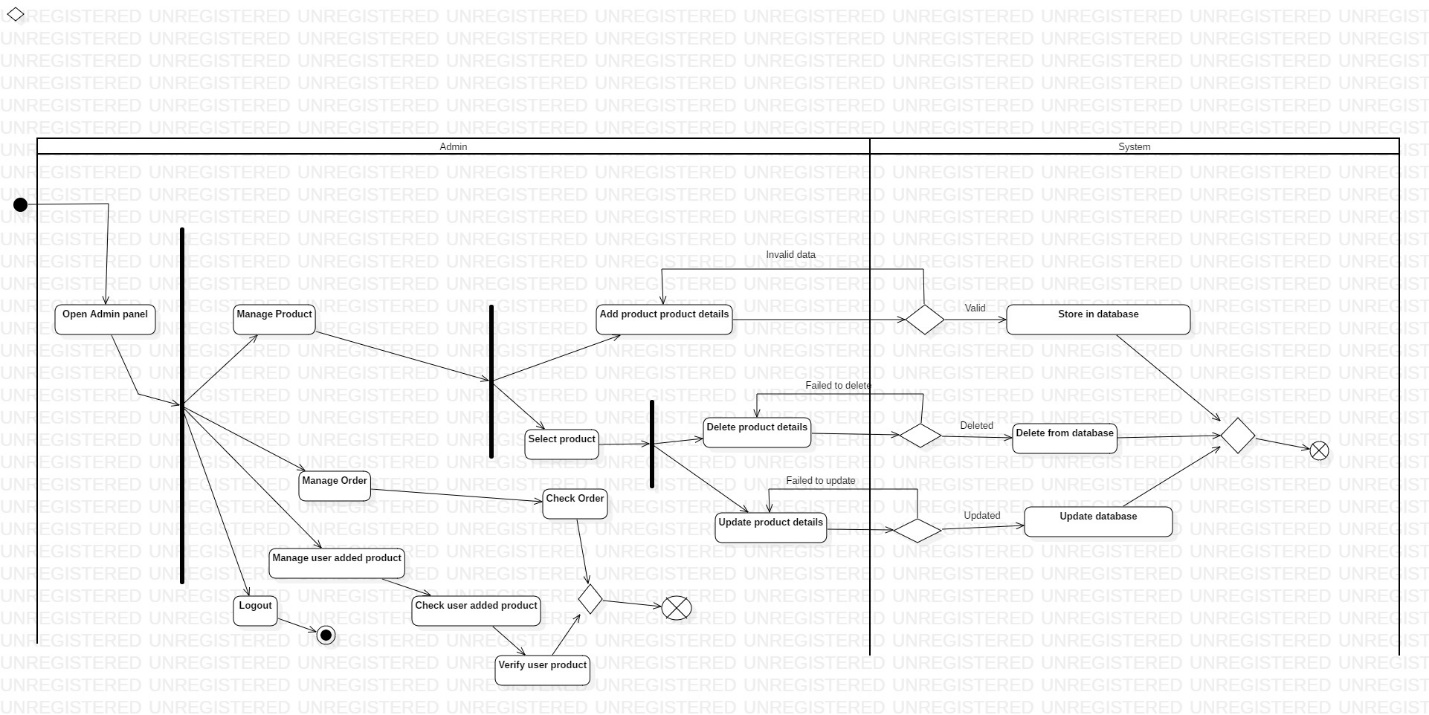


Figure 4**:** Activity diagram of admin action

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.

### 3.2.2) Sequence Diagram

A UML that describe how the system operate with one another in a sequence is called sequence diagram. It shows the interaction of objects in a time sequence. It helps to visualize the different runtime significances.

The following are the Sequence Diagram where the admin interacts with system like how admin open dashboard, add product, manipulated product which are saved in product DB, view order detail and user product detail.

