

**EQUIPMENT RENTING SYSTEM**

Why Buy When You Can Rent?



****

# **INDEX**

**Project Profile .........................................................................................................................5**

Need For New System ..........................................................................................................6

**Tools and Technologies...........................................................................................................7**

**Modules ....................................................................................................................................8**

**System Flow Diagram ...........................................................................................................10**

**UML Diagrams.......................................................................................................................12**

Types of UML Diagram ......................................................................................................12

Use-Case Diagram ...............................................................................................................13

Activity Diagram .................................................................................................................17

Sequence Diagram ...............................................................................................................22

Class Diagram .....................................................................................................................28

**ER Diagram ...........................................................................................................................30**

**Data Dictionaries ....................................................................................................................33**

## 2. PROJECT PROFILE

**Equipment Renting System** is the premier online platform for renting a wide range of equipment and devices. Whether you need the latest tech gadgets, outdoor gear, or specialized equipment for events, our website provides a convenient, affordable, and hassle-free solution.

We aim to promote sustainability and flexibility, allowing users to enjoy the latest technology and gear without the commitment of ownership. **"Experience More, Own Less."**

This website will provide very user-friendly access as we will provide payment through cash, credit and online UPI too.

Here is the detailed description of our project:

NEED FOR NEW SYSTEM!

EXISTING SYSTEM:

* Limited Selection

o High Costs

o Lack of online platform

* Security concerns

O No User Reviews

o Have to stick to limited

resource

PROPOSED SYSTEM:

* Improved user experience
* Wide Selection
* Payment options and security
* Customer reviews and complaint
* Convenient Delivery and Pickup
* User-Friendly Interface
* Affordable Pricing

## 3.TOOLS AND TECHNOLOGIES

|  |  |
| --- | --- |
| Tools: | * VS Code |
|  | * Chrome Browser |
|  | * Draw.io |
|  | * Postman for API testing |
|  |  |
| Frontend: | * React.js |
| Backend: | * Node.js * Express.js |
| Database: | * MongoDB atlas |

### 4. MODULES

* **Admin Module:**
* Login
* Manage Users
* Manage Vendors
* Manage Product
* Manage Profile
* Manage Bookings
* Manage Payments
* Manage Feedbacks/Complaints

* **User Module:**

* Registrations
* Login
* Manage Profile
* Search Product
* Manage Cart
* Make Payments
* Receive Confirmation
* Do feedback & Complaint
* Logout
* **Vendor Module:**

* Registrations
* Login
* Manage Profile
* Manage Product
* View Booking
* View Feedbacks/Complaint

##### SYSTEM FLOW

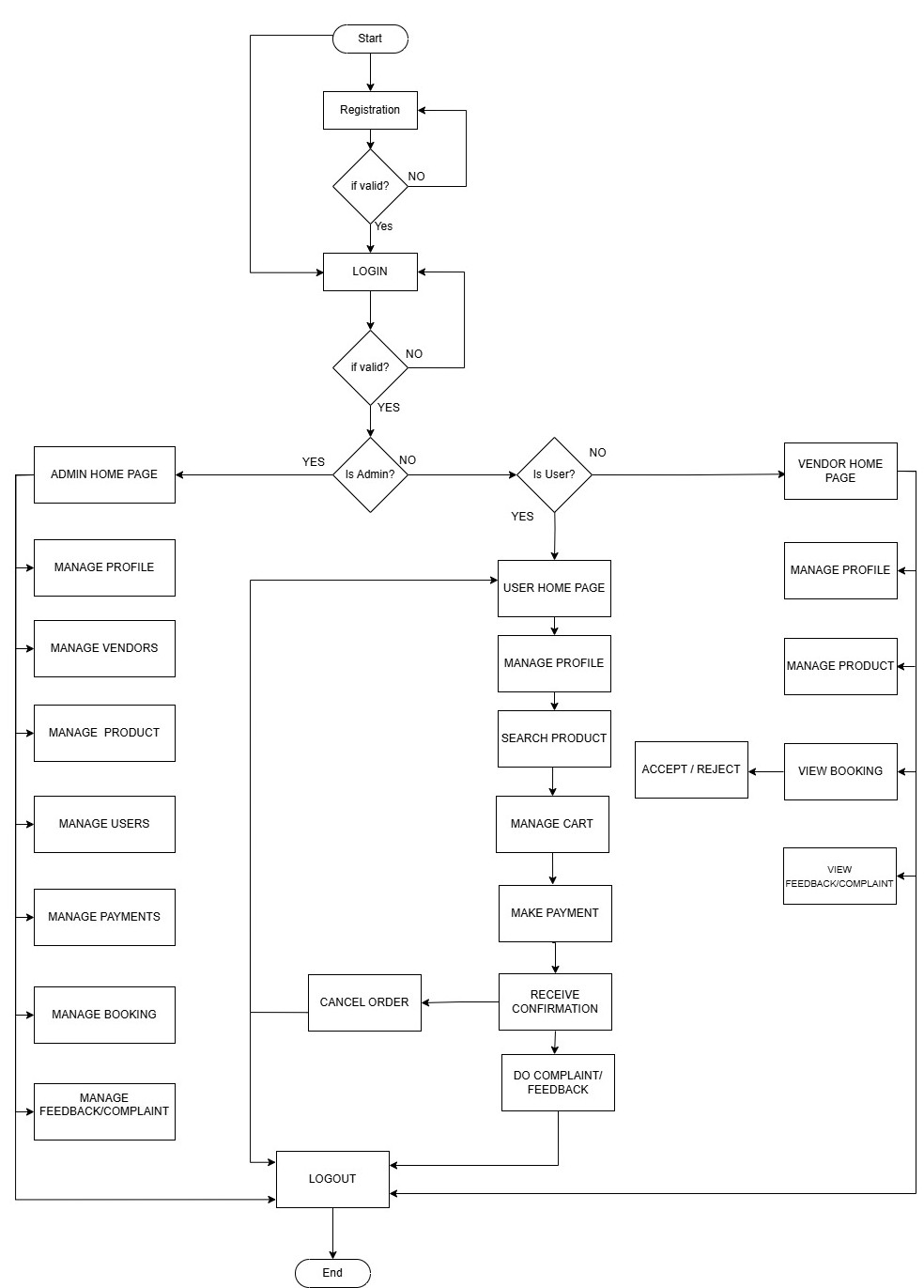
▪ System flowcharts are the diagram type that shows you the flow of data and how decisions can affect the events surrounding it.

▪ Data flowcharts let you see the overall data flow in a system.

* **System Flow Symbols and Notations:**

|  |  |  |
| --- | --- | --- |
| **Notations** | **Symbols** | **Descriptions** |
| Start/End |  | This represents start or end point of system. |
| System Flow |  | This shows the direction of the system flow. |
| Input/Output |  | This represents the input or output of system. |
| Process |  | All the process are shown in this symbol. |
| Decision |  | A diamond indicates the decision of system. |

❖Flowchart Diagram:



#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 5. UML DIAGRAMS

Unified Modelling Language (UML) diagrams are graphical representation used for system design to illustrate various aspects of system.

