# **Design**

## The Design Phase describes how the system will fulfill the user requirements. To achieve this, we must create both a logical design and physical design. In this phase the systems design functions and operations are described in detail, including storyboard and screen layouts with annotations, business rules, process diagrams and other documentation. The output of this stage will describe the new system as a collection of modules or subsystems.

# **Structural Design**

### **Final Class Diagram**

### **Flow Chart**

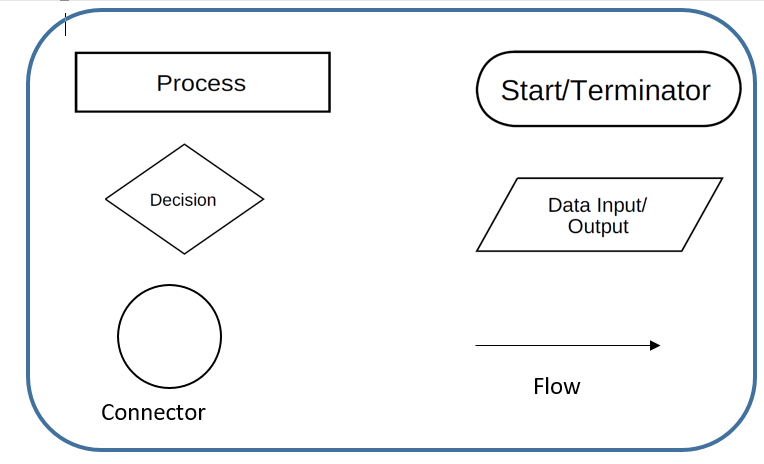
## **Flow Chart**

### Flow chart is a graphical representation of a computer program in relation to its sequence of functions (as distinct from the data it processes).

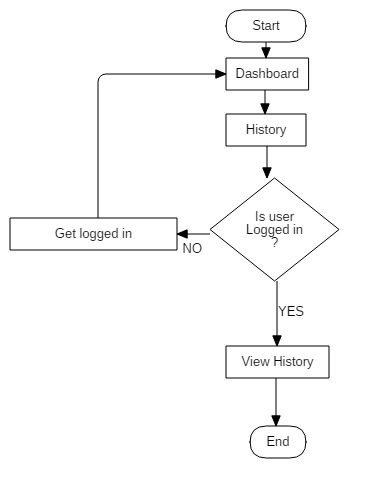
### **Justification**

It shows how a process is performed from start to finish, typically in sequential order Troubleshooting of the logical problems are easy while designing the ER model. It helps to identify how long it takes to complete each task in a specific process. It is difficult to fix bugs and problems in workflow, if you do not know what went wrong and at which stage in the process it went wrong.

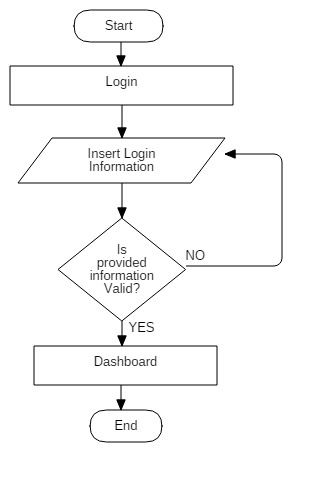
### **Notation used**



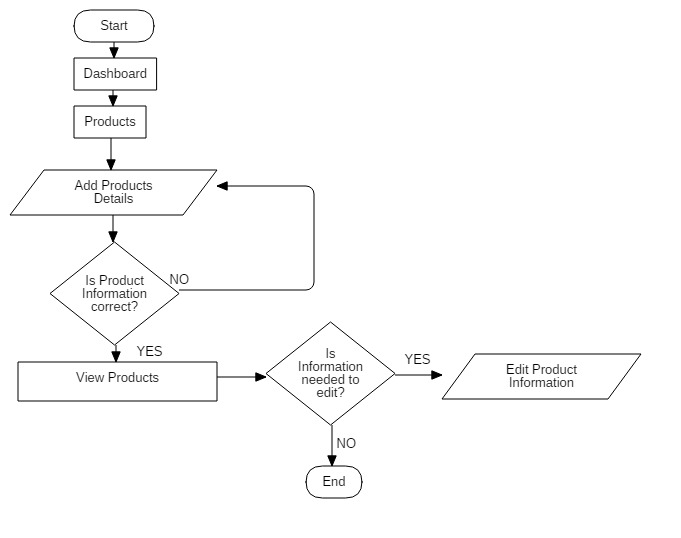
### **Actual Flow Chart**



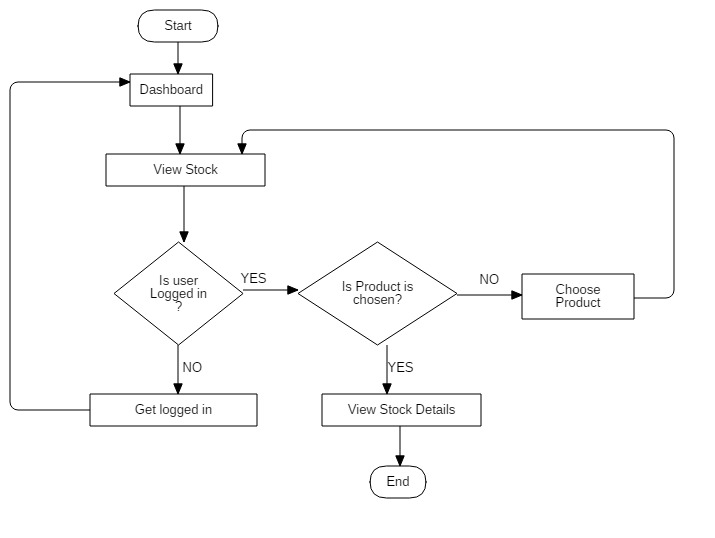
Here the flow of activity to view history is shown in aside flowchart. To view history user need to login.



