

# ONLINE RETAIL MANAGEMENT SYSTEM

BY

AISHWARYA SRINIVASAN	- 13BCE1006
SHASHANK DEVISETTY	- 13BCE1036
HARSHWARDHAN AGARWAL	- 13BCE1052
SHASHANK M V	- 13BCE1081

*A project submitted to the*  
**SCHOOL OF COMPUTING SCIENCES AND ENGINEERING**  
*in partial fulfillment of the award for the degree*  
*of*  
**BACHELOR OF TECHNOLOGY**  
*in*  
**COMPUTER SCIENCE ENGINEERING**



VELLORE INSTITUTE OF  
TECHNOLOGY  
CHENNAI CAMPUS  
NOV, 2016

## ACKNOWLEDGEMENT

Course Code: CSE328

Course Title: SOFTWARE PROJECT MANAGEMENT

This project report entitled “**ONLINE RETAIL MANAGEMENT SYSTEM**” is a bonafide work of **AISHWARYA SRINIVASAN (13BCE1006)**, **SHASHANK. D (13BCE1036)**, **HARSHWARDHAN AGARWAL (13BCE1052)** and **SHASHANK. M. V (13BCE1081)** and had been carried out under the supervision and guidance of **Prof. Aparna. V**. We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Prof. Aparna. V**, Assistant Professor, School of Computing Sciences and Engineering for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

**SHASHANK. D**  
**SHASHANK. M. V**

**HARSHWARDHAN AGARWAL**  
**AISHWARYA SRINIVASAN**

# ABSTRACT

Online Retail Management Software is a software application to be developed for the use in any retail unit/shop. The software aims at computerizing all the activities related to the retail unit. It should be a comprehensive one to cover all the aspects of a retail shop.

# INTRODUCTION

Online Retail Management System is a form of electronic shopping store where the buyer interacts directly with the seller usually via the internet. There is no intermediary service. The sale and purchase transaction is completed electronically and interactively in real-time. The development of this new system contains the following activities, which try to develop on-line application by keeping the entire process in the view of database integration approach.

**Administrator** of Online Retail Management System has multiple features such as Add,Delete, Update shopping Items.

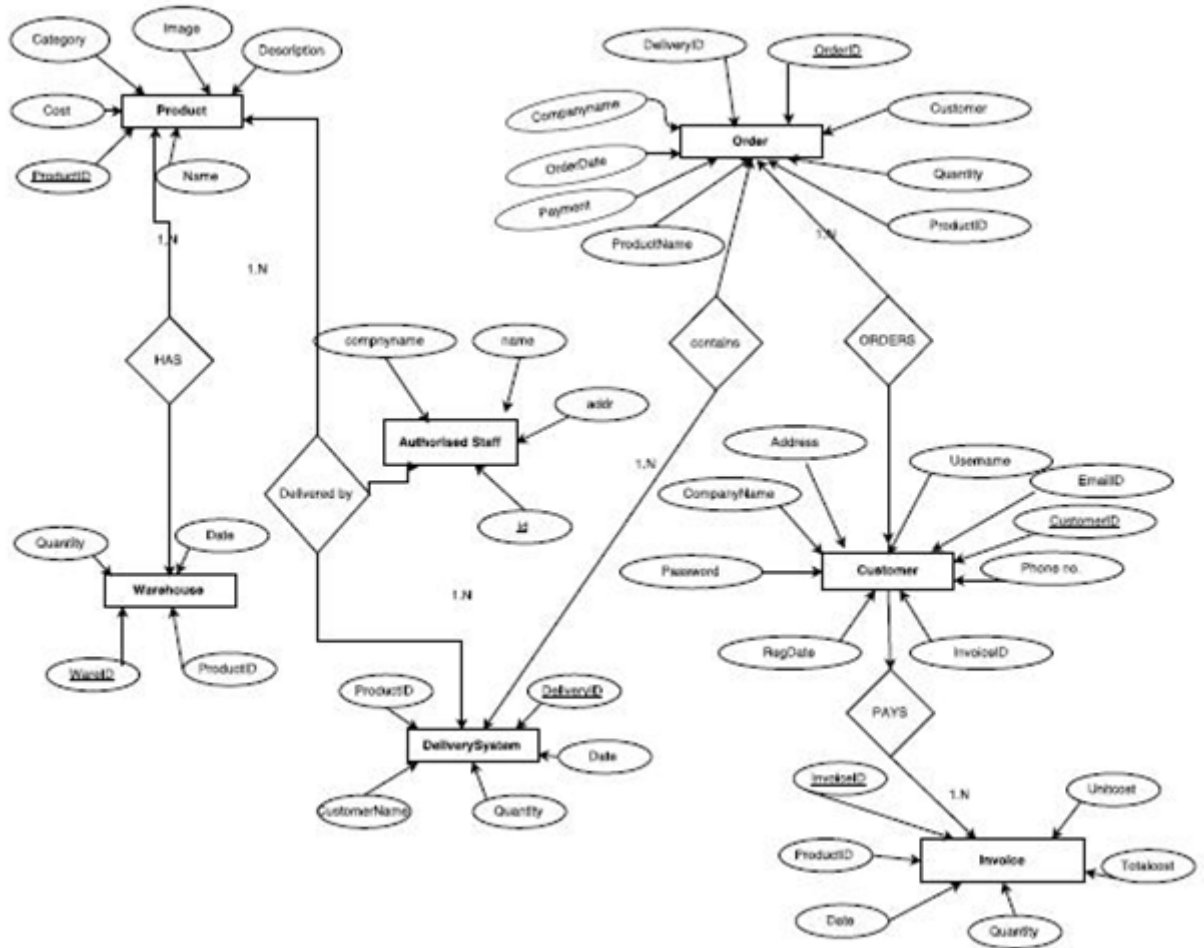
**Customer** of Online Retail Management System has multiple features such as Sign up,Login, Shop online, Get an invoice.