The overall goal of UML diagrams is to allow teams to visualize how a project is or will be working. UML can be used to develop diagrams and provide users with ready-to-use expressive modelling examples.

**Types of UML Diagrams:**

* Use Case Diagram:

An [UML](https://en.wikipedia.org/wiki/Unified_Modeling_Language) use case diagram is the primary form of system/software requirements for a new software program underdeveloped.

* Activity Diagram:

An activity diagram is used by developers to understand the flow of programs on a high level.

* Sequence Diagram:

A sequence diagrams is used to document and understand the requirements to establish new systems or learn about existing systems.

* Class Diagram:

A Class Diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

#### USE-CASE DIAGRAM

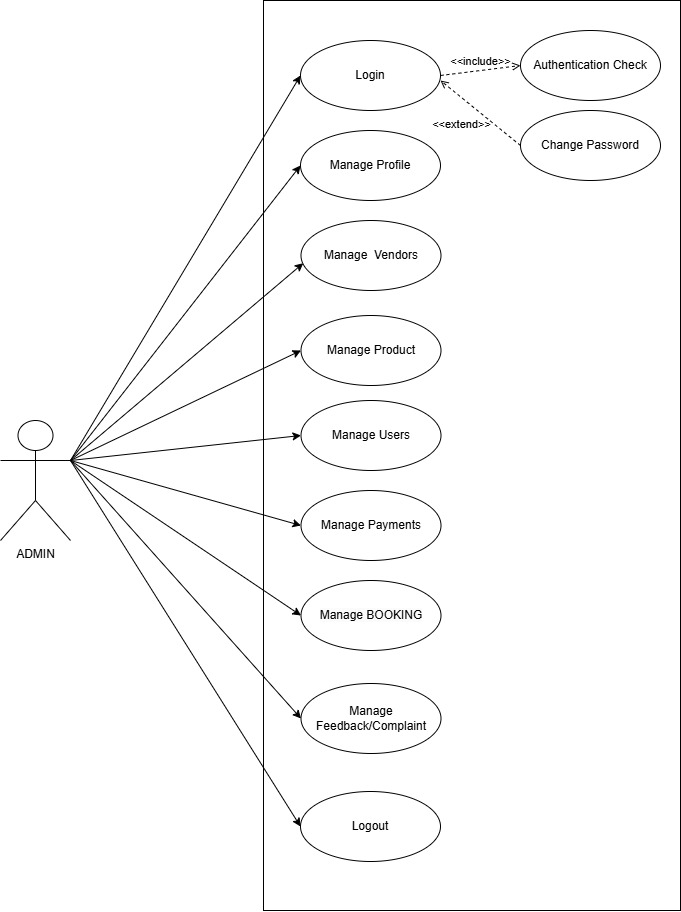
A Use-Case diagram is a way to summarize details of a system and the users within that system. It enhances the system's functionality by incorporating use cases, actors, and their relationships.

It specifies the events and relationship of entities of system.

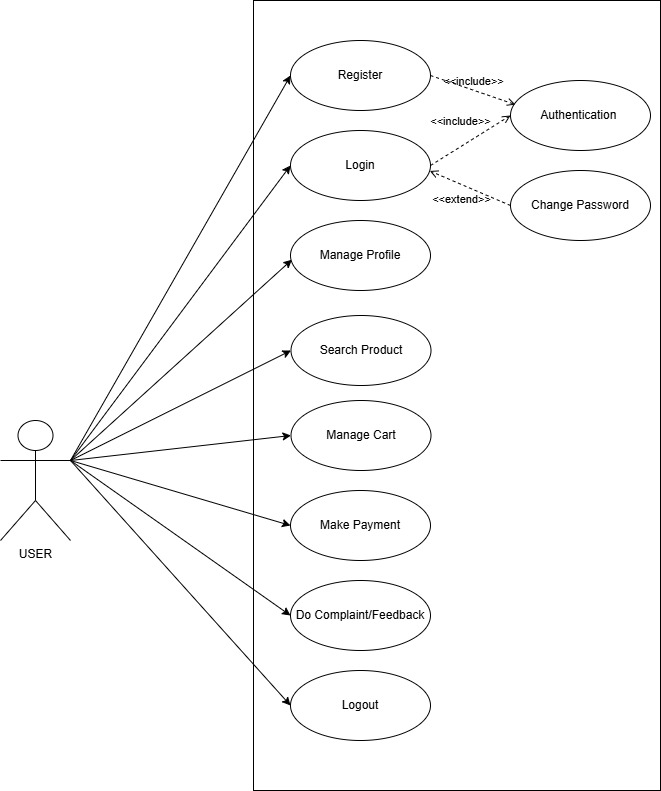
* **Use-Case Symbols and Notations:**

|  |  |  |
| --- | --- | --- |
| **Notations** | **Symbols** | **Descriptions** |
| Actor |  | Each actor performs a specific role in the system. Actors are always outside the system and interact directly with it by initiating a use case. An actor can be a person, such as a customer, or a computer, such as a database system. |
| Use-Case |  | An oval shaped use-case is used to represent the functionality of the system. It typically defines interactions between actor and system. |
| System |  | The system is used to define the scope of the use case and drawn as a rectangle |
| Communication  link |  | This link is used to show the  participation of actor in a use case. And is used to communicate |
| <<Include>> |  | It is represented by a dashed arrow in the direction of the base use case |
| <<Exclude>> |  | It is represented by a dashed arrow in the direction of the included use case |

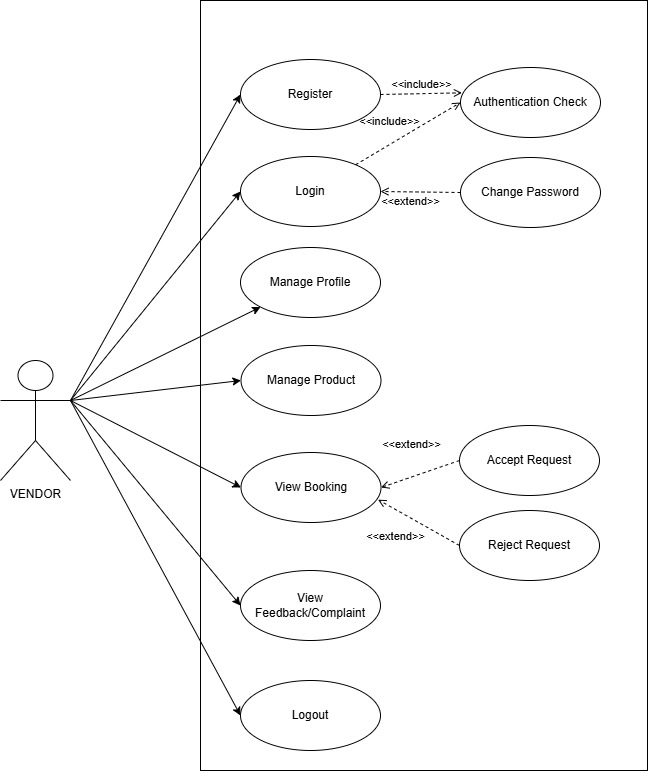
* **Use-Case for Admin Side:**



* **Use-Case for User Side:**



* **Use-Case for Vendor Side:**



#### ACTIVITY DIAGRAM

* **What is Activity Diagram?**

Activity Diagram is one of the important diagrams of UML. It represents the flow from one activity to another activity which can be shown as operations of system.