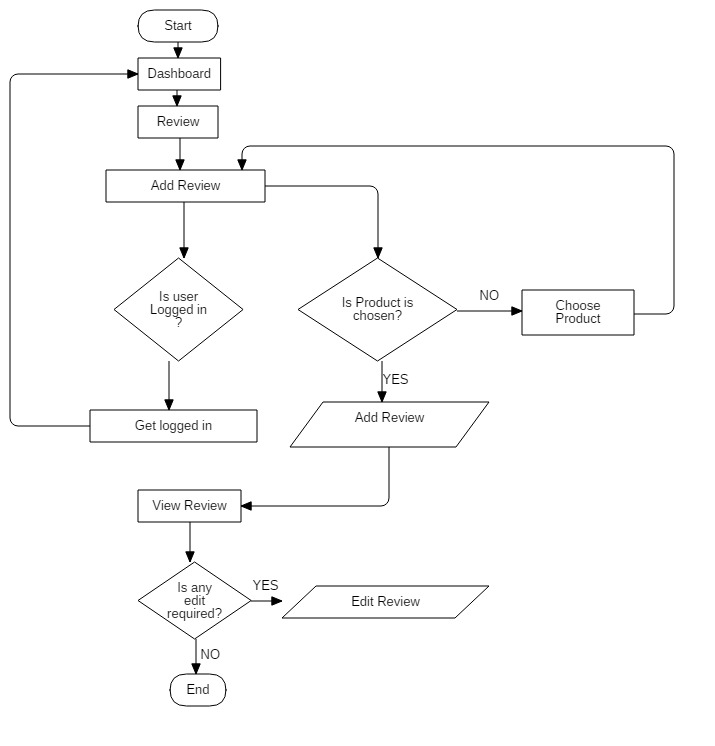
Here the flow of activity to login is shown in aside flowchart. All the valid information is to be entered to login.



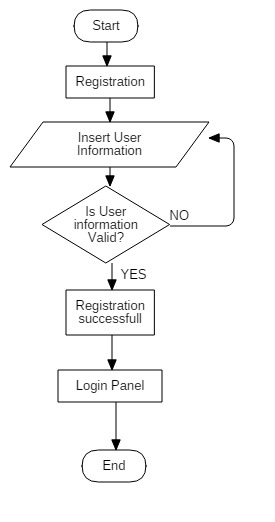
Here the flow of activity to add product details is shown in aside flowchart. Any changes that are to be made can be done.



Here the flow of activity to view stock is shown in aside flowchart. User must done login to view stock. Items are to be selected to view stock information



Here the flow of activity of review is shown in aside flowchart. User must be logged in to add and edit the reviews of products they wished. Just to view reviews login is not compulsory.



Here the flow of activity to register is shown in aside flowchart. All the valid information is to be entered to register.

## **Behavioral Diagram**

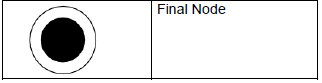
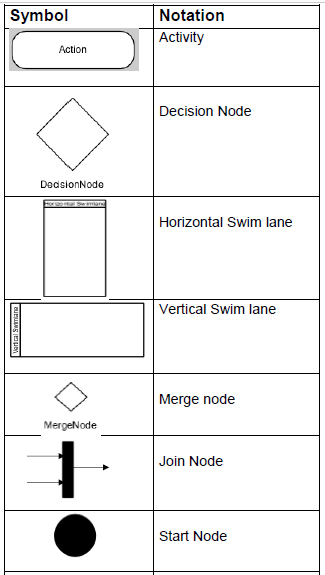
### **2.1 Activity Diagram**

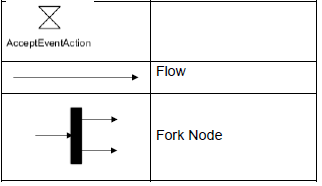
Graphical representation that shows the flow of the activity of the system is called activity diagram. It can describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

## **Justification**

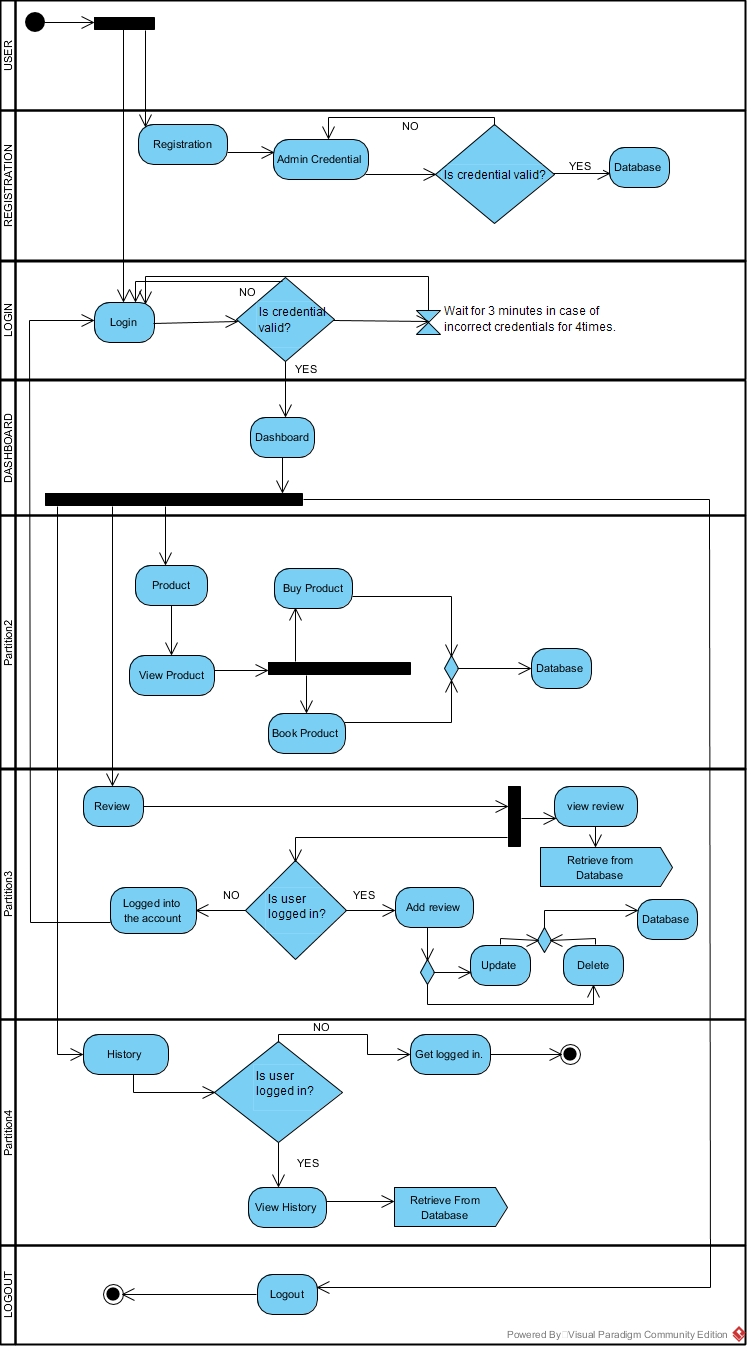
Activity diagram demonstrates the logic of an algorithm and describe the steps performed in a UML use case. It illustrate a business process or workflow between users and the system. Simplify and improve any process by clarifying complicated use cases.