## Features of Online Retail Management System:

1. Secure registration and profile management facilities for Customers.
2. Browsing through the e-Mall to see the items that are there in each category of products like Apparel, Kitchen accessories, Bath accessories, Food items etc.
3. Creating a Shopping cart so that customer can Shop N number of items and checkout finally with the entire shopping cart
4. Customers should be able to mail the Shop about the items they would like to see in the Shop
5. Secured mechanism for checking out from the Shop ( Credit card verification mechanism ). Updates to customers about the Recent Items in the Shop.

6. Uploading Most Purchased Items in each category of products in the Shop like Apparel, Kitchen accessories, Bath accessories, Food items etc.

## ER DIAGRAM



## MODULES

Our project consists of several modules namely,

- Administrator
  - Add a product
  - Modify a product
  - Delete a product
  - Search a product

- Customer
  - Login
  - Sign up
  - Shop products
  - Generate Invoice

## SQL

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database. This tutorial will provide you with the instruction on the basics of each of these commands as well as allow you to put them to practice using the SQL Interpreter.

## ESTABLISHING CONNECTION WITH DATABASE

```
$row_count=0;
$host="localhost";
$user="harsh";
$password="";
$database="mydb";
$connect=mysqli_connect($host,$user,$password,$database);
if($connect)
    { //echo "<p align='center'>Connected to the server...!!";
    }
else
    die(mysqli_error());

/*$select = mysql_select_db("mydb");
if($select)
    { //echo "<p align='center'>Selected Database...!!";
    }
else
    die(mysql_error());*/
```

## Administrator:

### 1. Add a product

```
$create = "CREATE TABLE product(
    p_id VARCHAR(30) PRIMARY KEY,
    p_name VARCHAR(30) NOT NULL,
    p_cost INT(6) NOT NULL,
    p_category VARCHAR(30) NOT NULL,
    p_quantity INT(4) NOT NULL,
    p_description VARCHAR(150) NOT NULL
)";

$result = mysqli_query($connect,$create);
if($result)
    echo " Table created...!!";
else
    die(mysqli_error($connect));

$insert = "INSERT INTO product (p_name, p_cost, p_category, p_quantity, p_description)
VALUES ('$pname', '$scost', '$scategory','$squantity', '$sdescription')";
$result = mysqli_query($connect,$insert);
if($result)
    echo " Your data is entered into the database...!!";
else
{
    // echo "Enter a unique Product ID";
    // exit();
    die(mysqli_error($connect));
}
```