**Notations used for making Sequence Diagram**

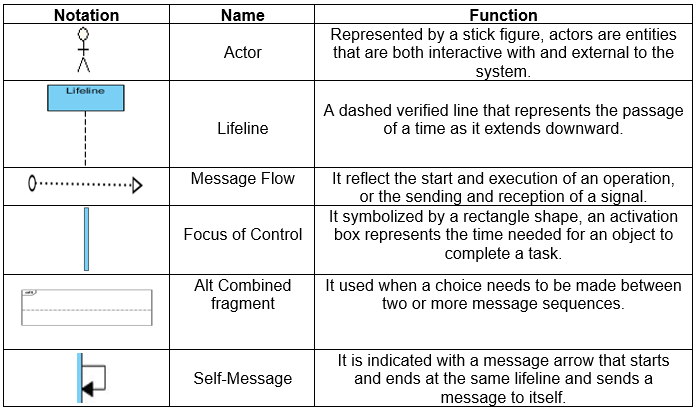


Figure 6: Notation used for making Sequence Diagram

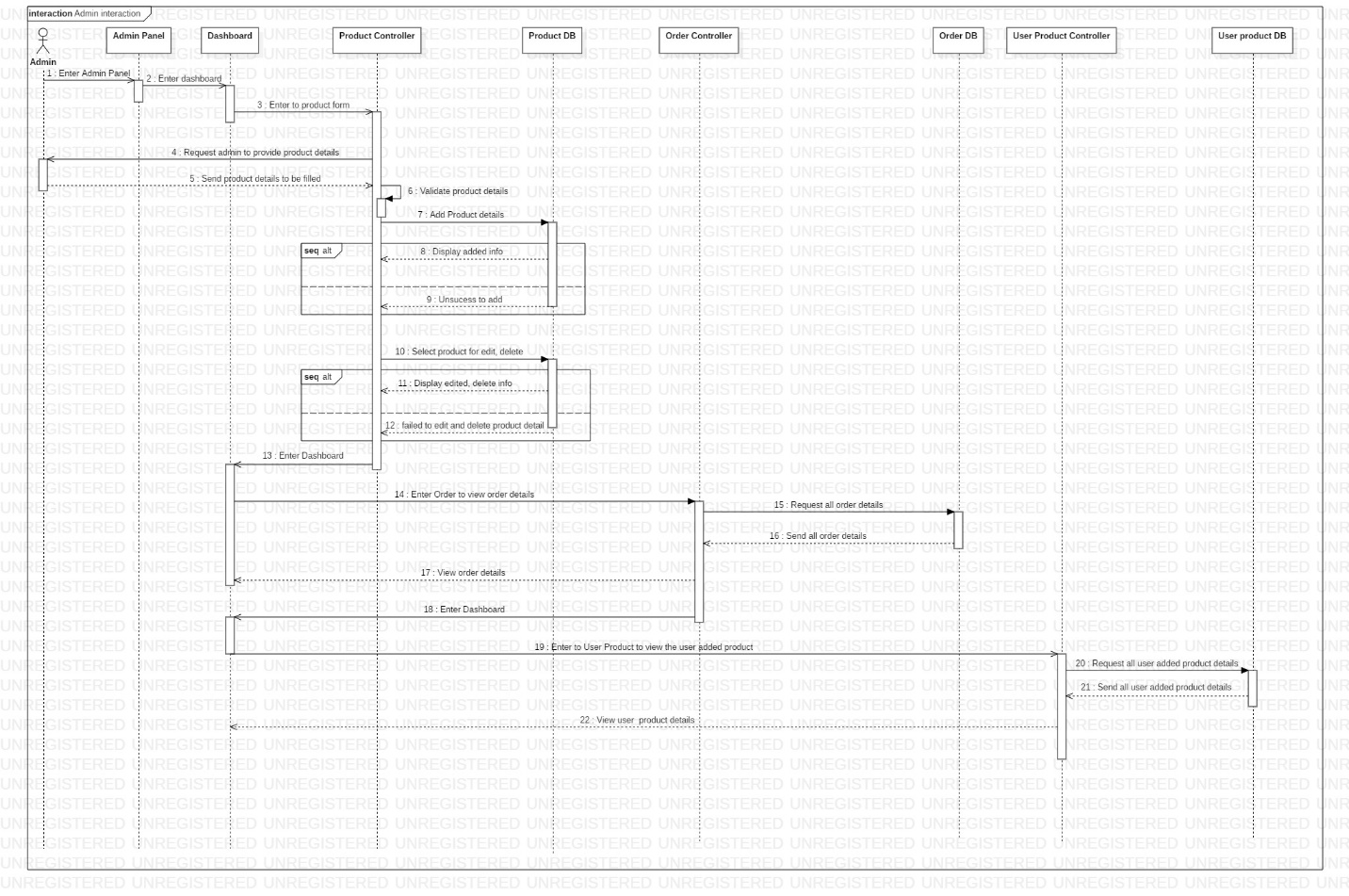


Figure 5: Sequence diagram on how admin interacts

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.