## **Notation**





**Actual Activity Diagram**



**Above** activity is the activity of the user activity the system. Here initially user login in to the system. If user is new to the system user, need to be register first. While login in the system, user need to provide valid username and password then only it get access to the system. If the user attempt for four times with incorrect information then it is block for 3 minutes and only user can try login.

After user is direct to dashboard user can do several functionality of the system. If user wish to buy any cosmetics then user need to navigates to product and then view. Then user can either book or buy the products.

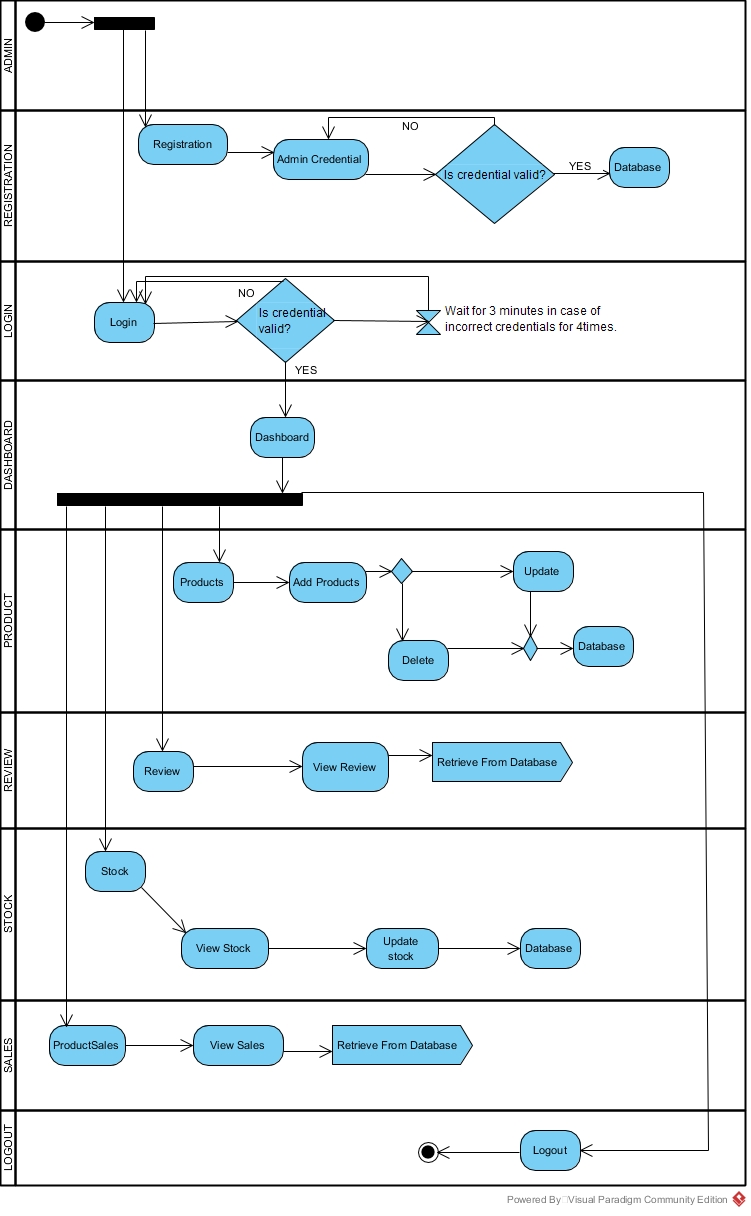
User can add and edit reviews of the product they bought. To do so user must be logged in into the system. However, to view reviews it is not necessary to get login.

History of all the activity is recorded. To view history user must be signed in to their account.

User can logout of the system whenever user want.

The **below** Activity Diagram shows the activity that can performed by the admin user in the system. Activities like :

* Initially like other user admin need to login and if not registered are registered.
* Admin can add products and their details like price, brands and quantity. These data are being stored in the database.
* Admin can have privilege of viewing the reviews of the products only.
* Admin can maintain the stock of the products. He/ she can edit the quantity of the items.
* A sales log is maintain in the system which is only given access to admin to view.