An activity diagram portrays the control flow from a start point to a finish point showing

the various decision paths that exist while the activity is being executed.

It represents what activities can be done in parallel and whether there are alternative paths through the workflow.

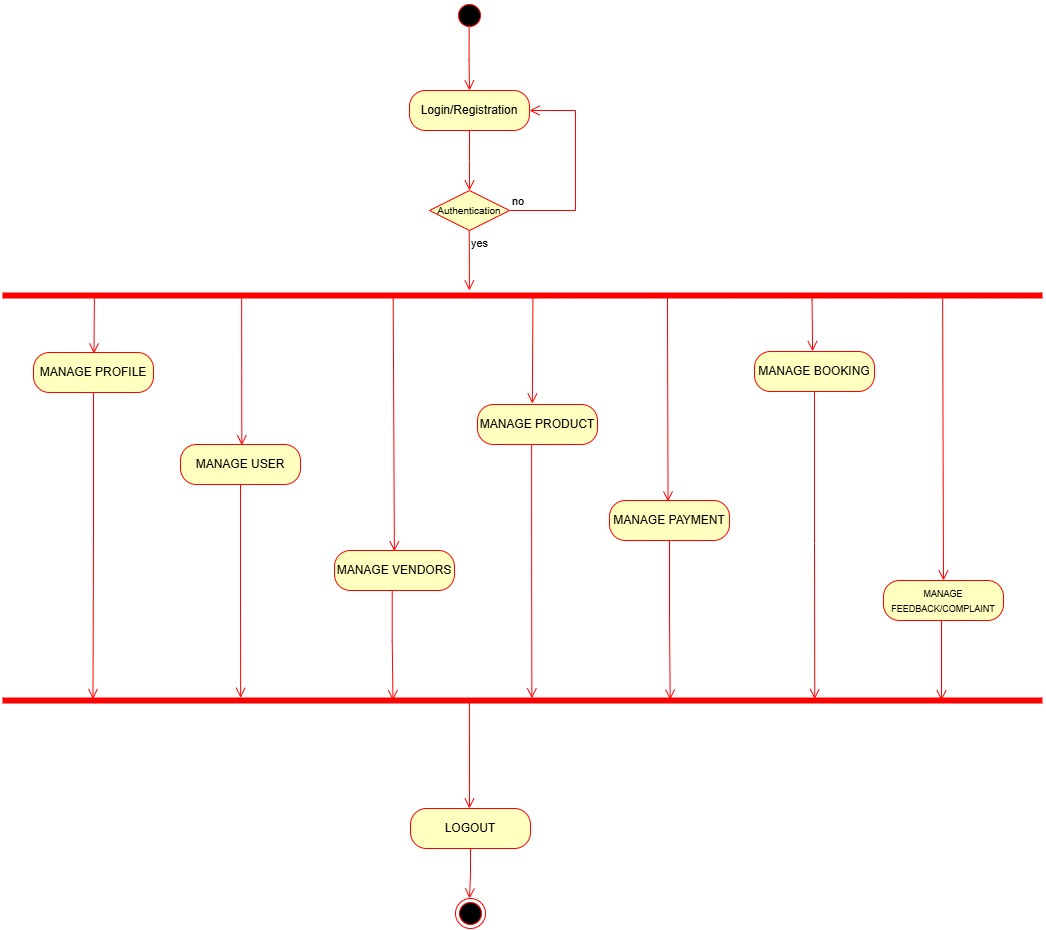
* **Benefits of Activity Diagram:**

* + Shows the progress of workflow amongst the users, and the system.
  + It provides the visual representation of logic of a particular algorithm.
  + It represents the relationship between one activity to another activity.
  + It provides a better understanding of how the system works.
  + We can use Activity diagrams to depict concurrent activities easily.

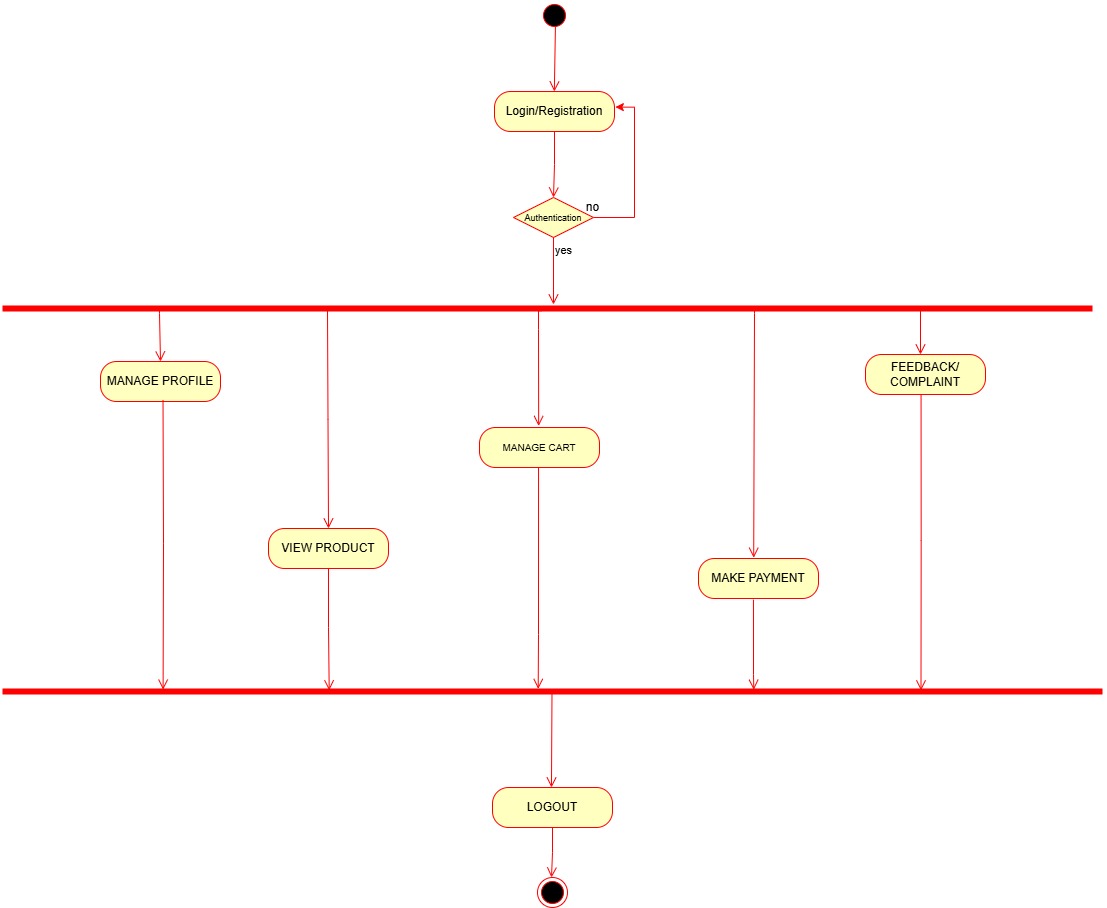
* **Activity Diagram Symbols and Notations:**