### 2. Modify a Product

```
$query = "SELECT*
FROM product
WHERE id='$mpid1'";

$result = mysqli_query($connect,$query) or die(mysqli_error());

while($rows = mysqli_fetch_assoc($result))
{
    $row_count++;
    extract($rows);

$update="UPDATE product
SET id='$mpid2',
    p_name = '$mpname' ,
    p_cost = '$mcost' ,
    p_category = '$mcategory' ,
    p_quantity = '$mquantity' ,
    p_description = '$mdescription'
WHERE id = '$mpid3' ";
$result=mysqli_query($connect,$update) or die(mysqli_error($connect));
if($result)
    echo " <p align='center'>Your data is modified into the database...!!";
else
    die(mysqli_error());
```

### 3. Delete a Product

```
$query = "SELECT*
          FROM product
          WHERE p_id='$pid'";

$result = mysqli_query($connect,$query) or die(mysqli_error());

$delete = "DELETE FROM product
           WHERE p_id = '$pid' " ;
$result = mysqli_query($connect,$delete);
if($result)
{
    echo "<p align='center'>Record Deleted..!!</p>";
}
```

### 4. Search a Product

```
$query = "SELECT*
          FROM product
          WHERE p_name='$sname'";

$result = mysqli_query($connect,$query) or die(mysqli_error());
```

## Customers:

### 1. Sign Up

```
$create = "CREATE TABLE customer(
    fname VARCHAR(30) NOT NULL,
    lname VARCHAR(30) NOT NULL,
    username VARCHAR(30) PRIMARY KEY NOT NULL,
    email VARCHAR(30) NOT NULL,
    phone BIGINT(10) NOT NULL,
    password VARCHAR(50) NOT NULL,
    address VARCHAR(50) NOT NULL
)";

$result = mysqli_query($connect,$create);
if($result)
    echo"table created";
else
    die(mysqli_error($connect));

$data = "INSERT INTO customer VALUES
('$fname','$lname','$username','$email','$phone','$userpassword','$address')";

$result = mysqli_query($connect,$data);
if($result)
{
    echo "<p align='center'>Congratulations!! Your account have been created!!!<br>";
}
```

### 2. Login

```

$myusername = mysqli_real_escape_string($connect,$_POST['username']);
$mypassword = mysqli_real_escape_string($connect,$_POST['password']);

// Retrieve username and password from database according to user's input
$login = mysqli_query($connect, " SELECT email FROM customer WHERE username = '$myusername'
and password = '$mypassword' ") or die(mysqli_error($connect));

```

### 3. Shop Products

```

$query = "SELECT * FROM product WHERE p_category = 'beverages'";
$result = mysqli_query($connect,$query) or die(mysqli_error());

```

```

$query = "SELECT * FROM product WHERE p_category = 'snacks'";
$result = mysqli_query($connect,$query) or die(mysqli_error());

```

```

$query = "SELECT * FROM product WHERE p_category = 'Fruits&Vegetables'";
$result = mysqli_query($connect,$query) or die(mysqli_error());

```

```

$query = "SELECT * FROM product WHERE p_category = 'grocery'";
$result = mysqli_query($connect,$query) or die(mysqli_error());

```

```

$query = "SELECT * FROM product WHERE p_category = 'others'";
$result = mysqli_query($connect,$query) or die(mysqli_error());

```

Also “**Recommendations**” have been provided based on the past purchases of a particular customer and also based on table-mapping i.e. for example, if a customer buys shoes, the portal would recommend socks for him/her.

```

$query = "SELECT *
FROM product
WHERE p_name = (SELECT product_name FROM invoice WHERE customer_email = '$email'
AND product_quantity = (SELECT max(product_quantity) FROM invoice WHERE customer_email = '$email'))";