### **2.2 Sequence Diagram**

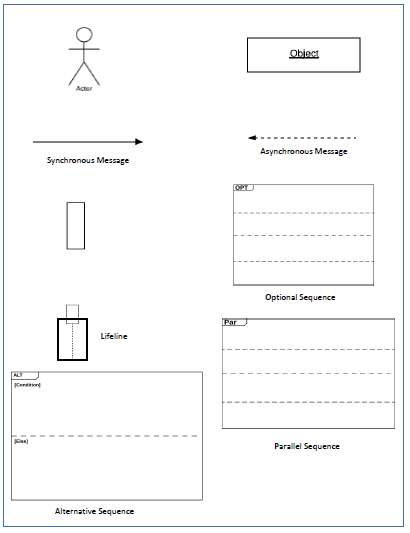
The diagrams that details how functions are carried out is clearly shown is sequence diagram. It shows the flow of message between the objects of the system.

## **Justification**

The major reasons to design sequence diagram are:

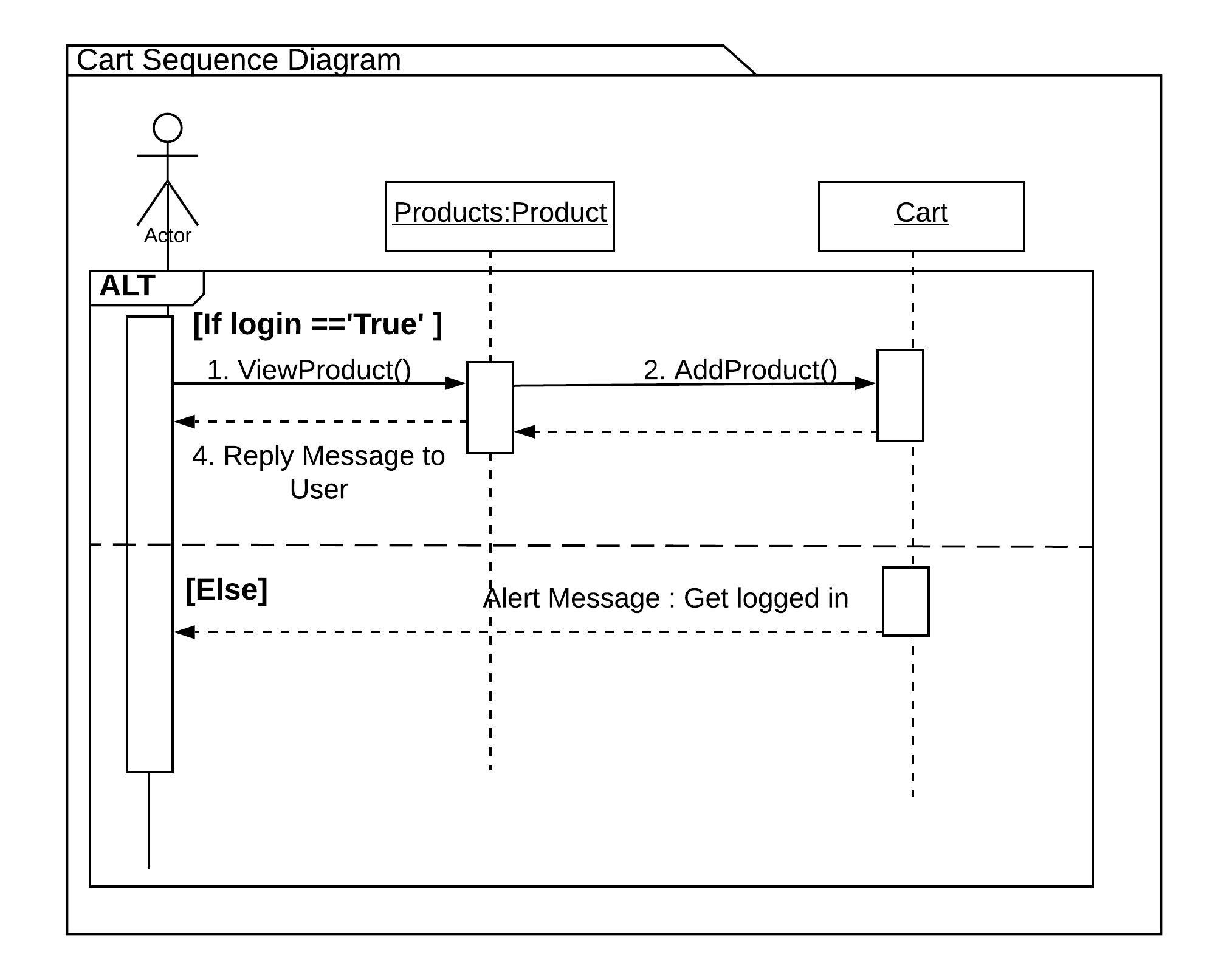
* It shows the communication between object instances within a relationship that realizes a use case
* It shows high-level interaction between active objects in a system
* It shows the message flow between the objects.

