Indian Institute of Technology, Indore
CS 257
DBIS Project
Online Retail Portal

Submitted by- Vinit Shah (150001029)
and Rahul Choudhary (150001027)

Project Description:

The aim of the project is to create an efficient online retail portal (a.k.a online shopping system) which facilitates online transaction of goods. The shopper would be prompted to login to the website, view products, add products to their carts that they wish to buy and eventually make payment for the goods. The website would also provide an interface for the sellers where they can display their stock inventory which is available for purchase.

ER Analysis: Identifying Entity Sets and Relationship Sets

Entity-Relation for Purchases:

1. Customer (E)
   - UserID

- Username
- E-Mail
- Name
- House number
- Locality
- City
- PIN
- Date of Birth
- Phone number
- Alternative phone number
- Password

2. Product (E)
    - Name
    - ProductID
    - Stock available
    - Brand
    - Category
    - Description
    - Price

3. Purchases (R)

Entity-Relation for Payment:

1. Customer (E)
    - Attributes same as before

2. Payment (E)
    - PaymentID
    - Amount
    - Username
    - Payment mode
    - Street address
    - City
    - State
    - Postal Code

3. Payment done by Customer (R)

Entity-Relation for Selling:

1. Seller (E)
    - Name
    - Username
    - Password
    - Aadhar number
    - Pan Card Number
    - SellerID
    - Phone number
    - E-mail

2. Product (E)
    - Attributes same as before

3. Sold by (R)

Entity-Relation for Addition to Cart:

1. Cart (E)
    - Username
    - ProductID
    - Total Quantity

2. Product (E)
    - Attributes same as before

3. Added to (R)

Entity-Relation for Reviews:

1. Review (E)
    - Username
    - ProductID

- Comment
- Rating

2. Product (E)
    - Attributes same as before

3. Made for (R)

Entity-Relation for Cart Payment:

1. Cart (E)
    - Attributes same as before

2. Payment (E)
    - Attributes same as before

3. Done for (R)

Entity Relation for Product Replacement:

1. Replacement (E)
    - ReplaceID
    - Description
    - ProductID
    - UserID
    - Status
    - Replace date

2. Customer (E)
    - Attributes same as before

3. Demands Replace (R)

Entity Relation for Product Request:

1. Customer (E)
    - Attributes same as before

2. Seller (E)
    - Attributes same as before

3. Requests (R)
    - ProductID
    - Quantity

Entity Relation for Order Placement:

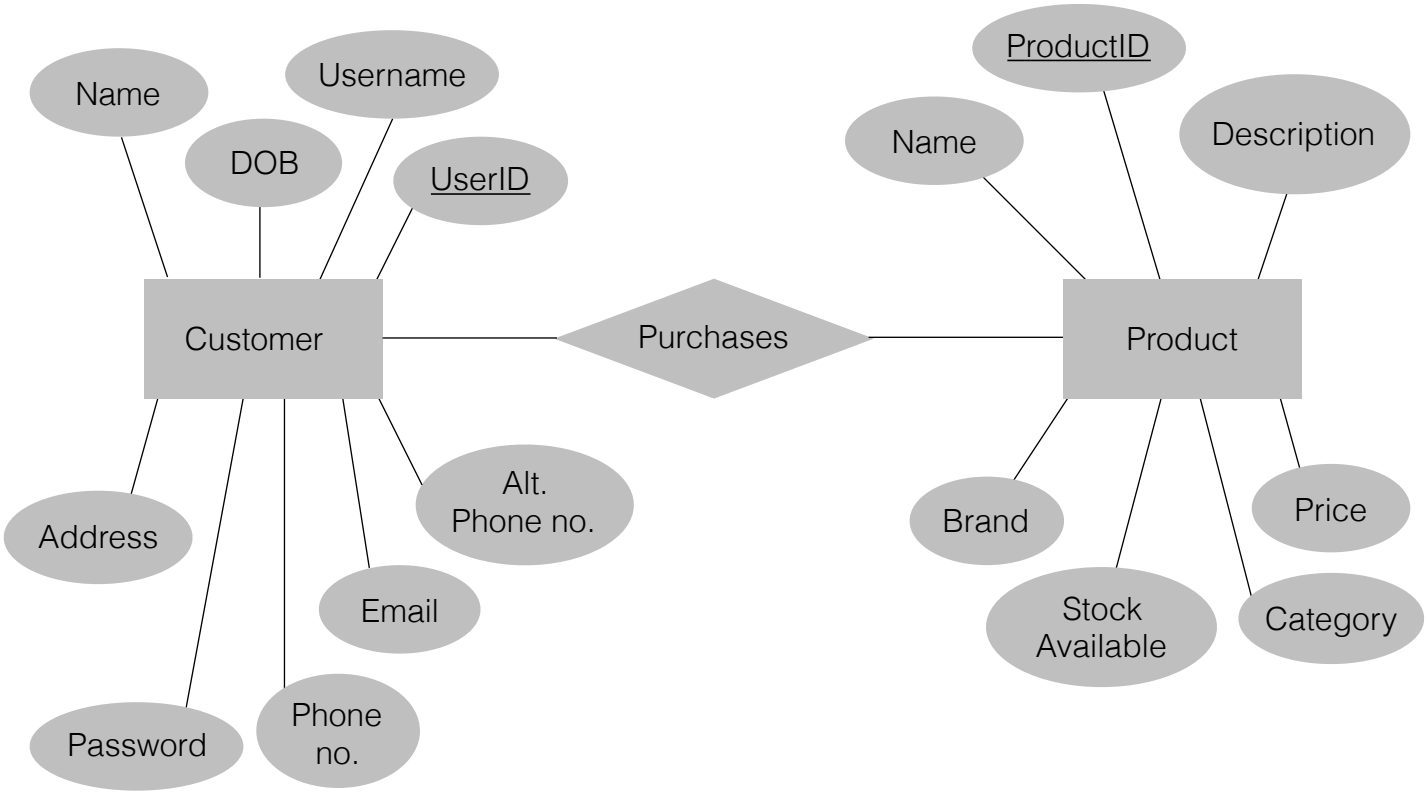
1. Orders (E)
    - Username
    - ProductID
    - Quantity
    - Order Date
    - Delivery Status

2. Payment (E)
    - Attributes same as before

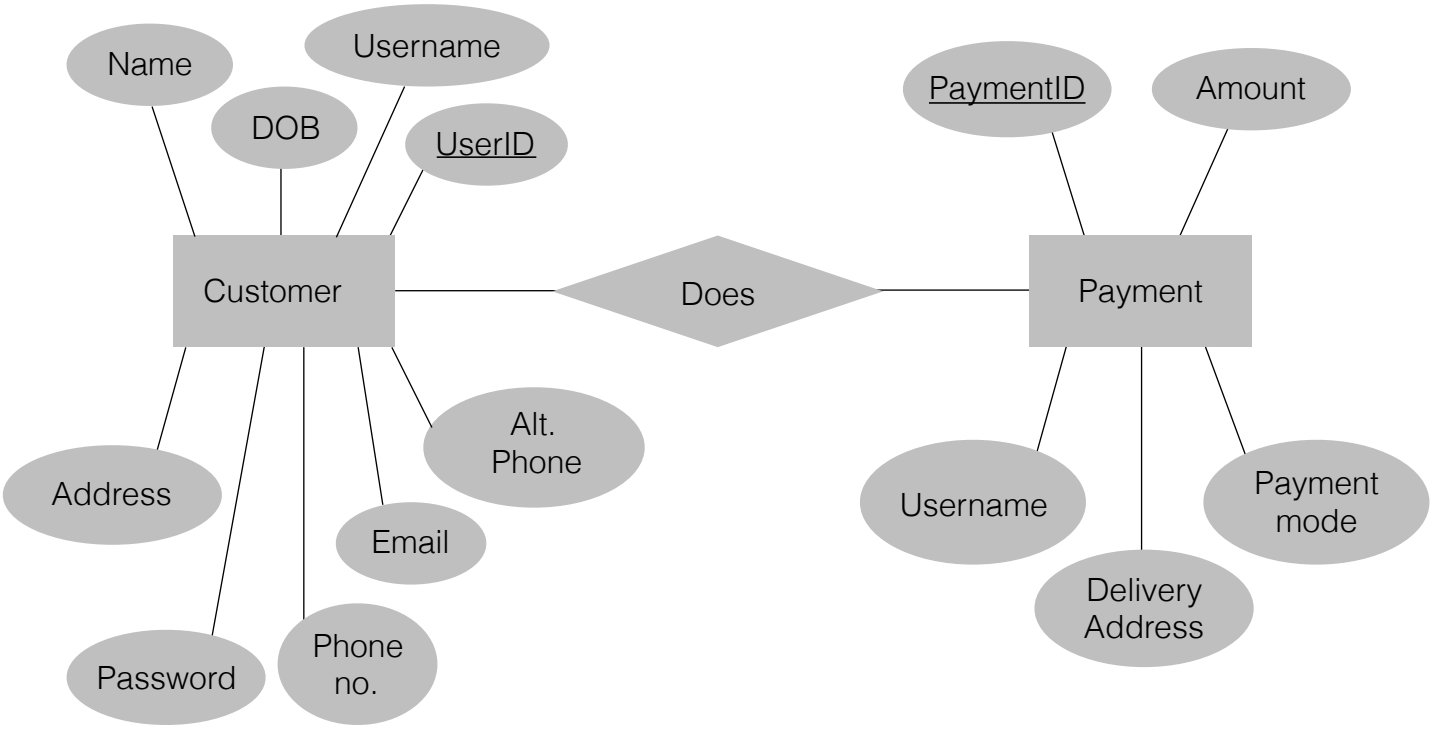
3. Placed after (R)