|  |  |  |
| --- | --- | --- |
| **Notations** | **Symbols** | **Descriptions** |
| Start/Initial point |  | Represents the beginning of the process |
| End point |  | Represents the completion of all the flow and end of process |
| Activity |  | Indicates the main activity with small description within the shape |
| Activity Flow |  | Shows the directional flow of the activity |
| Decision Symbol |  | Represents decision making with one input and multiple output |
| Fork |  | Indicates flow that split single flow into multiple concurrent flow |
| Join/Merge |  | Indicates flow that combine two or more concurrent flow into single flow |

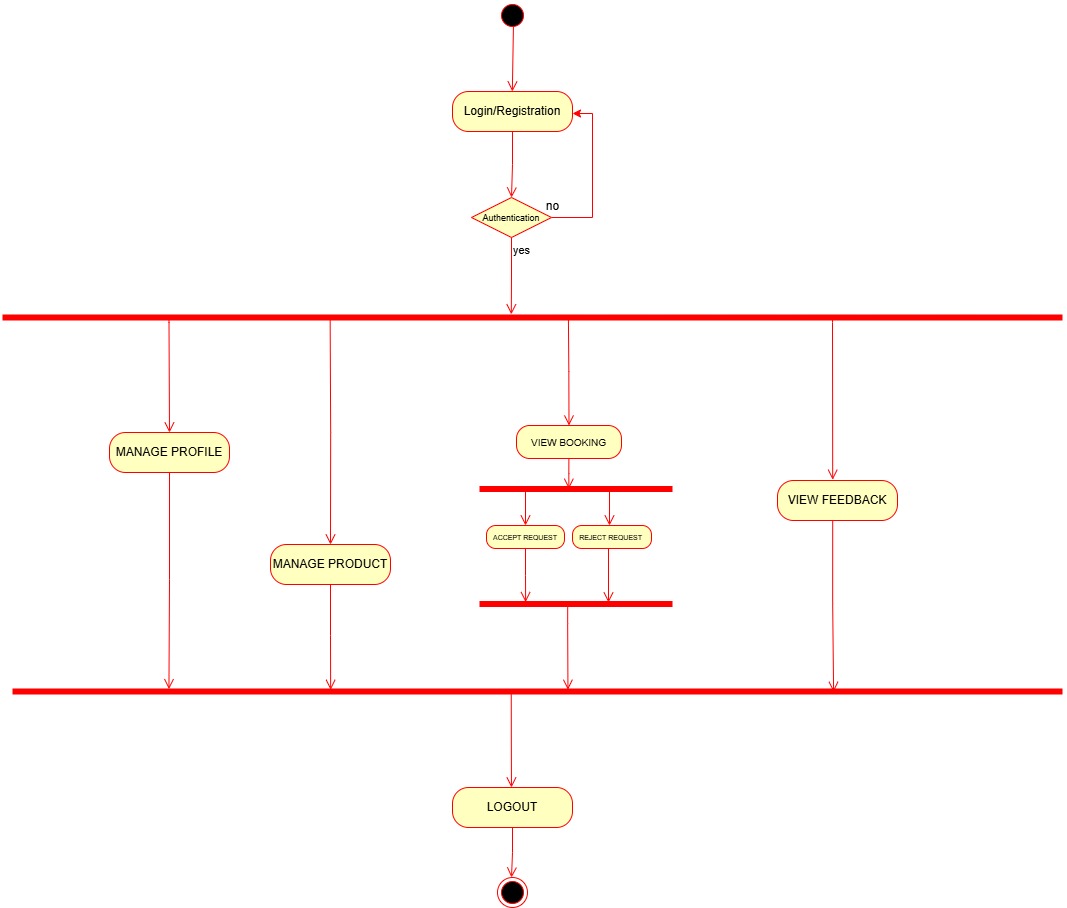
* **Activity Diagram for Admin Side:**



* **Activity Diagram for User Side:**



* **Activity Diagram for Vendor Side:**



#### SEQUENCE DIAGRAM

* **What is Sequence Diagram?**

Sequence diagram is an interaction diagram which focuses on message interchange between number of timelines.

It represents the flow of messages in the system and can also be termed as Event Diagram.

It illustrates how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

* **Benefits of Sequence Diagram:**

Following are some advantages of sequence Diagram:

* Sequence Diagram is used for high-level interactions hence helps in understanding developers design and complex system.
* It has easy maintenance.
* It depicts the message flow between the different objects.
* It can easily update as per the new change in the system.

