**Entity Relationship Diagram**

1. Possible entities

The possible entities for the ecommerce site are:

1. Users
2. To store information about the users logged into the system of the ecommerce
3. Product
4. To store information about the basics of the products from different ecommerce sites
5. Saved\_products
6. This is used to store information about the products that the user has saved
7. Notification
8. This is used to store the information about the changes in the price that the user has saved and has to notify
9. Relationship between entities
10. Users and products

The relationship between users and products is one to many as a user can view multiple products in the website

1. Users and Saved\_products

The relationship between users and Saved\_products is one to many as a user can save multiple products that they like

1. Users and Notification

The relationship between users and notification is many to one as a user can be verified whenever there is a change in price

1. Products and Saved\_products

The relationship between products and Saved\_products is one to many as a user can save multiple products that they like

1. Saved\_products and notification

The relationship between Saved\_products and notification is one to one as there can be single notification for a single product.

1. Attributes
2. Users

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Type** | **Others** |
| User\_id | Int(10) | PRIMARY KEY |
| Name | Varchar(25) |  |
| Address | Varchar(30) |  |
| Phone number | int(10) |  |
| Email\_address | Text |  |

1. Product

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Type** | **Others** |
| Product\_id | Int(10) | PRIMARY KEY |
| Product\_name | Varchar(25) |  |
| Price | Int(10) |  |
| Discounted\_price | Int(10) |  |
| Rating | Int(5) |  |

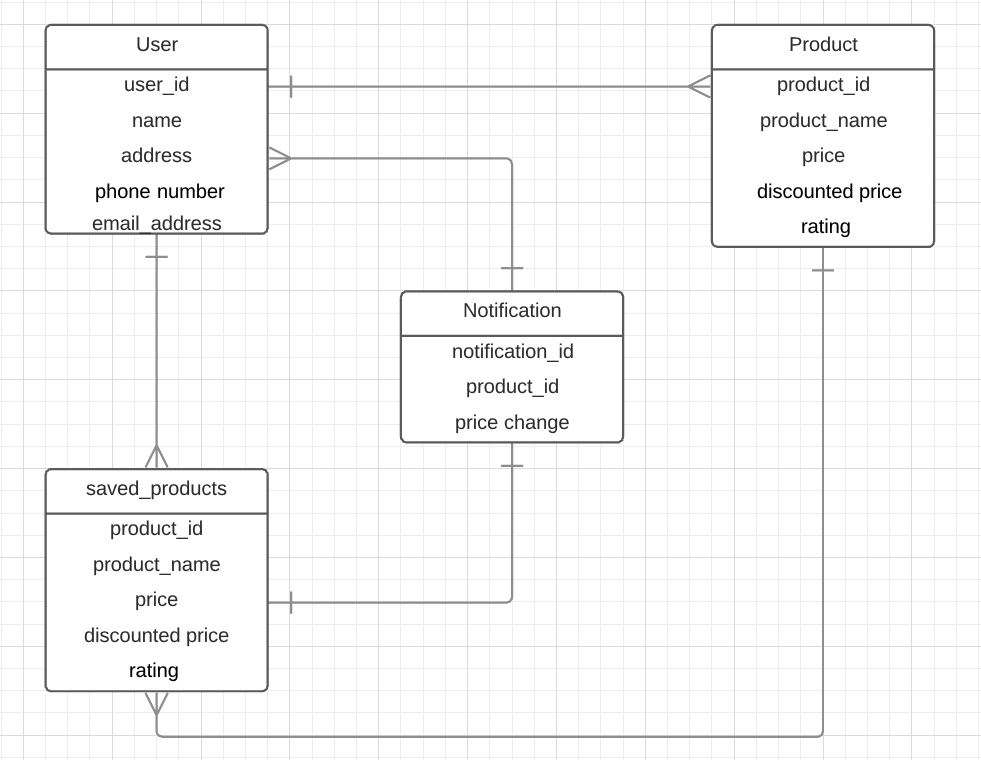
1. Saved\_products

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Type** | **Others** |
| Product\_id | Int(10) | FOREIGN KEY (products-product\_id) |
| Price | Int(10) |  |
| Discounted price | Int(10) |  |

1. Notification

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Type** | **Others** |
| Notification\_id | Int(10) | PRIMARY KEY |
| Product\_id | Int(10) | FOREIGN KEY (product-product\_id) |
| Price\_change | Int(10) |  |

1. ER Diagram



# Extended Use Case

Two expanded use cases

Use-Case: Login

Actors: Customer

Description: A customer can login into the system according to

their respective pages. The login of the user is verified and after the verification is made the process is completed to move forward to their Saved Products page.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User clicks on the login button | 1. Displays the login form |
| 1. User fills in the information |  |
| 1. User clicks the done button | 1. Verifies the user |
|  | 1. User is directed to the respective page |

Alternative:

Line 3: If the user leaves any field empty then error is displayed, and the use-case ends.

Line 5: if the user information is not valid after the user checks it, then the use-case ends.

Use-Case: Register User

Actors: Customer

Description: A customer registers itself into the system by providing all of its information. After the information is given the user is verified and then enters into the system.

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. User clicks on the Registration button | 1. Displays the Registration form |
| 1. User fills in the form with all the personal information required |  |
| 1. User clicks the done button | 1. Verifies the registration form |
|  | 1. Stores the registration |

Line 3- If the user leaves any field empty then error is displayed, and the use-case ends.

Line 5- if the user is not verified then it shows error and the use-case ends.

ACTIVITY DIAGRAM

**ACTIVITY DIAGRAM FOR LOGIN**