ER Diagrams:

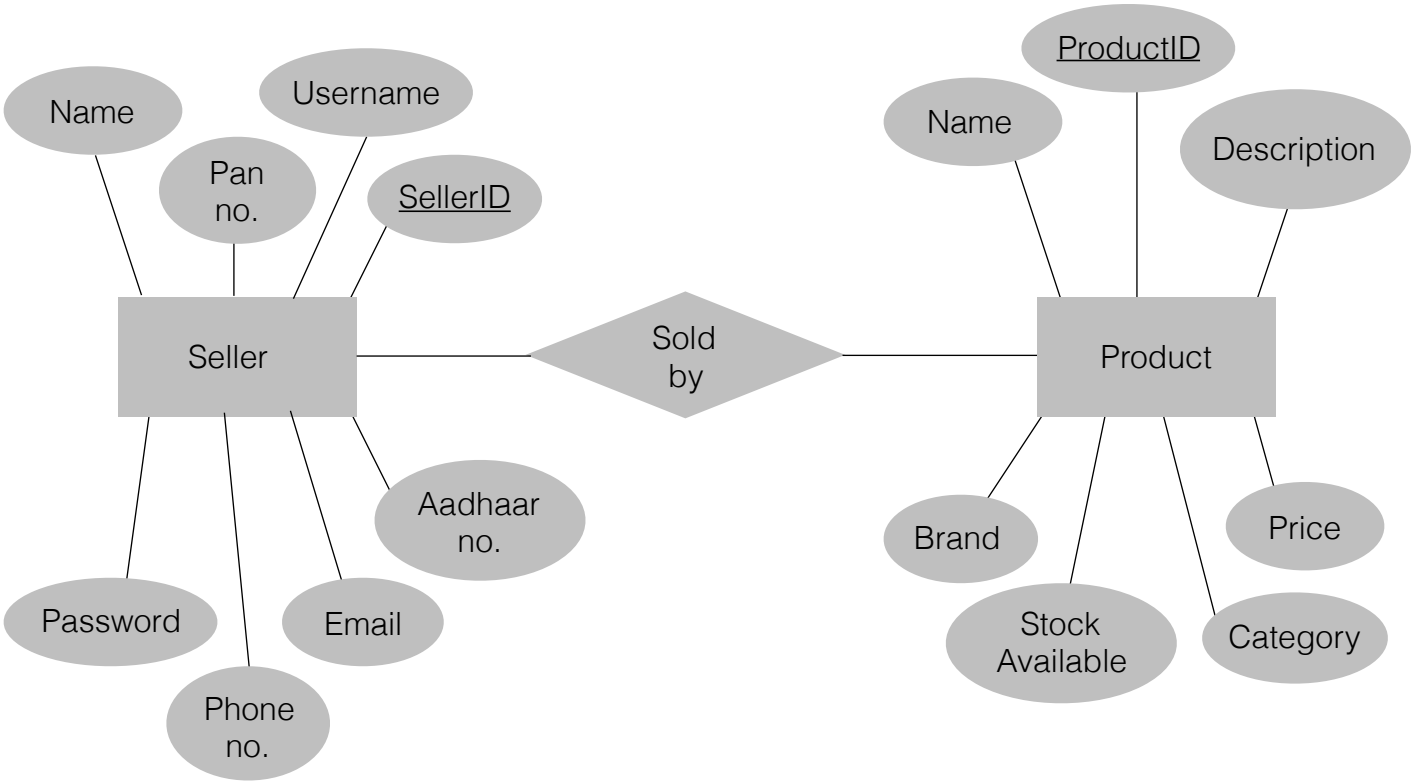
ER diagram for Purchases:



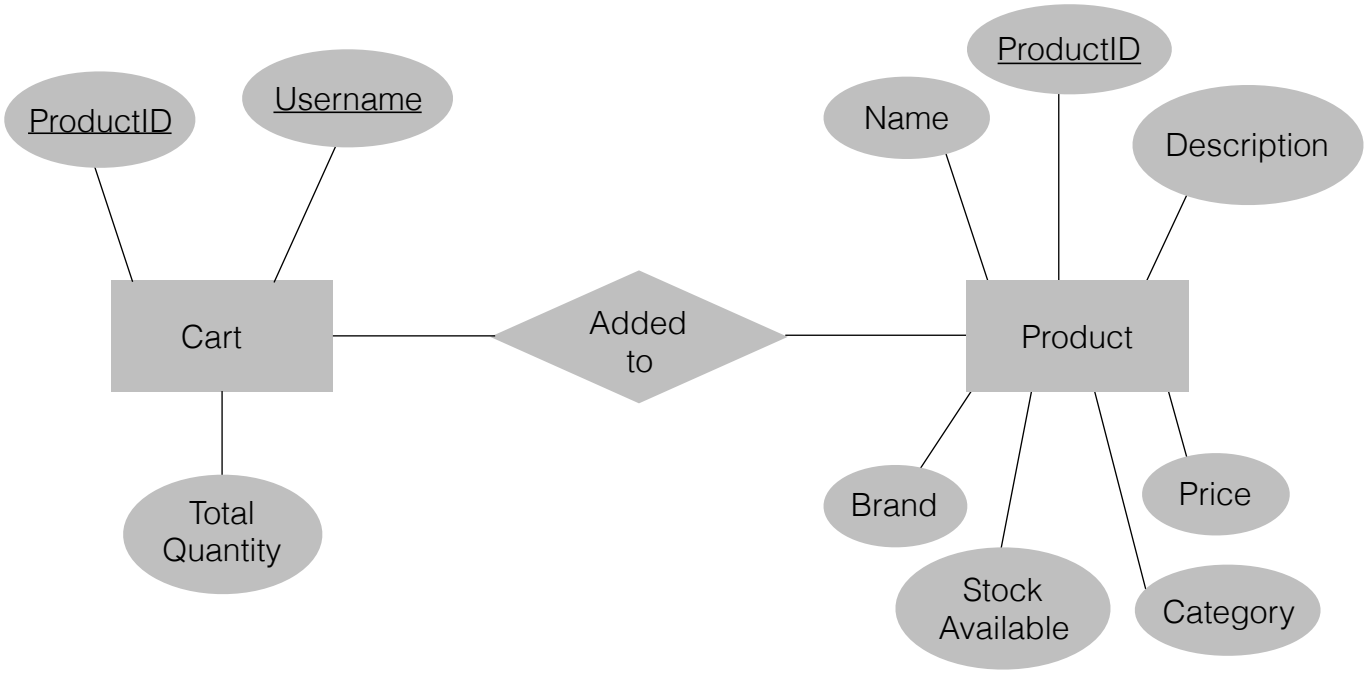
ER diagram for Payment:



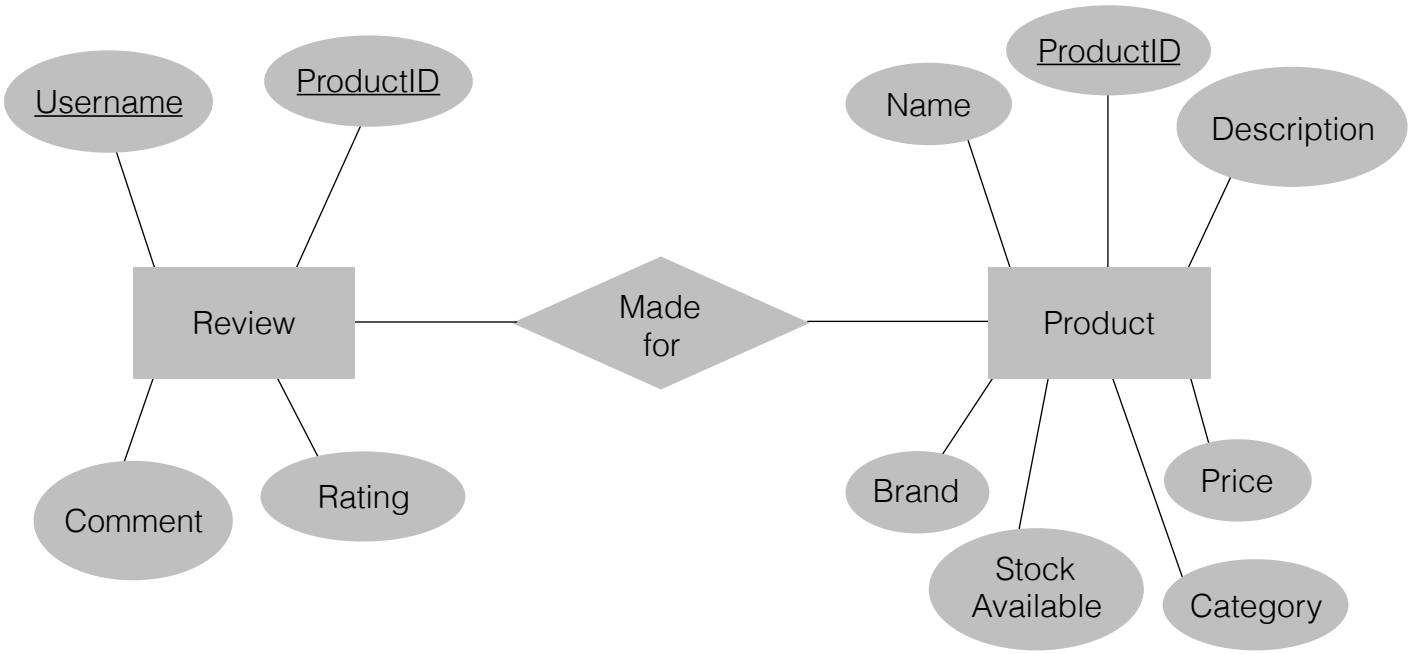
ER diagram for Selling:



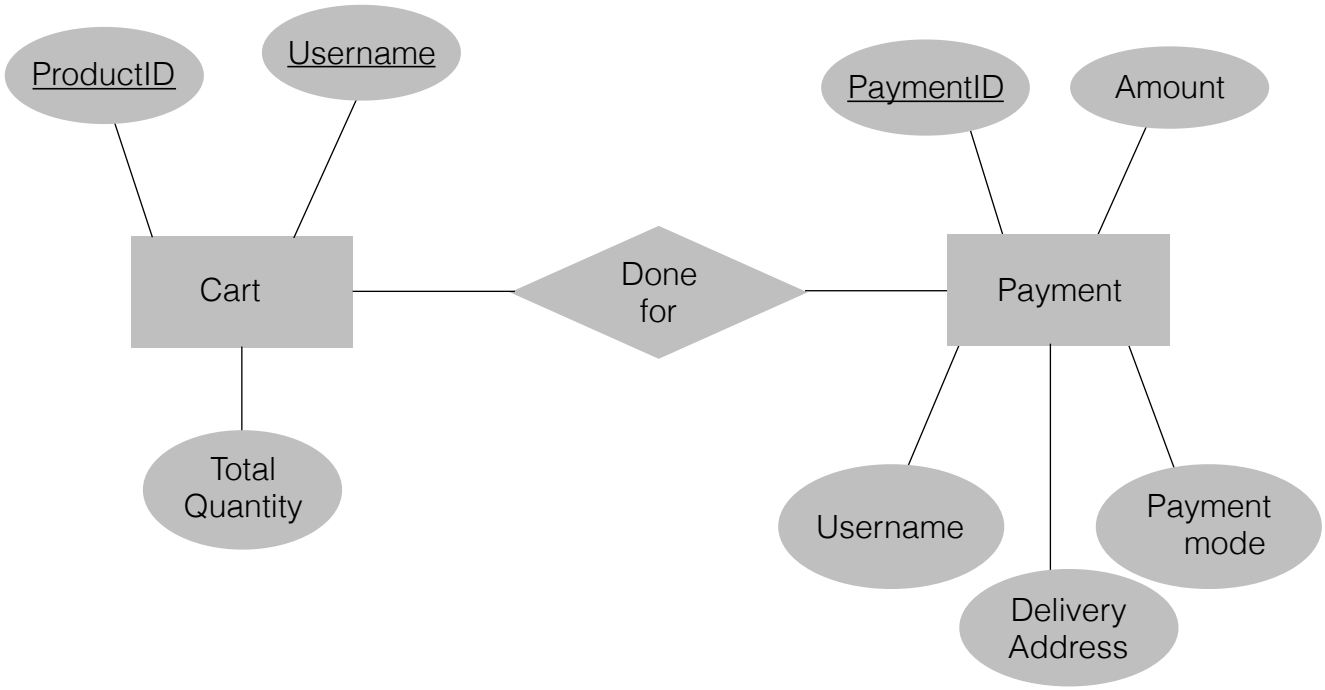
ER diagram for Addition to Cart:



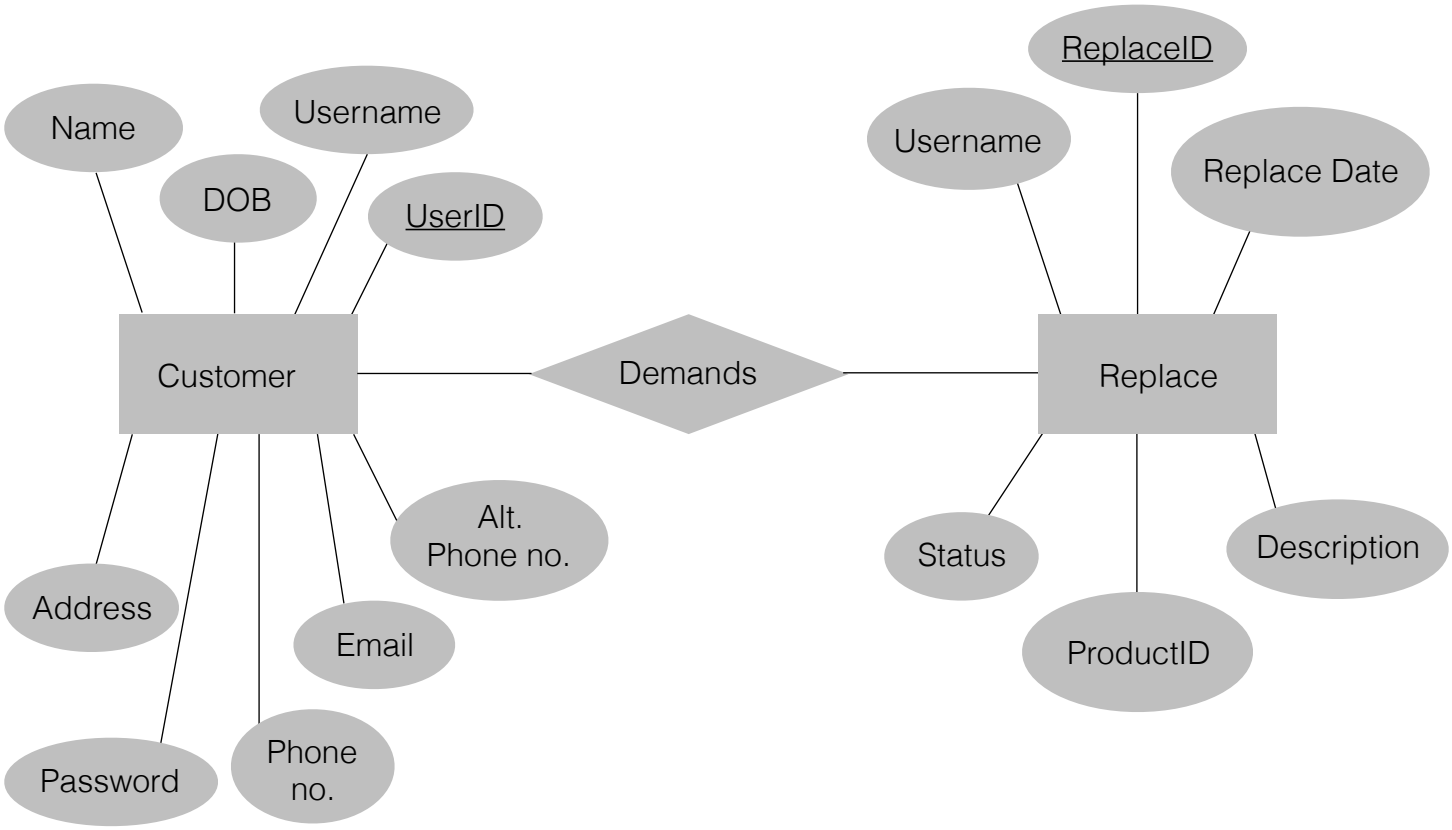
ER digram for Reviews:



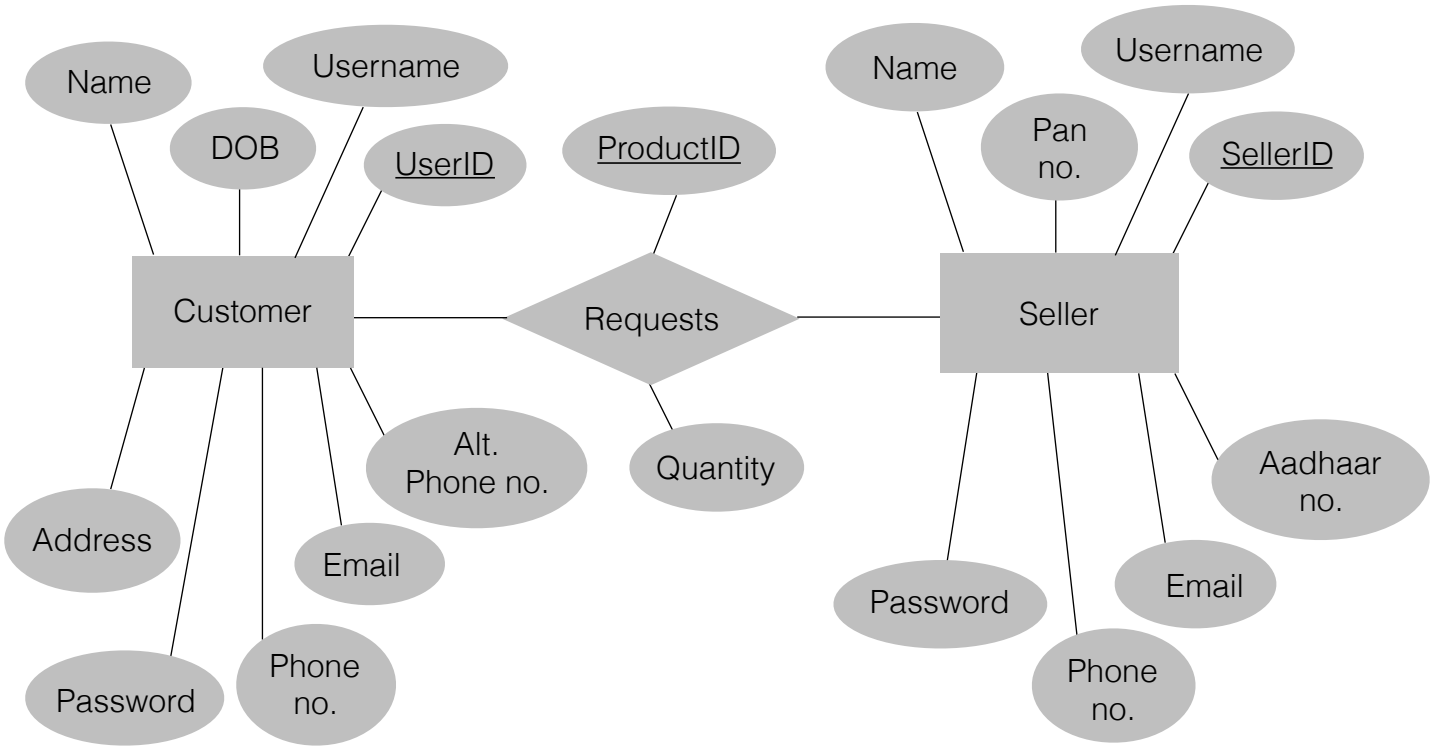
ER diagram for Cart Payment:



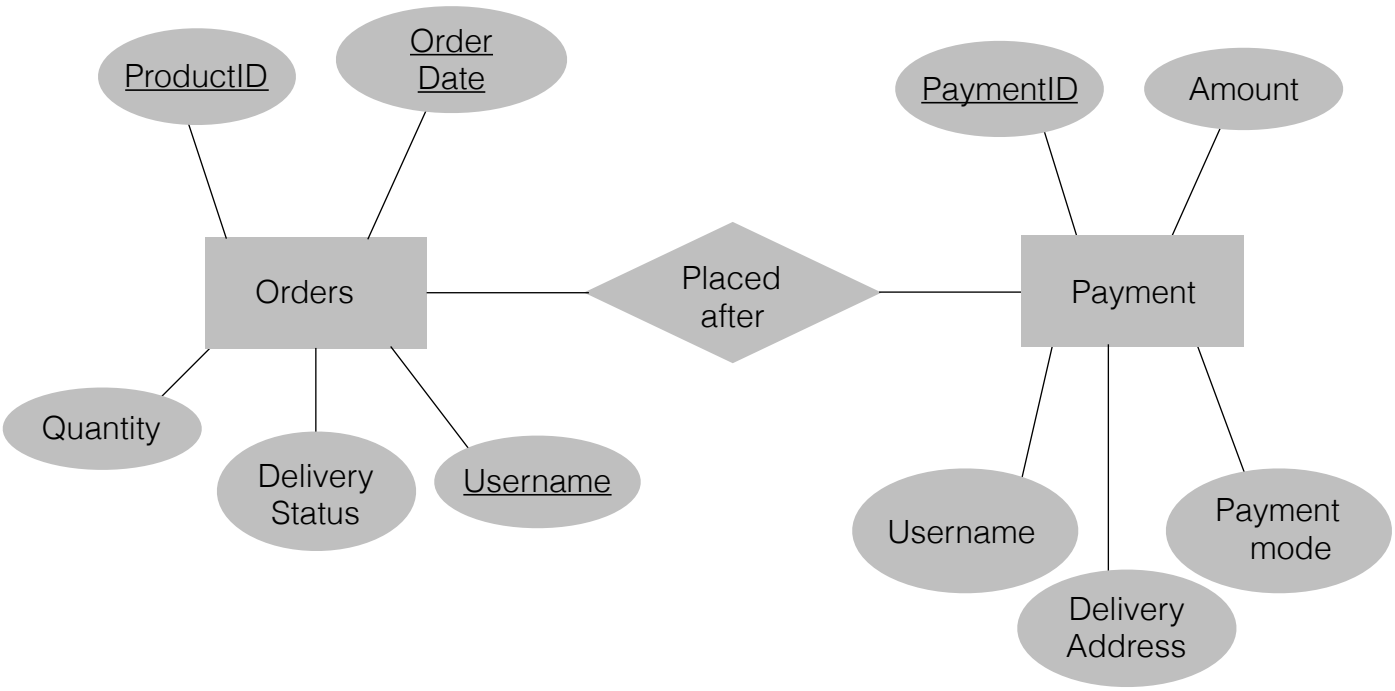
ER diagram for Product Replacement:



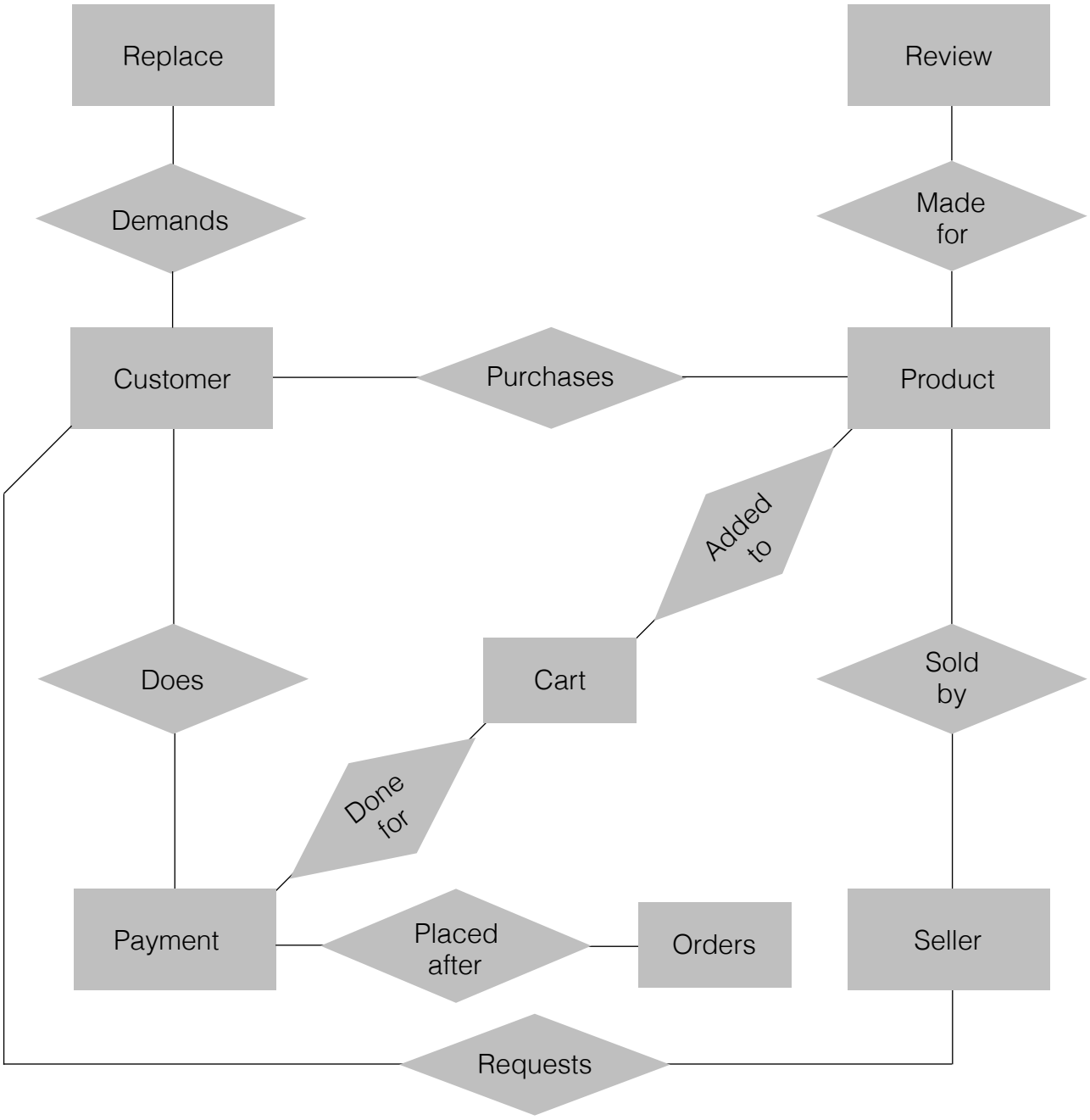
ER diagram for Product Request:



ER diagram for Order Placement:



ER diagram for whole Online Retail Portal:
(Attributes have not been attached)



Transformation of ER diagrams into set of Tables:

- **Customer**

```
CREATE TABLE `Customer` (
  `UserID` int(11) NOT NULL,
  `Username` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `DOB` date NOT NULL,
  `Password` varchar(100) NOT NULL,
  `Phone_no` bigint(10) NOT NULL,
  `Alt_Phone_no` bigint(10) NOT NULL,
  `House_no` text NOT NULL,
  `Locality` text NOT NULL,
  `City` text NOT NULL,
  `PIN` int(11) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=latin1;

  ALTER TABLE `Customer`
  ADD PRIMARY KEY (`UserID`),
  ADD UNIQUE KEY `Username` (`Username`),
  ADD UNIQUE KEY `Email` (`Email`),
  ADD UNIQUE KEY `Password` (`Password`),
  ADD UNIQUE KEY `Username_2` (`Username`),
  ADD UNIQUE KEY `Username_3` (`Username`,`Email`),
  ADD UNIQUE KEY `Email_2` (`Email`),
  ADD KEY `Username_4` (`Username`);

ALTER TABLE `Customer`
MODIFY `UserID` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=14;
```

- **Product**

```
CREATE TABLE `Product` (
  `Name_product` varchar(50) NOT NULL,
  `ProductID` int(11) NOT NULL,
  `Brand` varchar(20) NOT NULL,
  `Stock_available` int(11) NOT NULL,
  `Price` int(11) NOT NULL,
  `Description` varchar(100) NOT NULL,
  `Category` varchar(50) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=latin1;

  ALTER TABLE `Product`
  ADD PRIMARY KEY (`ProductID`);

ALTER TABLE `Product`
MODIFY `ProductID` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=17;
```

- **Payment**

```
CREATE TABLE `Payment` (
  `PaymentID` int(11) NOT NULL,
  `Username` varchar(30) NOT NULL,
  `Payment_mode` varchar(50) NOT NULL,
  `Amount` int(11) NOT NULL,
  `StreetAddress` varchar(30) NOT NULL,
  `City` varchar(30) NOT NULL,
  `State` varchar(30) NOT NULL,
  `PostalCode` int(6) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

  ALTER TABLE `Payment`
  ADD PRIMARY KEY (`PaymentID`);

ALTER TABLE `Payment`
```

```
MODIFY `PaymentID` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=3;
```

- **Seller**

```
CREATE TABLE `Seller` (
  `Name_Seller` char(100) DEFAULT NULL,
  `SellerID` int(11) NOT NULL,
  `Username` char(30) DEFAULT NULL,
  `Pan_Number` int(11) NOT NULL,
  `Aadhar_Number` int(11) DEFAULT NULL,
  `Email` char(50) DEFAULT NULL,
  `Password` char(50) DEFAULT NULL,
  `Phone_Number` char(15) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

  ALTER TABLE `Seller`
  ADD PRIMARY KEY (`SellerID`);

ALTER TABLE `Seller`
MODIFY `SellerID` int(11) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=4;
```

- **Cart**

```
CREATE TABLE `Cart` (
  `Username` varchar(50) NOT NULL,
  `ProdID` int(11) NOT NULL,
  `Total_Quantity` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

  ALTER TABLE `Cart`
  ADD PRIMARY KEY (`Username`,`ProdID`);
```

- **Review**

```
CREATE TABLE `Review` (
  `Username` varchar(45) NOT NULL DEFAULT '',
  `ProductID` int(11) NOT NULL DEFAULT '0',
  `Rating` int(11) DEFAULT NULL,
```

```
  `Comment` char(200) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

  ALTER TABLE `Review`
  ADD PRIMARY KEY (`Username`,`ProductID`);
```

- **Replacement**

```
CREATE TABLE `Replacement` (
  `ReplaceID` int(11) NOT NULL,
  `UserName` varchar(45) NOT NULL,
  `ProductID` int(11) NOT NULL,
  `Description` varchar(450) NOT NULL,
  `Status` varchar(5) NOT NULL,
  `Replace_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

  ALTER TABLE `Replacement`
  ADD PRIMARY KEY (`ReplaceID`);

ALTER TABLE `Replacement`
MODIFY `ReplaceID` int(11) NOT NULL AUTO_INCREMENT;
```

- **Requests**

```
CREATE TABLE `Requests` (
  `ProductName` varchar(45) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

  ALTER TABLE `Requests`
  ADD PRIMARY KEY (`ProductName`);
```

- **Orders**

```
CREATE TABLE `Orders` (
  `Username` varchar(45) NOT NULL,
  `ProductID` int(11) NOT NULL,
  `Quantity` int(11) NOT NULL,
  `Order_Date` date NOT NULL,
  `DeliveryStatus` varchar(5) NOT NULL
```

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `Orders`
ADD PRIMARY KEY (`Username`,`ProductID`,`Order_Date`);

Triggers:

- $result="CREATE TRIGGER stock_dec
  BEFORE INSERT ON Cart
  FOR EACH ROW
  UPDATE Product SET Stock_available=Stock_available-New.Total_Quantity WHERE ProductID = NEW.ProductID";

- $result="CREATE TRIGGER stock_upd
  BEFORE UPDATE ON Cart
  FOR EACH ROW
  UPDATE Product SET Stock_available=Stock_available-New.Total_Quantity +Old.Total_Quantity WHERE ProductID = NEW.ProductID";

- $result="CREATE TRIGGER stock_del
  BEFORE DELETE ON Cart
  FOR EACH ROW
  UPDATE Product SET Stock_available=Stock_available+Old.Total_Quantity WHERE ProductID = OLD.ProductID";