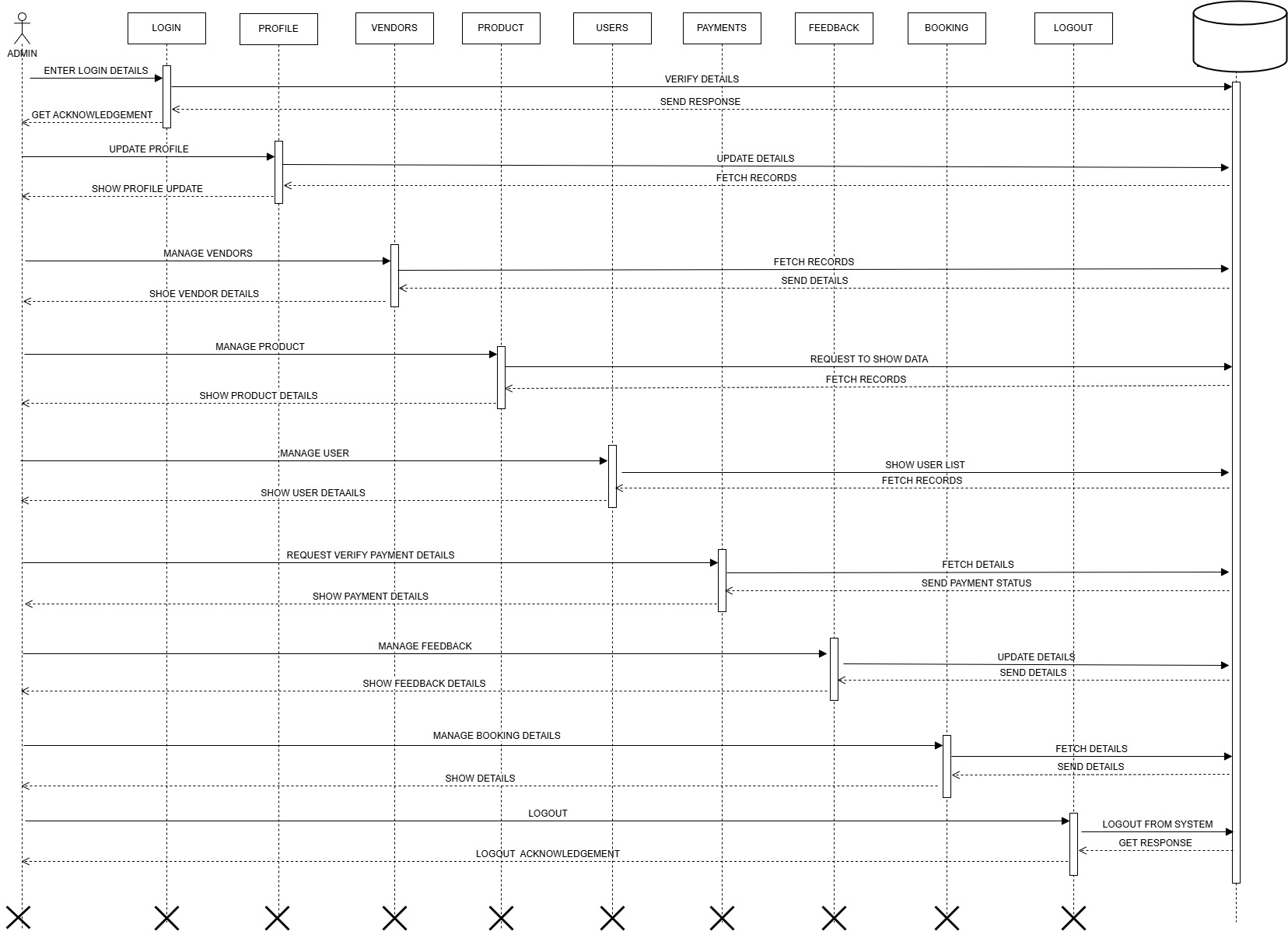
* **Sequence Diagram Symbols and Notations:**

|  |  |  |
| --- | --- | --- |
| **Notations** | **Symbols** | **Descriptions** |
| Actor |  | It represents the role, which involves human users and external hardware or subjects |
| Lifeline |  | It represents individual participants and interactions |
| Activations |  | It describes the time in which operations are performed by elements. |
| Object class |  | Represents a class or object in UML. |

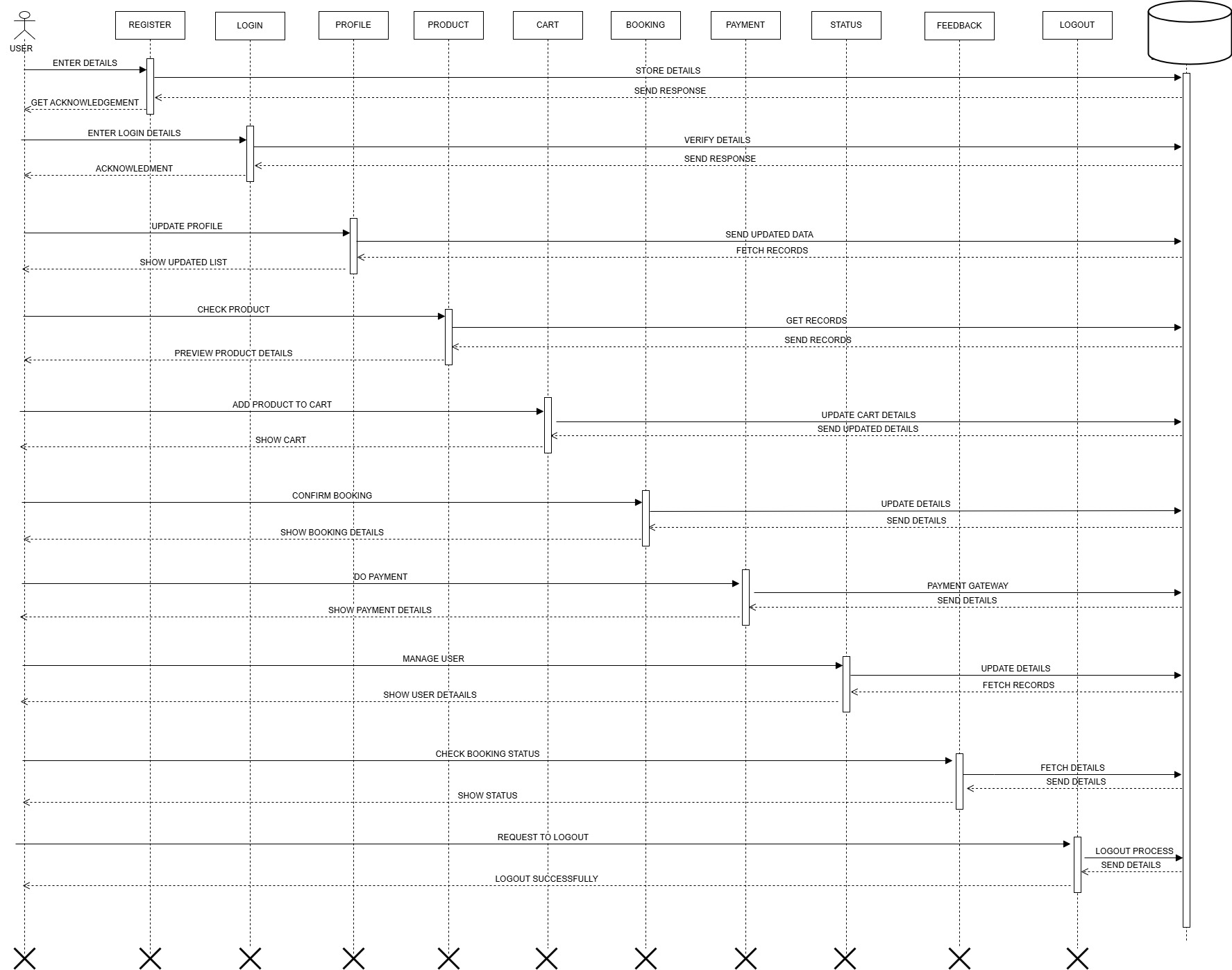
* **Sequence Diagram Symbols and Notations for messages:**

|  |  |  |
| --- | --- | --- |
| **Notations** | **Symbols** | **Descriptions** |
| Call Message |  | This show both the call and the reply of the message. |
| Return Message |  | This shows the replies to the messages. |
| Self Message |  | This shows the interactions between the same lifelines. |
| Create Message |  | This shows  communication between Lifelines of an Interaction. |
| Destroy Message |  | Destroy message is a kind of message that represents the request of destroying the lifecycle of target lifeline. |