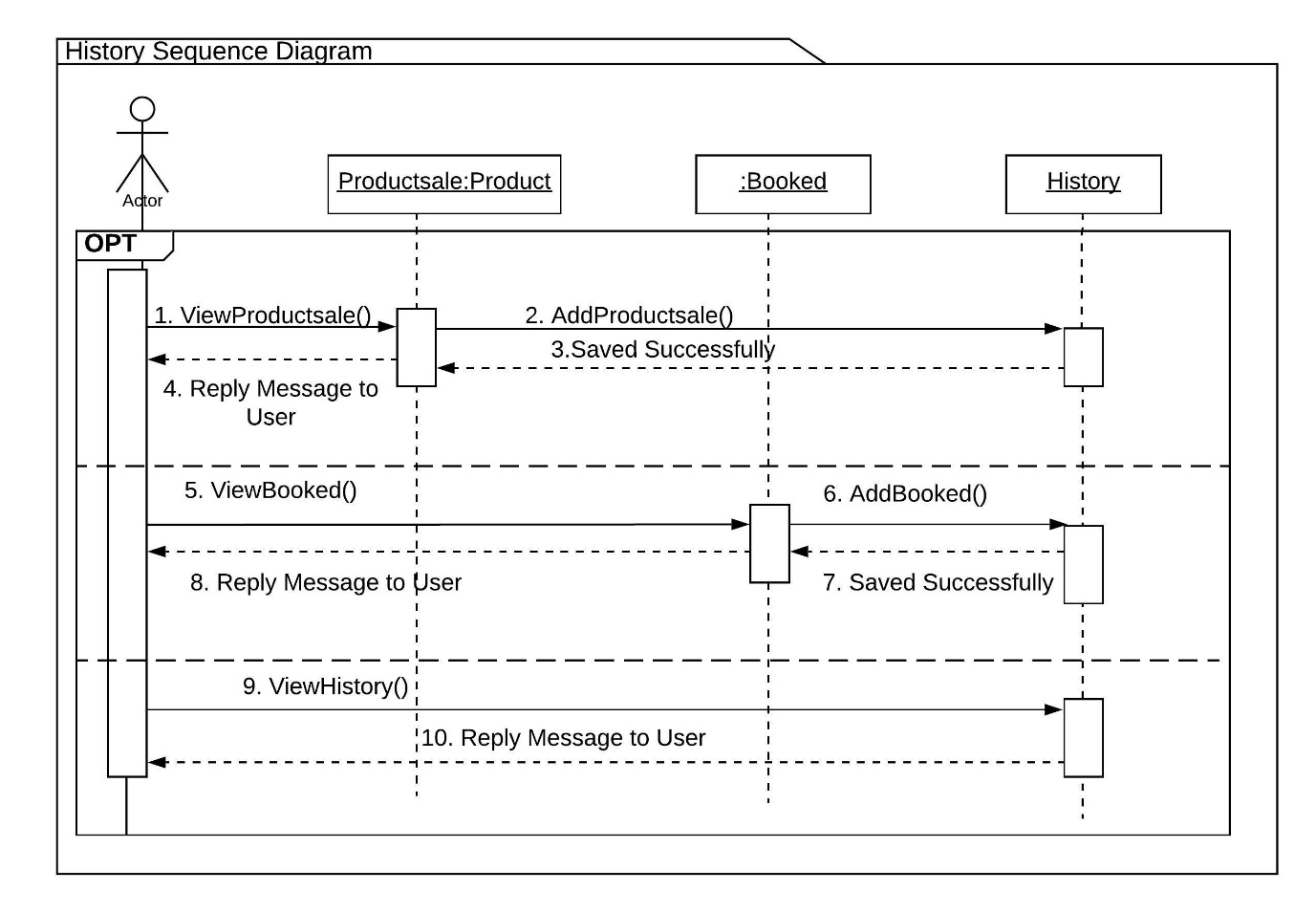
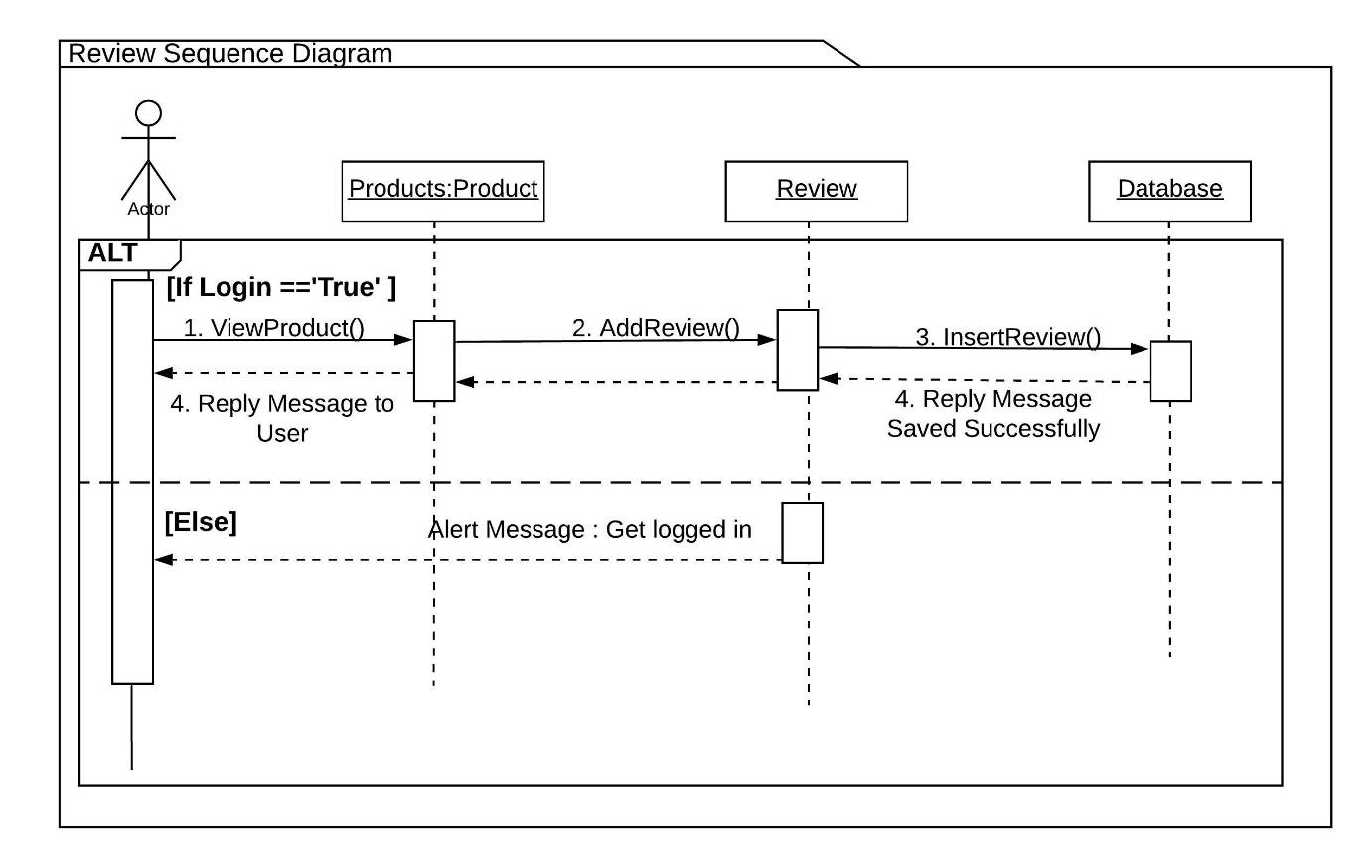
## **Notation Used**