Events:

- CREATE DEFINER=`root`@`localhost` EVENT `update_delivery` ON SCHEDULE EVERY '0:1' MINUTE_SECOND STARTS '2016-11-17 13:16:55' ON COMPLETION NOT PRESERVE ENABLE DO update orders set DeliveryStatus = 'YES' where datediff(curdate(), Order_Date)>2

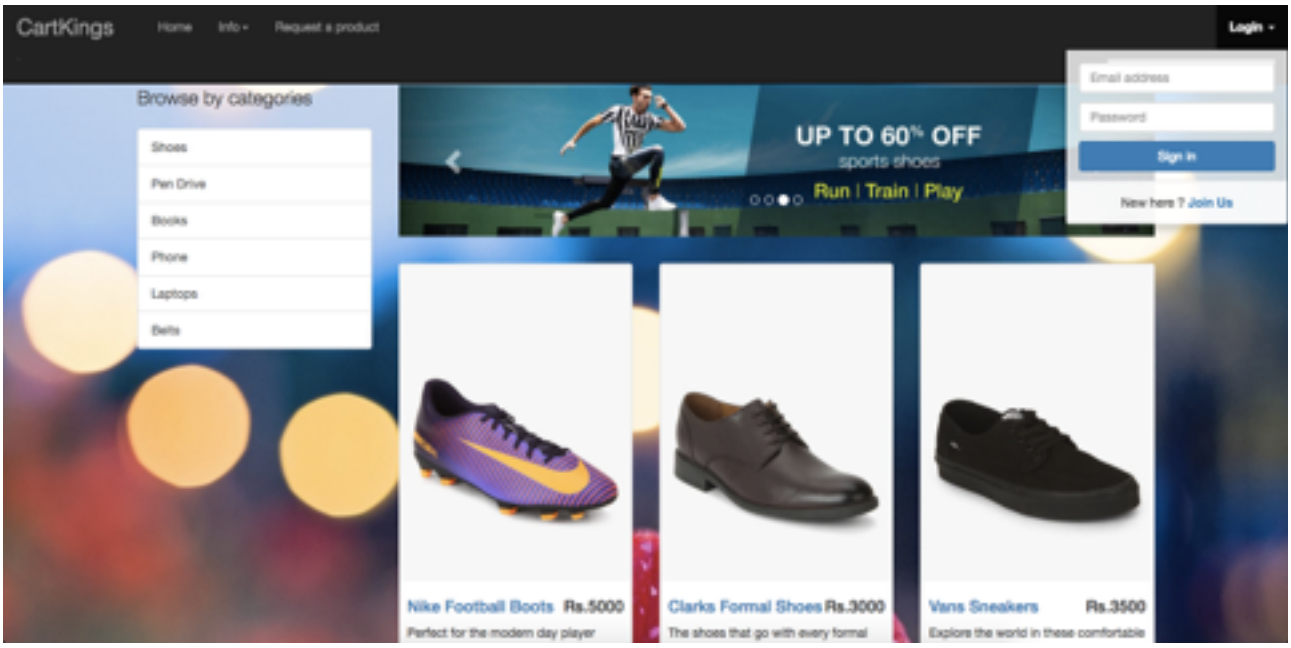
Procedure:

- CREATE PROCEDURE findTot(tot OUT INT) IS
  BEGIN
  tot=(SELECT SUM(Cart.Total_Quantity*Product.Price)
  FROM Cart
  LEFT JOIN Product
  ON Cart.ProductID = Product.ProductID WHERE Cart.Username='rahul')
  END

Features and functions of the Online Retail Portal:

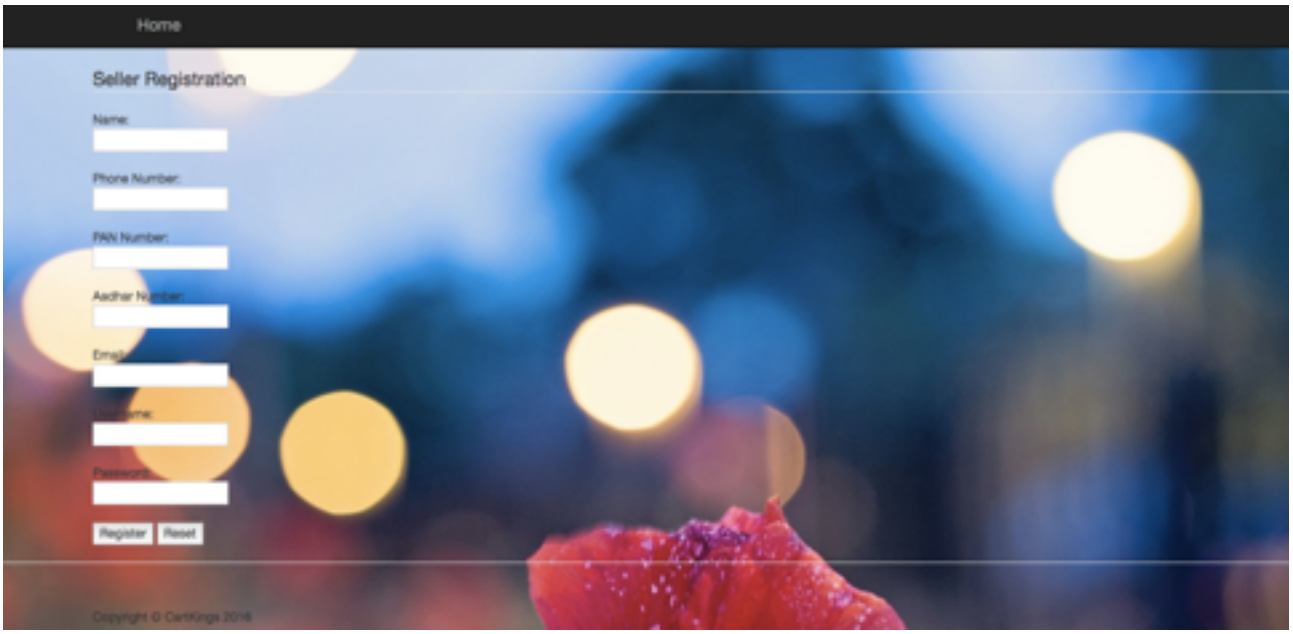
Customer Registration/Login:

In order to view the products, finalise his/her needs and eventually buy the goods, the customer needs to create an account. For this, he/she would be prompted to a signup page. The customer would have to enter his/her important details like Address(where the delivery has to be done), phone number,email address through which the customer can be uniquely located. An automatic UserID would be generated which would act as the primary key. All such data would be entered into the table with the help of sql insert query.



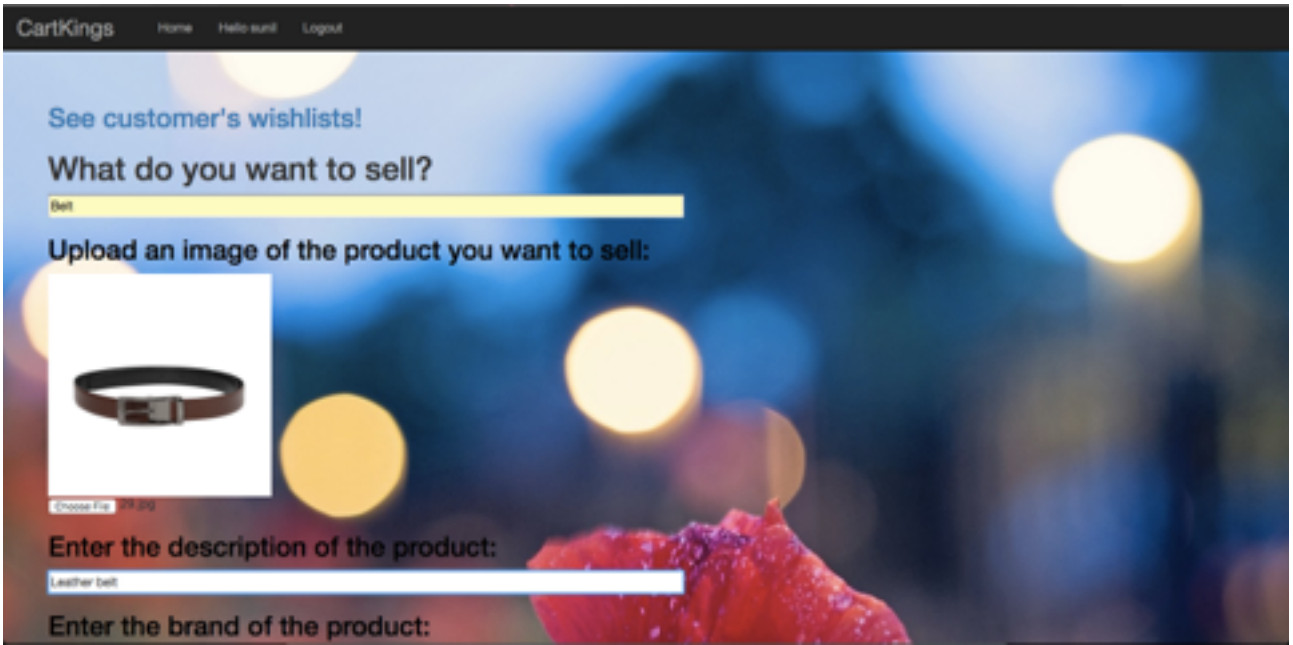
Seller Registration/Login:

Any seller who is willing to sell goods, is provided an option to create an account for himself. A seller_id is automatically generated which will be used as a primary key. If a seller wants to register himself, he will have to provide his verification details like Pan Card number and Aadhar card number. After providing all the required information, his entry will be made in the table which will be used for further needs throughout the database.