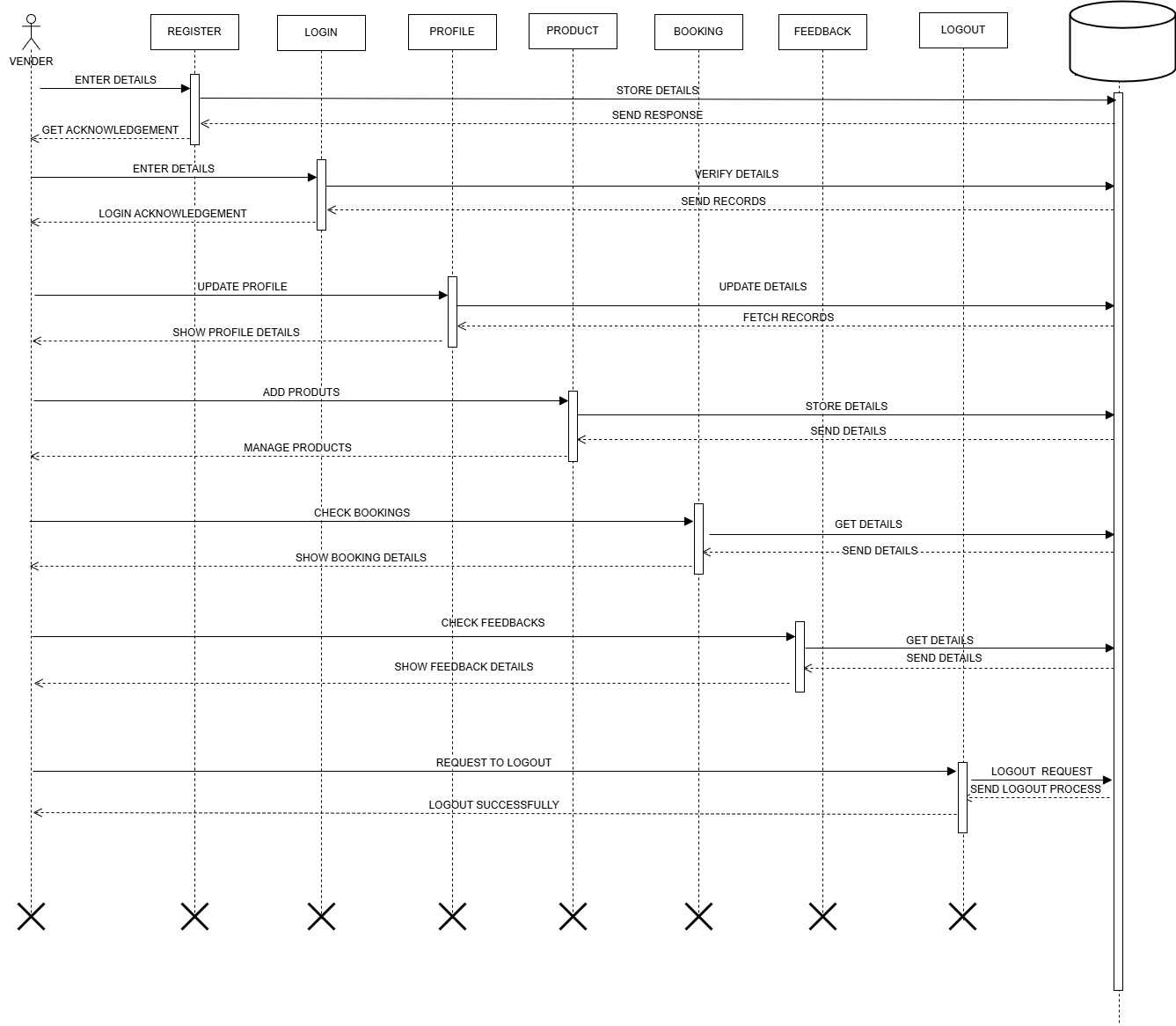
* **Sequence Diagram for admin:**



* **Sequence Diagram for user:**



* **Sequence Diagram for Vendor:**



#### CLASS DIAGRAM

• What is Class Diagram?

Class Diagram shows the static view of application that gives overview of software system by displaying classes, attributes, operations, and relationship between each other.

It is the most popular UML diagram which is widely used for constructions and can be mapped with other languages.

It helps to construct the code for the software application development.

Class diagram is collection of things or concepts that have the same characteristics.