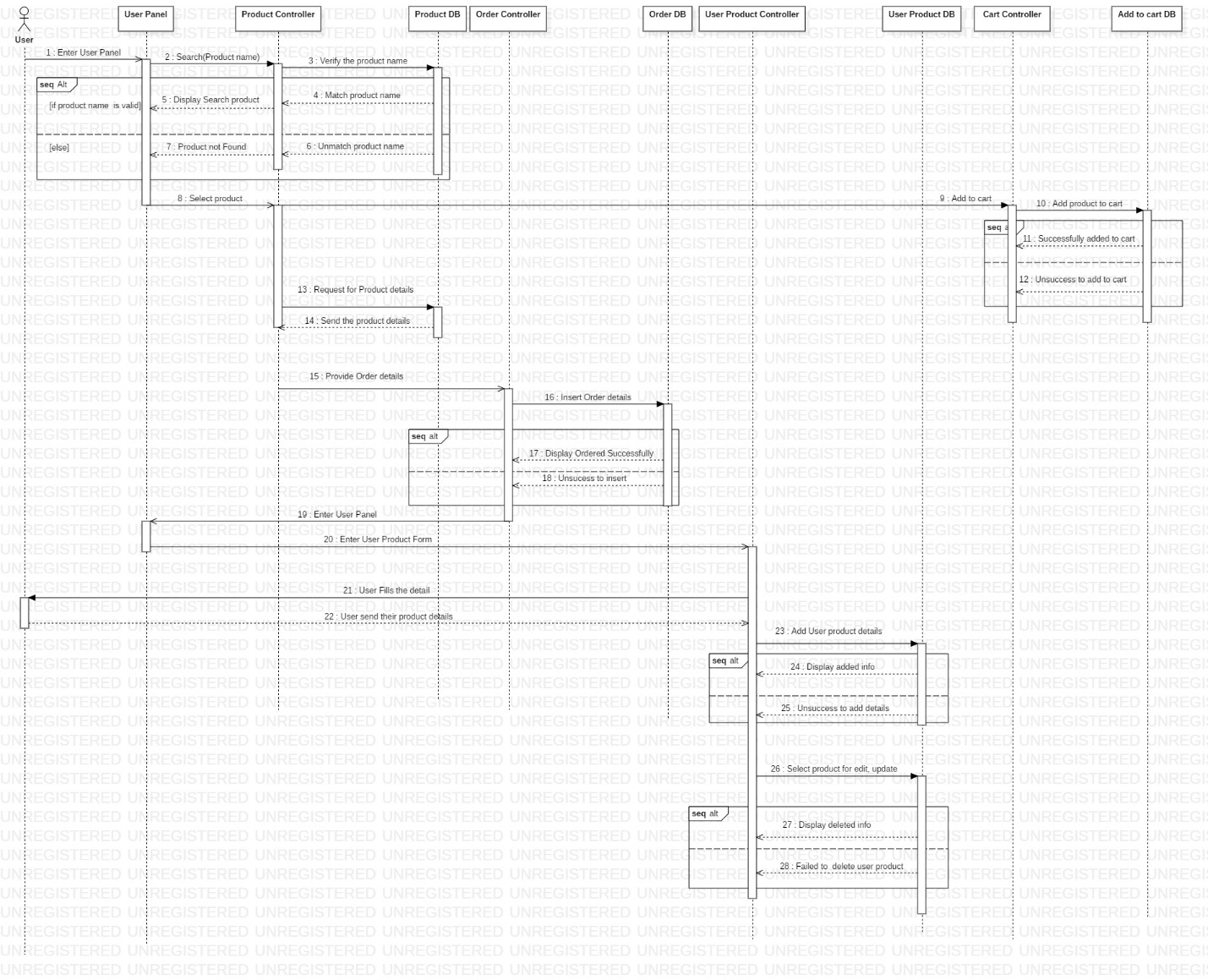


Figure 6: Sequence diagram of user interacting

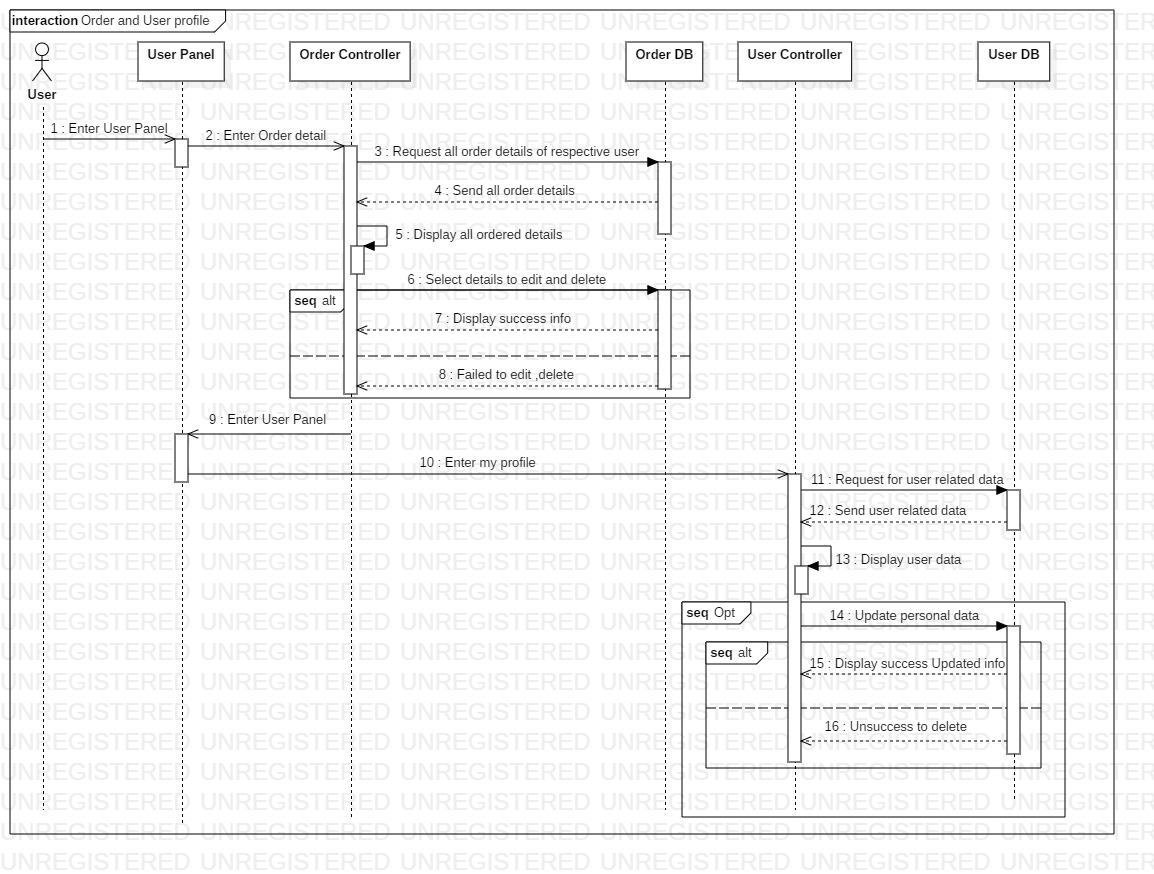


Figure 7: Sequence diagram on user interacting to view order detail and update profile

Below is the sequence diagram of user doing login:

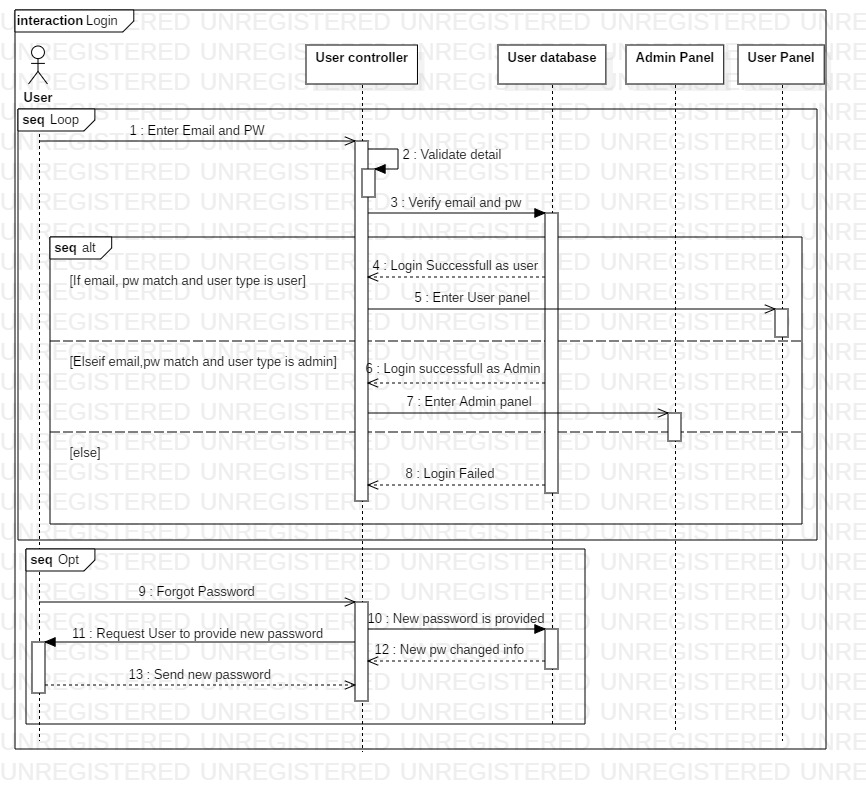


Figure 8: Sequence diagram on user interact while login

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.

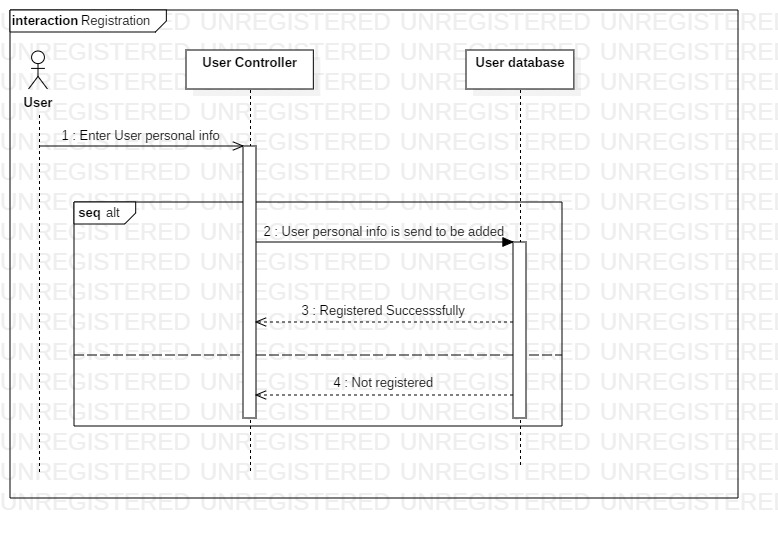
Below is the sequence diagram showing registration:

Figure 9: Sequence diagram of user interacting while doing registration

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.

## 3.3) Database Modeling

### **3.3.1)** Database **Model:**

Database Model is the process which involves in managing and placing the data to its database system that determines the logical structure of database. It includes relationships and constraints that determines the way data is stored and accessed.

1. Entity Relation diagram: An entity relationship is the pictorial representation of physical structure of database with its model through use of many entities, relationship and attributes among them. It can be represented in different forms like: Crows Feet, UML or CHEN notation.

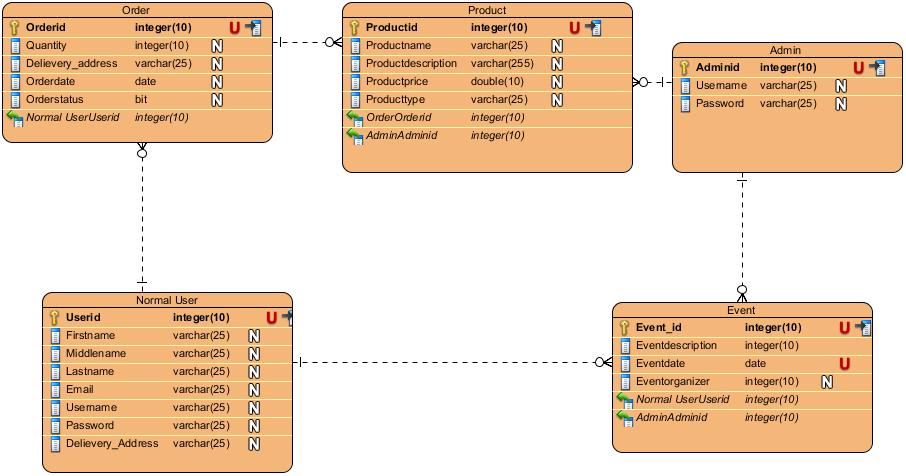


Figure Entity Relationship Diagram

### 3.3.2) Data Dictionary

The table which contains many set of information that clarifies how the database is designed is known as data dictionary. It includes names, functions of data elements and related explanation and many other contents. The data dictionary is used to handle the entrance and to retain and handle the database.