```



#### 4. Generate an invoice

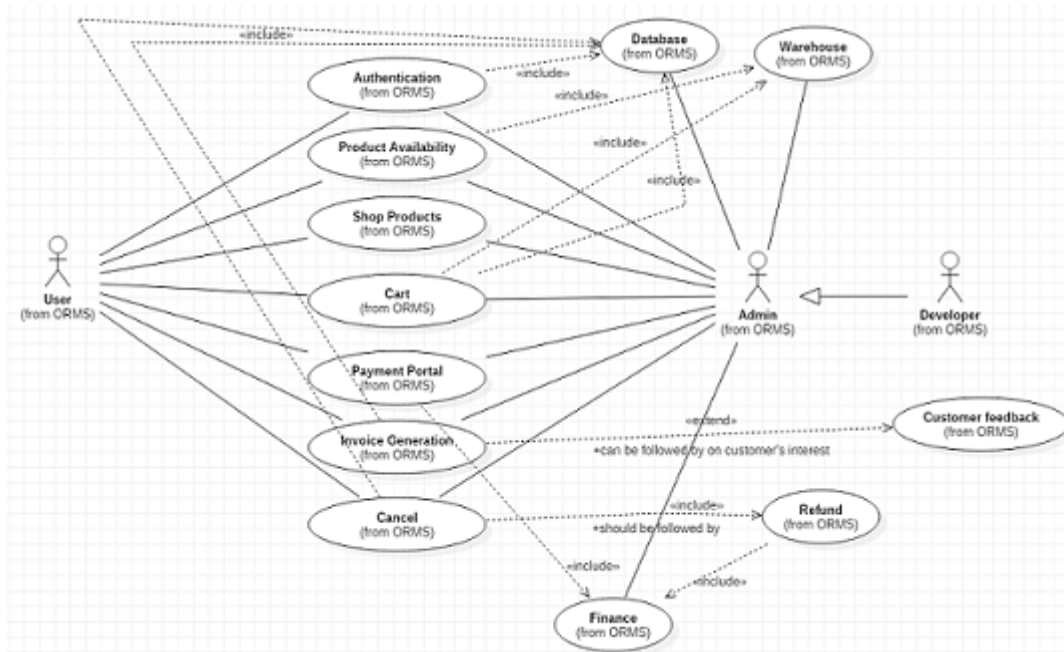
```
$create = "CREATE TABLE invoice(
    order_date DATE NOT NULL,
    customer_email VARCHAR(50) NOT NULL,
    product_name VARCHAR(30) NOT NULL,
    product_category VARCHAR(40) NOT NULL,
    product_cost INT(10) NOT NULL,
    product_quantity INT(10) NOT NULL,
    total_cost int(10) NOT NULL
)";

$result = mysqli_query($connect,$create);
if($result)
    echo"table created";
else
    die(mysqli_error($connect));
```

```
$query = "SELECT*
FROM product
WHERE id='$myid'";
$result = mysqli_query($connect,$query) or die(mysqli_error());
while($rows = mysqli_fetch_assoc($result))
{
    extract($rows);
    $total = $p_cost * $myquantity;

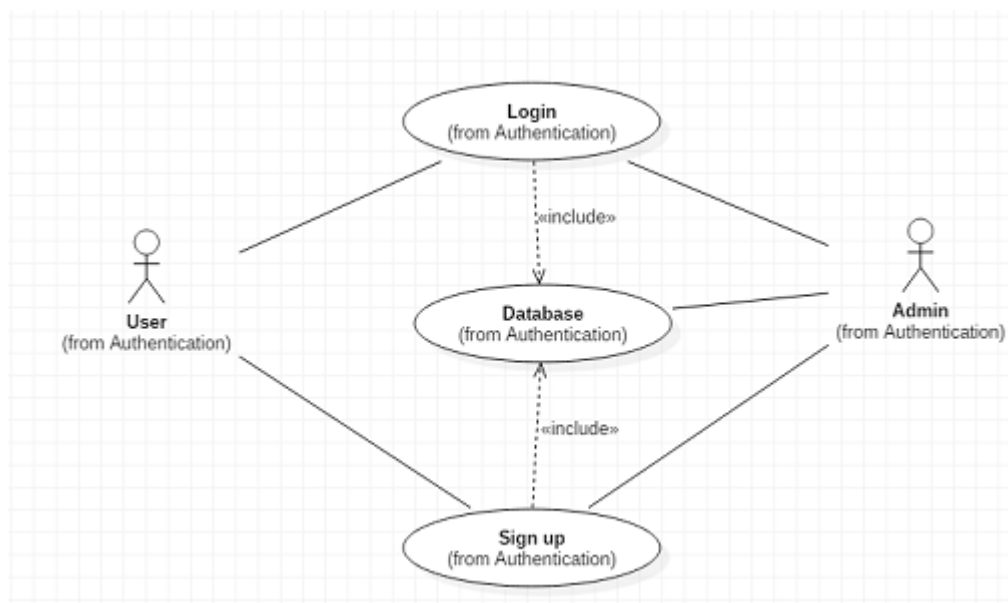
    $query = "INSERT INTO invoice (order_date,customer_email,product_name,product_category,
product_cost,product_quantity,total_cost) VALUES
(now(),'$email','$p_name','$p_category','$p_cost,$myquantity,$total)";
    $result = mysqli_query($connect,$query);
    if($result)
        echo"RECORD ENTERED!!!";
    else
        die(mysqli_error($connect));
}
```