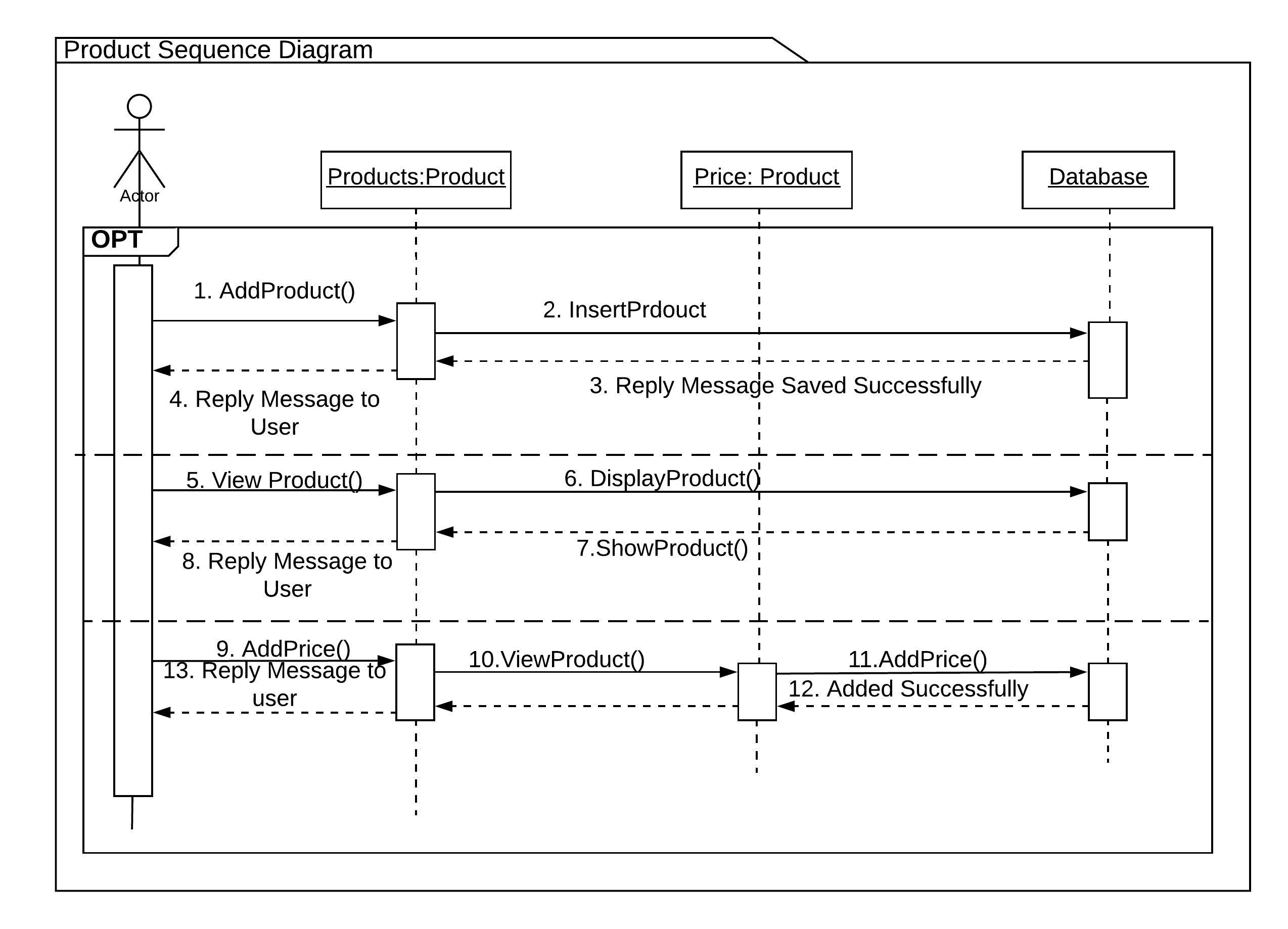


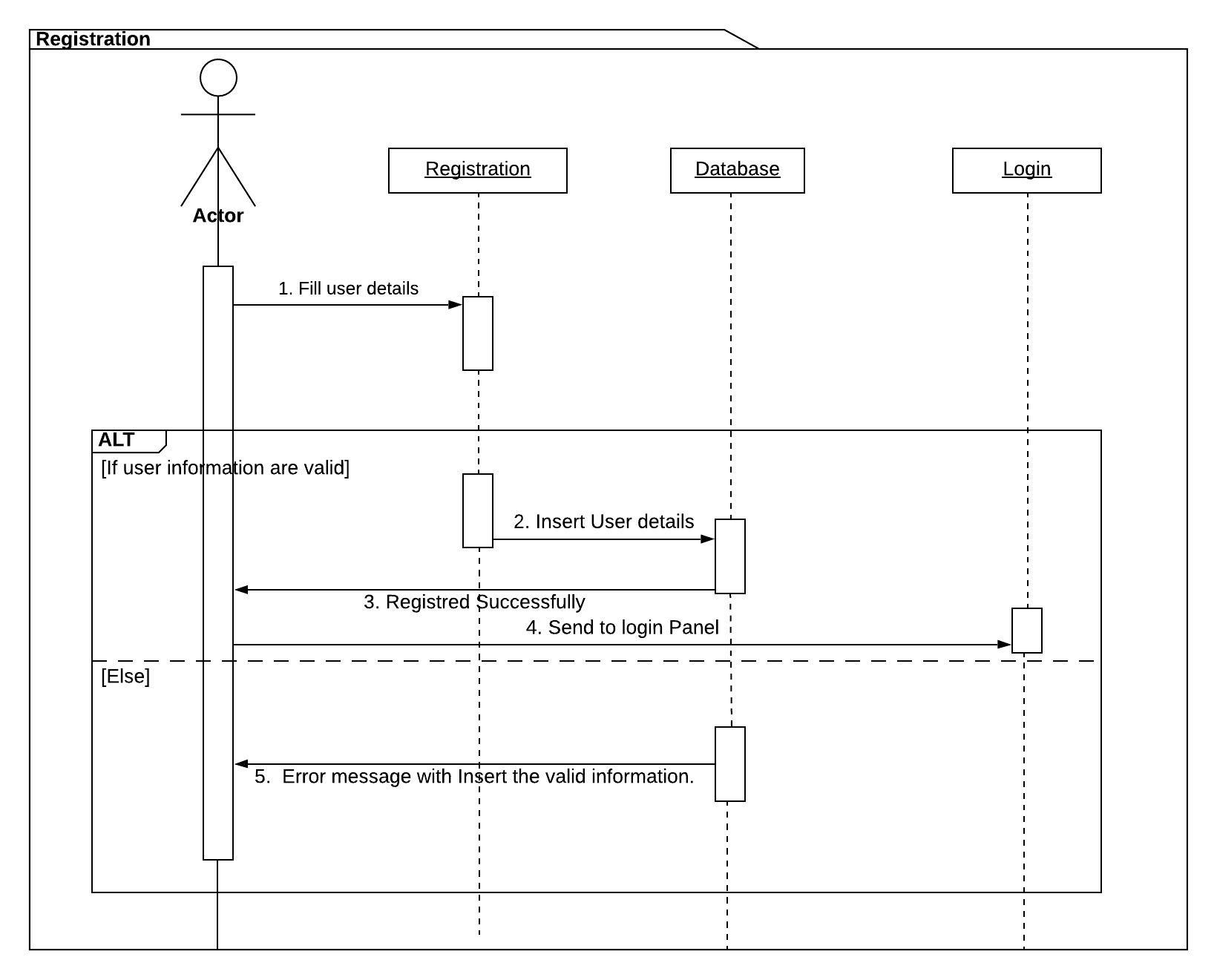
## **Actual Diagram**

* The **review sequence** diagram is in the alternate sequence. This shows the message between products and its reviews. There is condition to be applied i.e. if user is logged in then only review for the product is allowed otherwise error message will be flowed.
* The **product sequence** diagram is in the optional. User can either add new product and message from object products flows into the database. Or user can view products or allocate the price of the products. So message from respective object is passed to database.
* The **cart sequence** diagram is in the alternate sequence. This shows the message between products and its cart. There is condition to be applied i.e. if user is logged in then only products can be added into cart otherwise error message will be flowed.
* The **history sequence** diagram is in the optional. User can either add product sale and message from object products flows into the database. Or user can view booked products and add to class history or view the history. So, message from respective object is passed to database.