The following are the data dictionary for my project:

Normal User Information Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Userid | integer(10) | Yes | No | No |
| Firstname | varchar(25) | No | No | Yes |
| Middlename | varchar(25) | No | No | Yes |
| Lastname | varchar(25) | No | No | Yes |
| Email | varchar(25) | No | No | Yes |
| Username | varchar(25) | No | No | Yes |
| Password | varchar(25) | No | No | Yes |
| Delievery\_Address | varchar(25) | No | No | Yes |

Admin Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Adminid | int(10) | Yes | No | No |
| Username | varchar(25) | No | No | Yes |
| Password | varchar(25) | No | No | Yes |

Product Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Productid | int (10) | Yes | No | No |
| Adminid | int(10) | No | Yes | No |
| Orderid | int(10) | No | Yes | No |
| Productname | varchar(25) | No | No | Yes |
| Productdescription | varchar(255) | No | No | Yes |
| Productprice | double(10) | No | No | Yes |
| Producttype | Varchar(25) | No | No | Yes |

Order Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| order\_id | int(10) | Yes | No | No |
| Userid | int(10) | No | Yes | No |
| Productid | int(10) | No | Yes | No |
| Quantity | integer | No | No | Yes |
| Delievery\_Address | Varchar(25) | No | No | Yes |
| Orderdate | Date | No | No | Yes |
| Orderstatus | bit | No | No | Yes |

Event table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Event\_id | int(10) | Yes | No | No |
| Userid | Integer(10) | No | Yes | No |
| Adminid | Integer(10) | No | Yes | No |
| Eventdescription | int(10) | No | Yes | No |
| Eventdate | date | No | No | Yes |
| Eventorganizer | Integer(10) | No | No | Yes |

## 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:

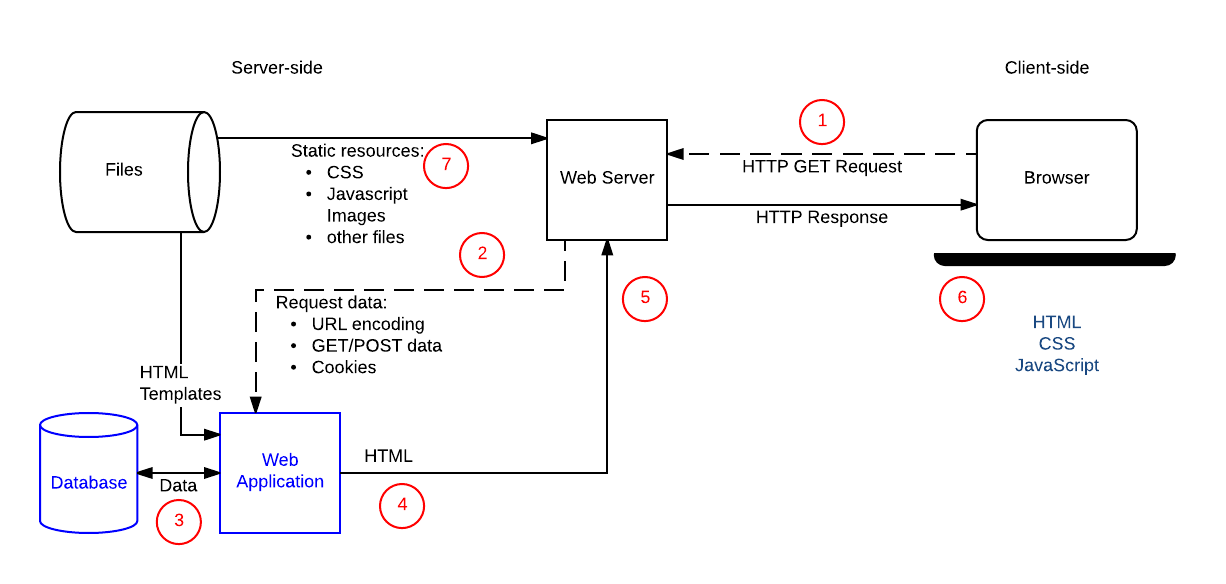
* It helps to share resources among client and servers easily.
* It reduces data of client by storing data on server instead.

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.