New product entry:

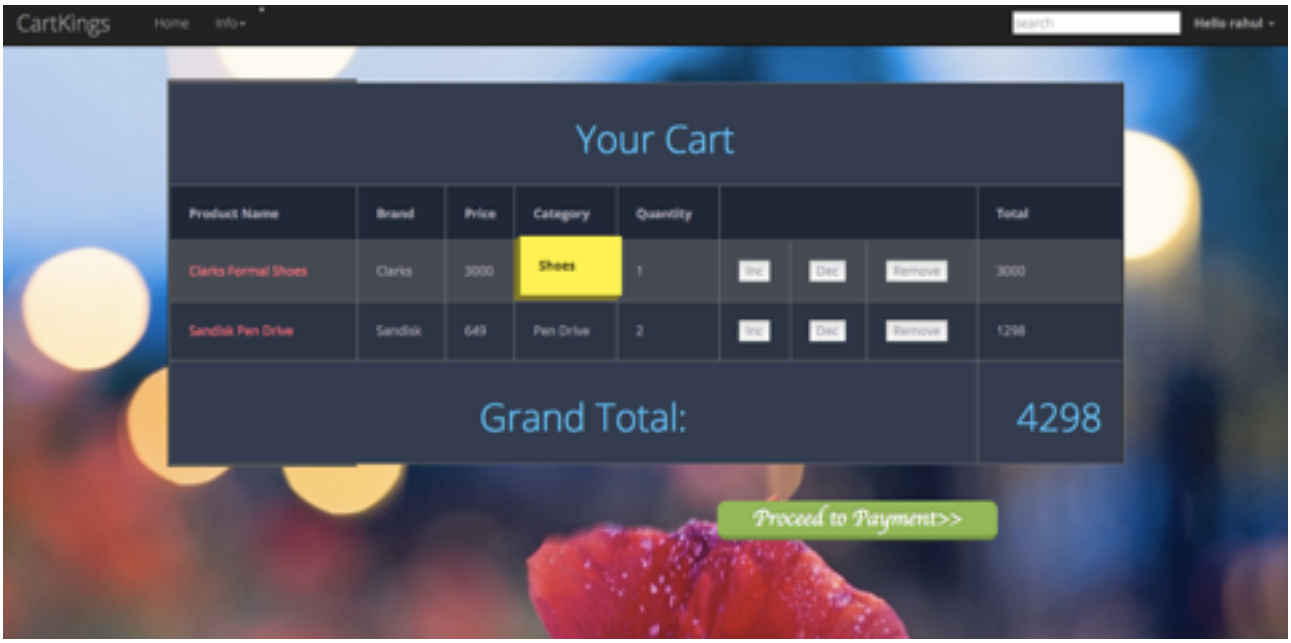
The seller who wishes to sell the products, needs to enter all the product details beforehand. Required information needs to be provided like the product



name, brand of the product and price. If the product is available before, the quantity of the product is increased by one. Otherwise, the product is added to the table of available products along with its ProductId which is generated automatically.

Cart Generation:

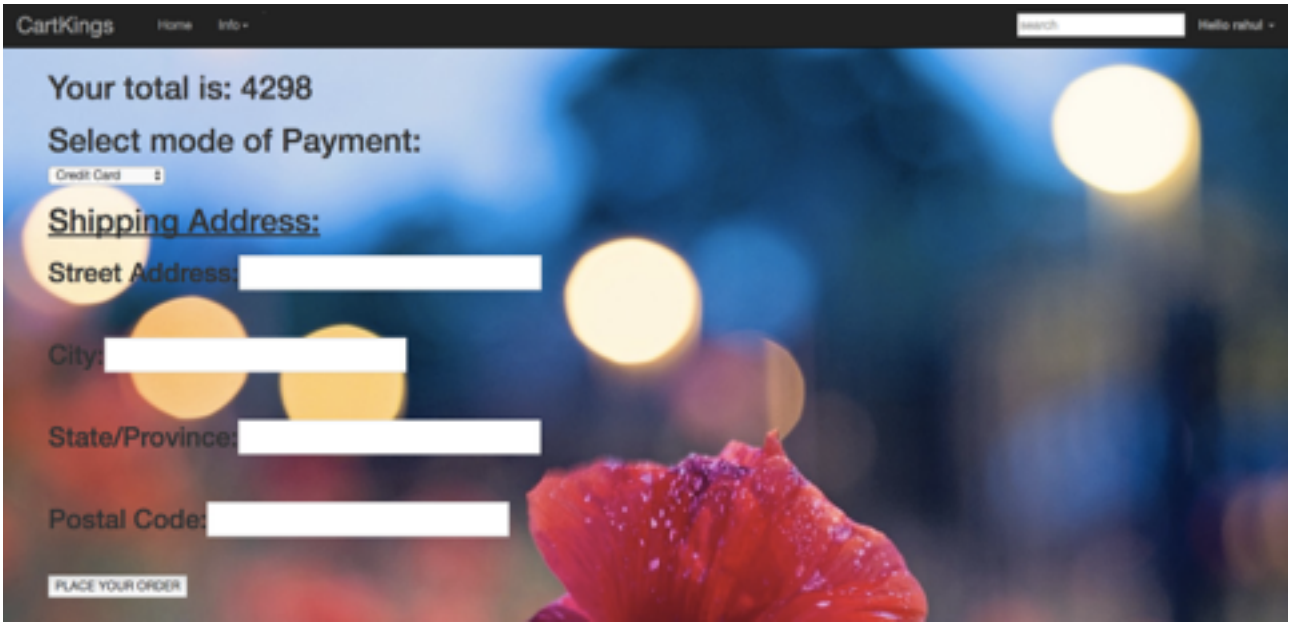
When the customer finalises a product, he would be provided an option of adding the product to the cart. Each cart would too have a Username and ProductID as a primary key so as to uniquely identify the cart for a particular user. The cart would have an attribute of cost which would be calculated as the sum of all products added to the cart along with the shipping charges(if any). Total quantity would also be calculated accordingly and thus the corresponding record of cart would be added to the table using the sql insert query.