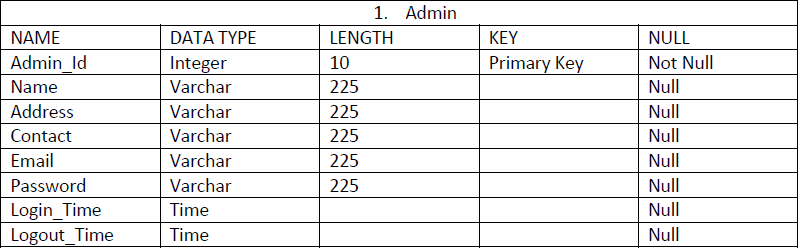


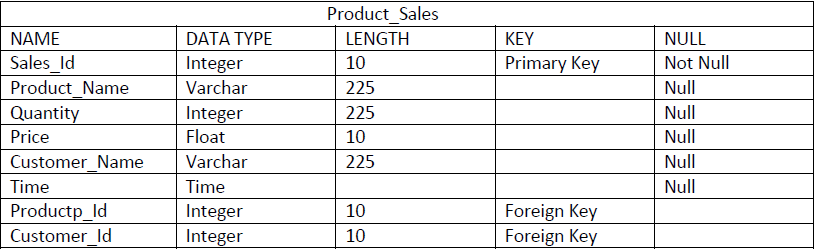


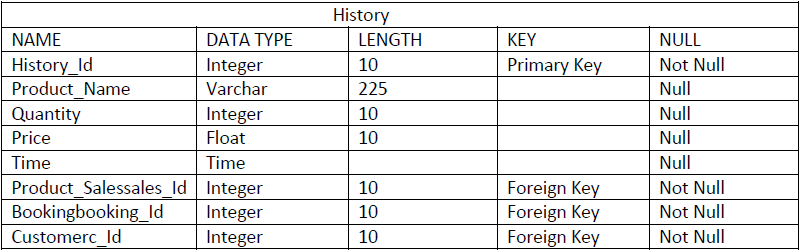


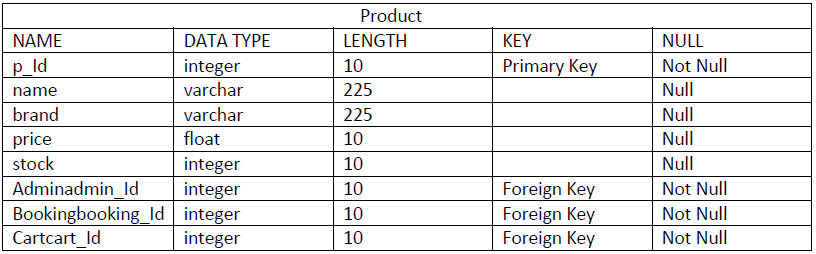
## **Database Model**

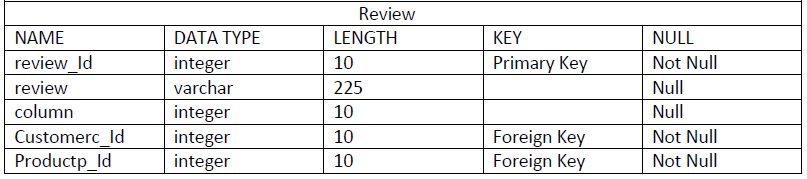
### **Data Dictionary**

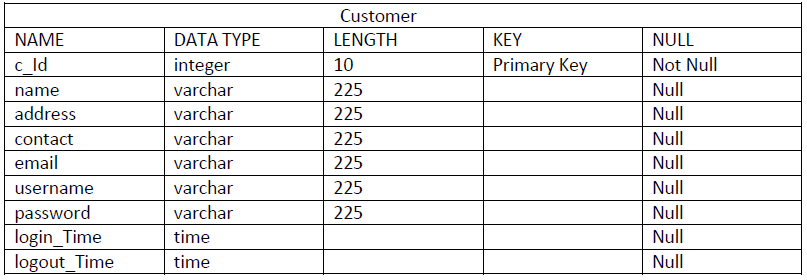




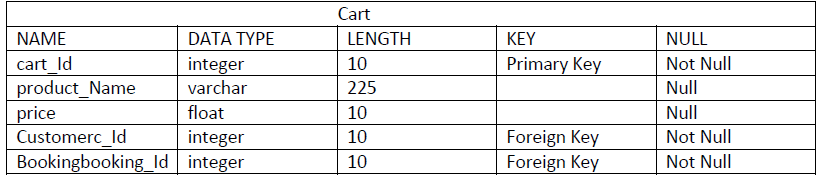


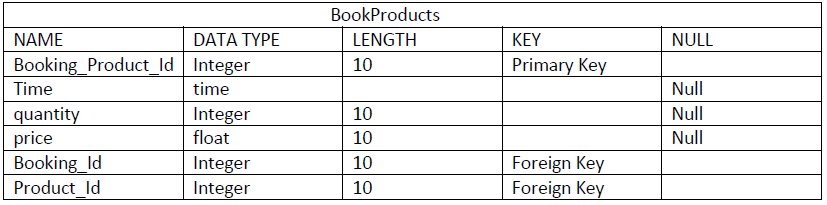












### **3.2 ER Model**

ER modelling is a graphical illustration of an information system that describes the relationships between entities. It helps to design the database of the system as all the possible anomalies are figure out and minimized as possible.

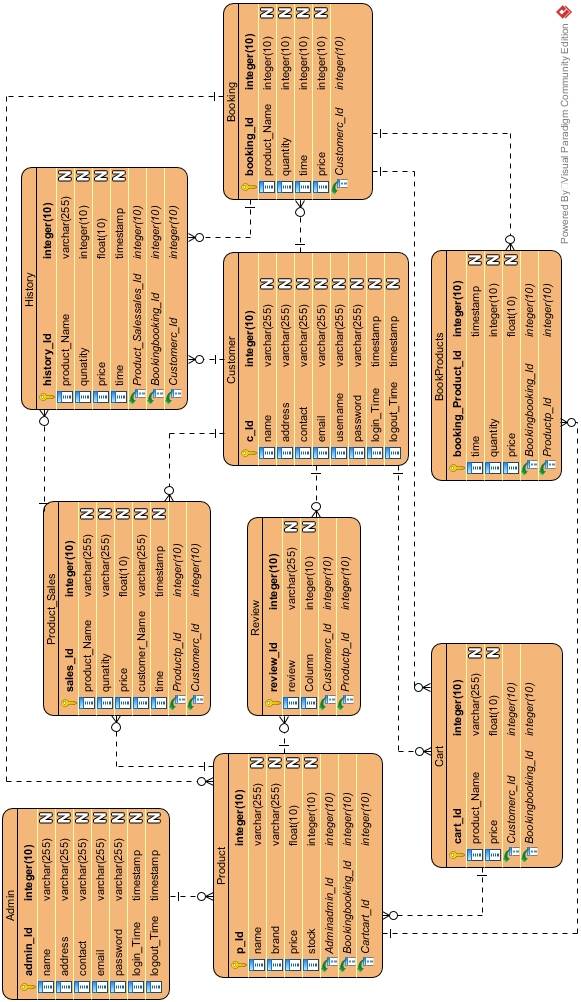
**Justification**

ER diagram help us to identify the actual requirements of the system. Is helps to resolve the possible traps of database. All the entities are shown in appropriate relationships. These relationships can help to normalize the database. Therefore, ER Diagram is designed for the system.

## **Notation Used**

Actual Diagram

In Online Cosmetic Portal admin can add multiple products so in the ER above one to many relationship is shown with admin and product. Many products will be added by the admin in the system so there is many to one relationship. Likewise one product can have multiple sales so there establish one to many relationship in product and product sales table. One customer can have multiple reviews on products. Likewise one customer can book multiple products. All the records can be recorded in the history table.



## **Architectural Model**

## **UI Modelling**

## **Prototype:**