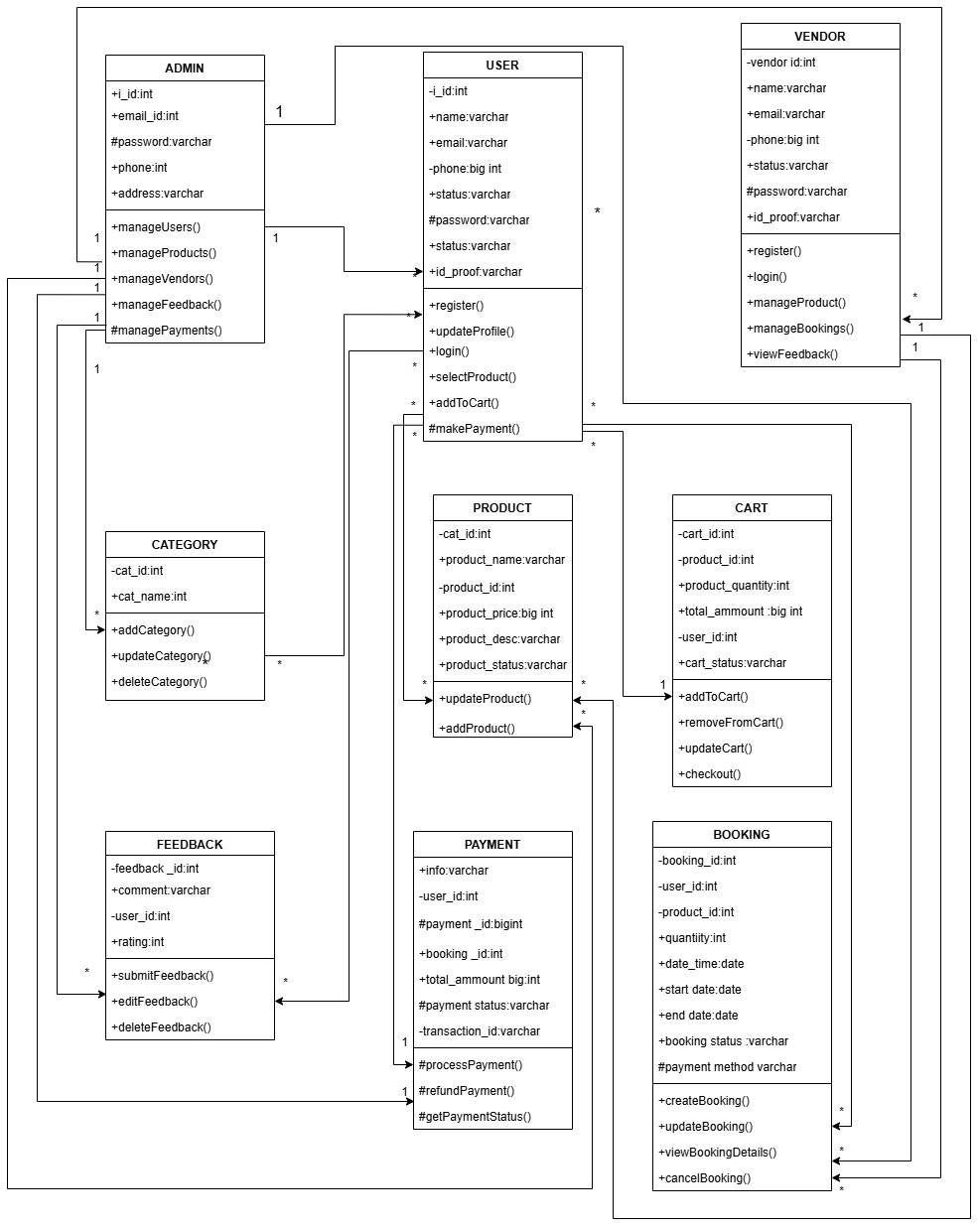
* **Components of Class Diagram:**

Class Diagram is made up of three sections which are:

* **Upper Section**: This section consists of the name of the class.

* **Middle Section**: This section consists of attributes, which describe the quality of the class.
* **Lower Section**: The lower section contains methods or operations. The methods are represented in the form of a list.

The relationship between classes is described on some cardinality which are as follows:

* One-To-One o One-To-Many o Many-To-One
* 

##### 7. ER DIAGRAM

* **What is ER Diagram?**

ER - Entity Relationship diagram is a flowchart which shows how entities such as people, objects or concepts relate to each other within a system.

ER Diagram is a diagram which focus on the relationships of elements within entities instead of relationships between entities themselves.

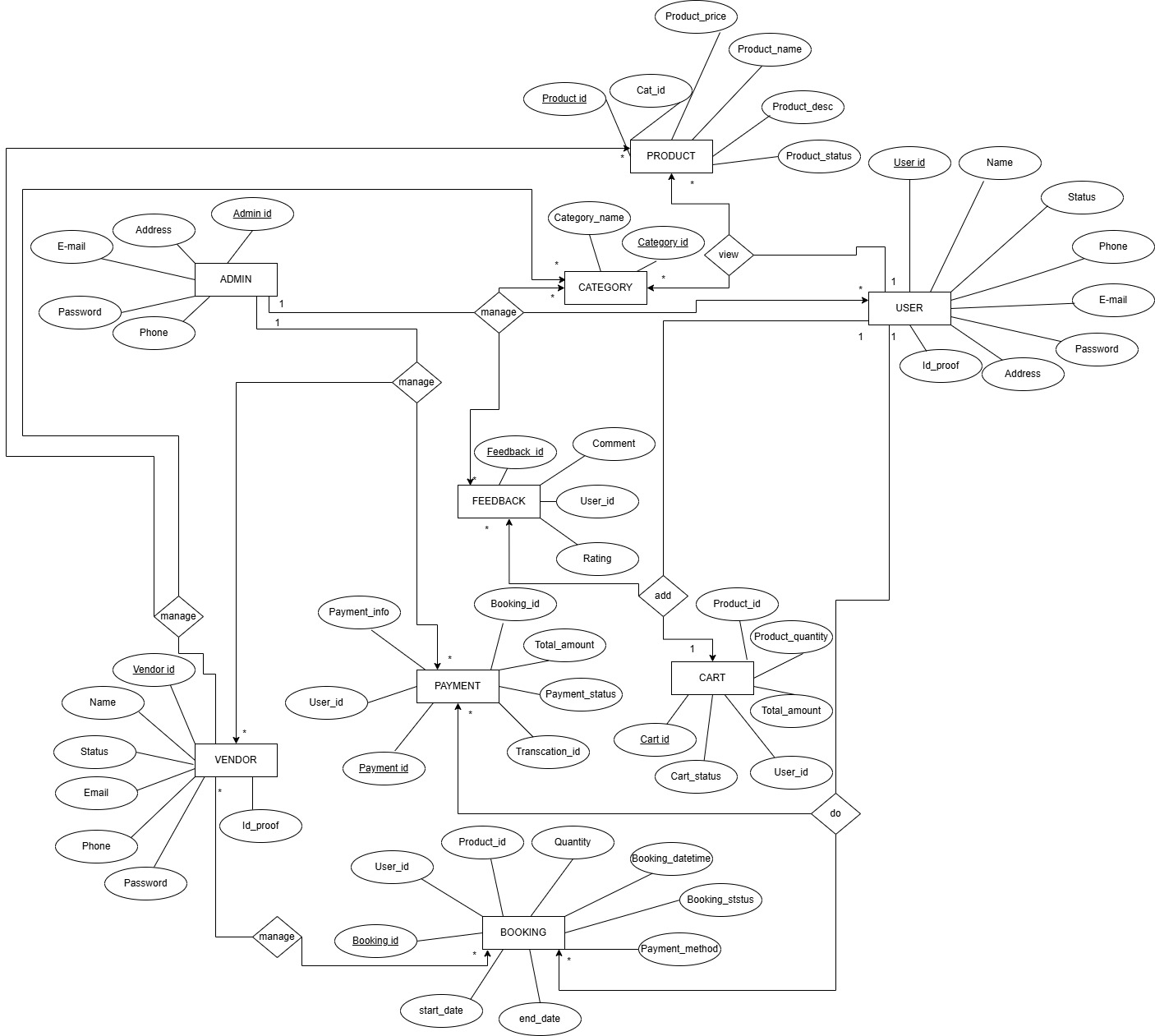
Here are some benefits of using ER Diagram:

* + One of the main benefits of ER is it gives a clear and graphical depiction of relationships between elements. This makes it easy to understand.
  + ER models are less challenging to maintain, which lowers costs and increases production.
  + ER models are highly flexible which makes it easy to make any changes.

o ER diagrams can play a key role in setting up useful databases to analyse the data.