Payment modes:

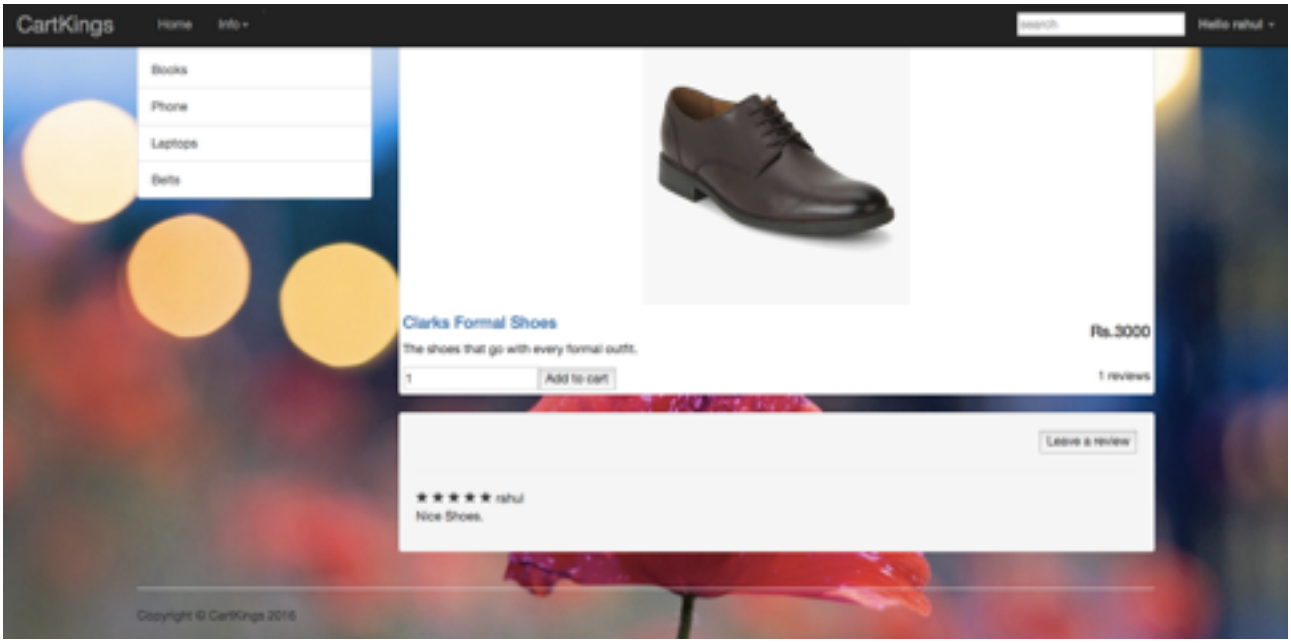
After the customer finishes selecting his/her products, he/she would be directed to the payment page wherein he would be asked for the payment mode. For online transactions, the transaction number would also be added to the database while for cash on delivery(COD), this field would be kept NULL in the database. Along with this, PaymentId for each payment done would be the generated and will be the primary key. Payment details like PIN CARD number and credit card/debit card number for online transactions would be handled appropriately thereafter. After the payment is done, the cart is emptied,

corresponding entries are entered into the payment table and the orders table. The product will be delivered within 2 days.



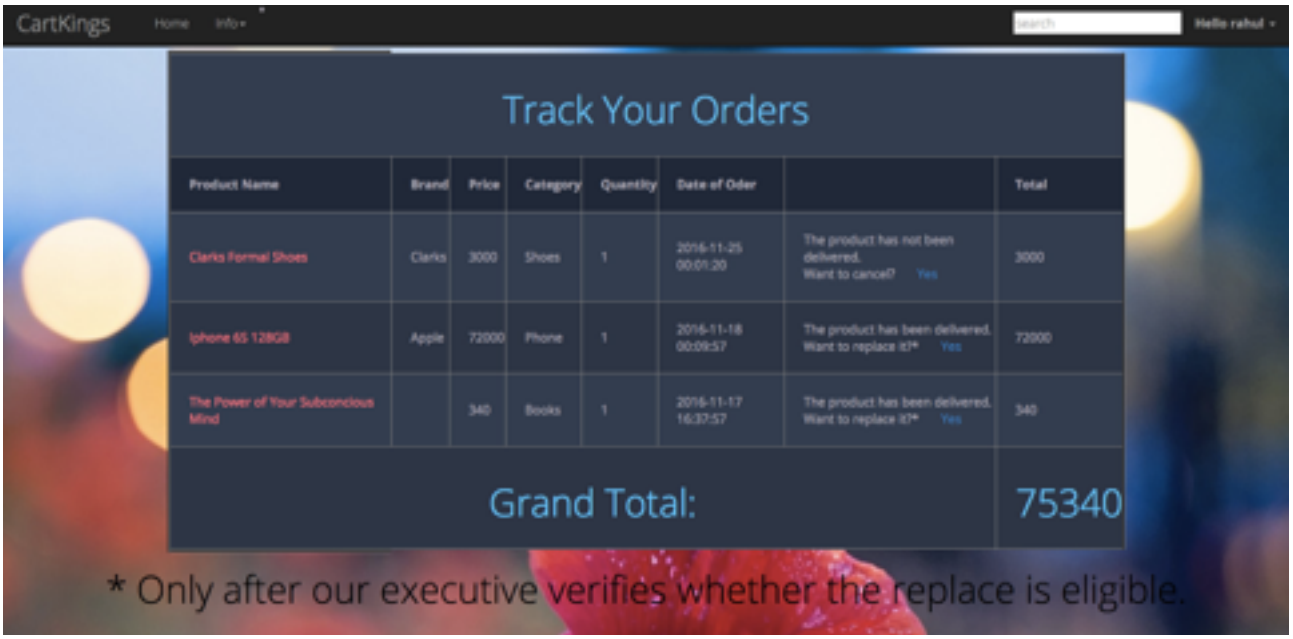
Review options:

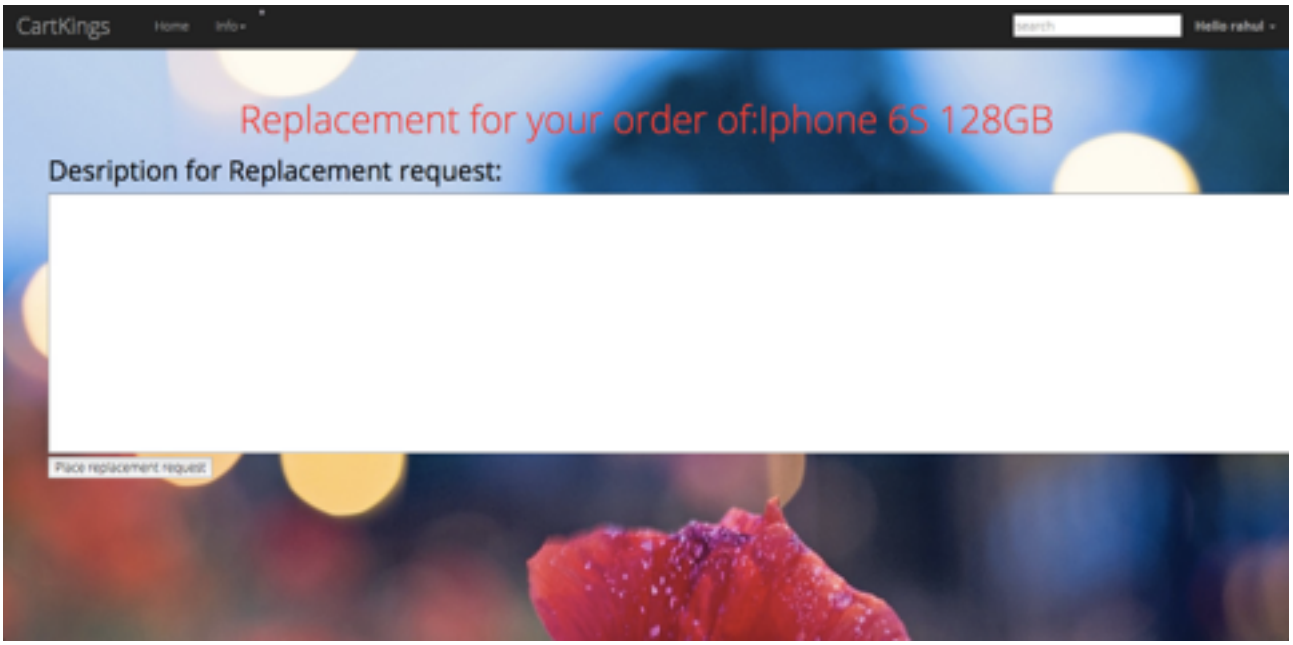
The customers after the purchase of the product would be provided an option of giving their feedbacks and ratings. The customer would be asked to give their response and rating of the product. Username and ProductID would be used as primary keys so as to identify the customer's review details. Also, a user can give a review only if he/she has bought the product already. This is done by using the sql CHECK constraint.



Replacement Procedure:

If the customer claims a replacement for the ordered product, he/she would be provided a replacement page. He would be requested to input his UserId and the productId for the unique identification of the user as well as the product. He/She should then describe the issue. Also, there would be a status attribute added which would be filled by moderators about whether the customer claim should be accepted or not. Along with this, each replacement claim would have a unique replaceId associated with it. The customer can ask for replacement only after the product has been delivered. After 5 days of replacement, the product will be delivered back and will be added to the orders of the customer.





Requests:

If the customer requires a product which is not currently in stock, he/she would be provided an option where he/she can request that product. He/She would have to enter the productId for the required product as well as the quantity required. If the product is requested again, the quantity would be increased by one. After the quantity for a product crosses a certain threshold the seller may want to make the product available for purchase.