* **ER Symbols and Notations:**

|  |  |  |
| --- | --- | --- |
| **Notations** | **Symbols** | **Descriptions** |
| Entity |  | This shows the entity of entire system |
| Attribute |  | Shows the property or characteristic of an entity. |
| Relationship |  | This represents the relation between entities. |
| Relationship Link |  | This indicates the link of relationship between entities. |
| Multivalued Attribute |  | This represents multiple attributes of an entity. |
| Primary Key |  | This shows the value which holds primary key |



## 8.DATA DICTONARIES

1. **ADMIN\_TABLE:**

Table name: Admin\_table Primary\_key: ADMIN\_ID Foreign\_key: N/A

| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| --- | --- | --- | --- | --- |
| **ADMIN\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **ADMIN\_E-MAIL** | NOT\_NULL | VARCHAR | 25 | E-MAIL OF ADMIN |
| **ADMIN\_PHONE** | NOT\_NULL | BIG\_INT | 10 | PHONE NUMBER OF ADMIN |
| **ADDRESS** | NOT\_NULL | VARCHAR | 50 | ADDRESS OF ADMIN |
| **PASSWORD** | NOT\_NULL | VARCHAR | 15 | PASSWORD OF ADMIN |

1. **USER\_TABLE:**

Table name: User\_table Primary\_key: User\_ID

| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| --- | --- | --- | --- | --- |
| **USER\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **USER\_NAME** | NOT\_NULL | VARCHAR | 30 | NAME OF USER |
| **USER\_E-MAIL** | NOT\_NULL | VARCHAR | 25 | E-MAIL OF USER |
| **USER\_PHONE** | NOT\_NULL | BIG\_INT | 10 | PHONE NUMBER OF USER |
| **ADDRESS** | NOT\_NULL | VARCHAR | 50 | ADDRESS OF USER |
| **PASSWORD** | NOT\_NULL | VARCHAR | 15 | PASSWORD OF USER |
| **ID\_PROOF** | NOT\_NULL | VARCHAR | 50 | AADHAR CARD IMAGE OF USER |
| **USER\_STATUS** | NOT\_NULL | VARCHAR | 7 | STATUS OF USER |

1. **VENDOR\_TABLE:**

Table name: Vendor\_table

Primary\_key: Vendor\_ID

Foreign\_key: N/A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **VENDOR\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **PASSWORD** | NOT\_NULL | VARCHAR | 15 | PASSWORD OF VENDOR |
| **VENDOR\_NAME** | NOT\_NULL | VARCHAR | 30 | NAME OF VENDOR |
| **VENDOR\_PHONE** | NOT\_NULL | BIG INT | 10 | PHONE NUMBER OF VENDOR |
| **VENDOR\_E-MAIL** | NOT\_NULL | VARCHAR | 25 | EMAIL OF VENDOR |
| **ID\_PROOF** | NOT\_NULL | VARCHAR | 50 | AADHAR CARD IMAGE OF VENDOR |
| **VENDOR\_STATUS** | NOT\_NULL | VARCHAR | 7 | STATUS OF VENDOR |

1. **CATEGORY\_TABLE:**

Table name: Category\_Details

Primary\_key: Cat\_ID

Foreign\_key: N/A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **CAT\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **CAT\_NAME** | NOT\_NULL | VARCHAR | 30 | NAME OF CATEGORY |

1. **PRODUCT\_TABLE:**

Table name: Product\_table

Primary\_key: PRODUCT\_ID

Foreign\_key: CAT\_ID

1. **CART\_TABLE:**

Table name: Cart\_Table

Primary\_key: CART\_ID

Foreign\_key: PRODUCT\_ID, USER\_ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **PRODUCT\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **CAT\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH CATEGORY\_TABLE |
| **PRODUCT\_NAME** | NOT\_NULL | VARCHAR | 30 | NAME OF PRODUCT |
| **PRODUCT\_STATUS** | NOT\_NULL | VARCHAR | 15 | 0-NOT AVAILABLE, 1-AVAILABLE |
| **PRODUCT\_PRICE** | NOT\_NULL | FLOAT | 8 | PRICE OF PRODUCT |
| **PRODUCT\_DESCRIPTON** | NOT\_NULL | VARCHAR | 100 | DESCRIPTION OF PRODUCT |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **CART\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **PRODUCT\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH PRODUCT\_TABLE |
| **TOTAL\_AMOUNT** | NOT\_NULL | FLOAT | 7 | TOTAL AMOUNT OF PRODUCTS |
| **PRODUCT\_QUANTITY** | NOT\_NULL | INT | 5 | PRODUCT ITEM QUANTITY |
| **USER\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH USER\_TABLE |
| **CART\_STATUS** | NOT\_NULL | VARCHAR | 15 | 0-REMOVE PRODUCT,  1-ADD PRODUCT |

1. **BOOKING\_TABLE:**

Table name: Booking\_Table

Primary\_key: BOOKING\_ID

Foreign\_key: PRODUCT\_ID, USER\_ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **BOOKING\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **PRODUCT\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH PRODUCT\_TABLE |
| **USER\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH USER\_TABLE |
| **QUANTITY** | NOT\_NULL | INT | 15 | PRODUCT ITEM QUANTITY |
| **START\_DATE** | NOT\_NULL | DATETIME | 15 | STARTING DATE OF BOOKING |
| **END\_DATE** | NOT\_NULL | DATETIME | 15 | ENDING DATE OF BOOKING |
| **BOOKING\_DATETIME** | NOT\_NULL | DATETIME | 30 | DATE AND DURATION TIME OF BOOKING |
| **BOOKING\_STATUS** | NOT\_NULL | VARCHAR | 10 | 0-PENDING,1-CANCELLED,2-DONE |
| **PAYMENT\_METHOD** | NOT\_NULL | VARCHAR | 10 | 0-ONLINE, 1-OFFINE |

1. **PAYMENT\_TABLE:**

Table name: Payment\_Table

Primary\_key: PAYMENT\_ID

Foreign\_key: BOOKING\_ID, USER\_ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **PAYMENT\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **BOOKING\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH BOOKING\_TABLE |
| **USER\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH USER\_TABLE |
| **TOTAL\_AMOUNT** | NOT\_NULL | INT | 8 | TOTAL AMONT OF CART |
| **TRANSACTION\_ID** | NOT\_NULL | VARCHAR | 10 | TRANSACTION ID OF USER |
| **PAYMENT\_STATUS** | NOT\_NULL | VARCHAR | 7 | 0-PENDING,1-CONFIRM |
| **PAYMENT\_INFO** | NOT\_NULL | VARCHAR | 30 | INFORMATION OF PAYMENT |

1. **FEEDBACK\_TABLE:**

Table name: Feedback\_Table

Primary\_key: FEED\_ID

Foreign\_key: USER\_ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTRIBUTES** | **CONSTRAIN** | **DATATYPE** | **SIZE** | **DESCRIPTION** |
| **FEED\_ID** | PRIMARY\_KEY | INT | 5 | PRIMARY KEY FOR THIS TABLE,UNIQUE FOR EVERY ROW |
| **COMMENTS** | NOT\_NULL | VARCHAR | 30 | COMMENTS FROM USER |
| **USER\_ID** | FOREIGN\_KEY | INT | 5 | MAPPED WITH USER\_TABLE |
| **RATING** | NOT\_NULL | INT | 1 | RATING FROM USER |