# USECASE DIAGRAM

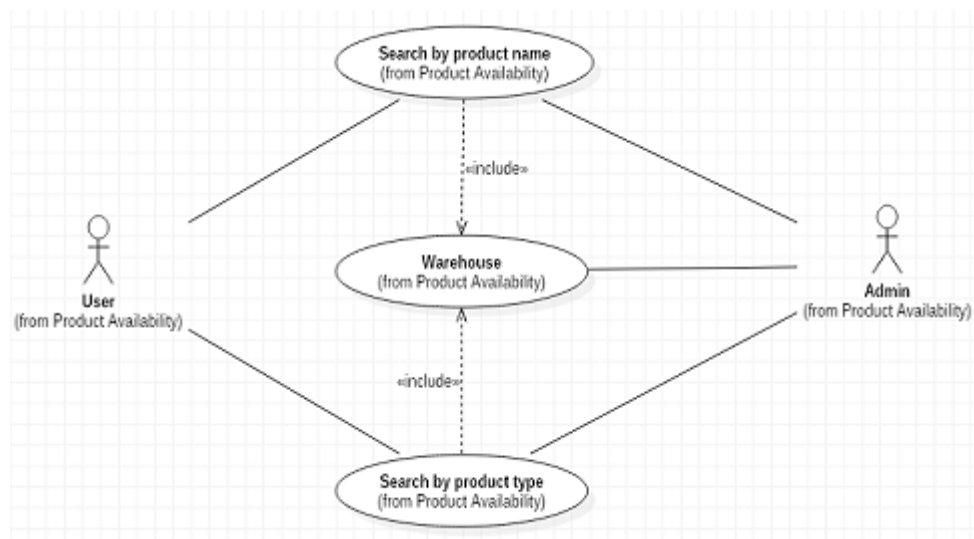


## SUB-USECASE DIAGRAMS

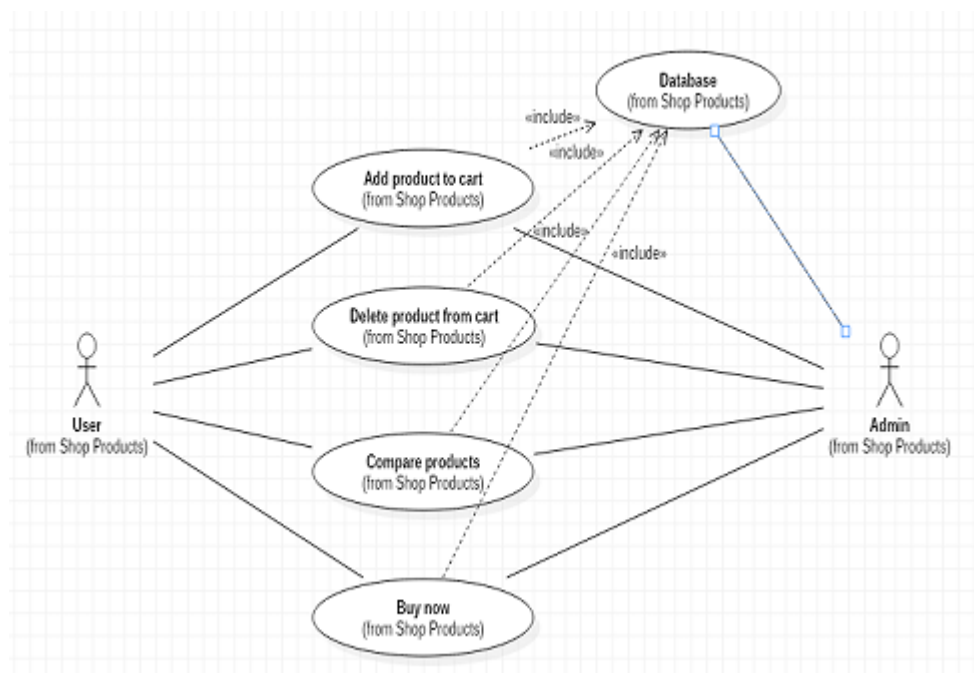
## Authentication:



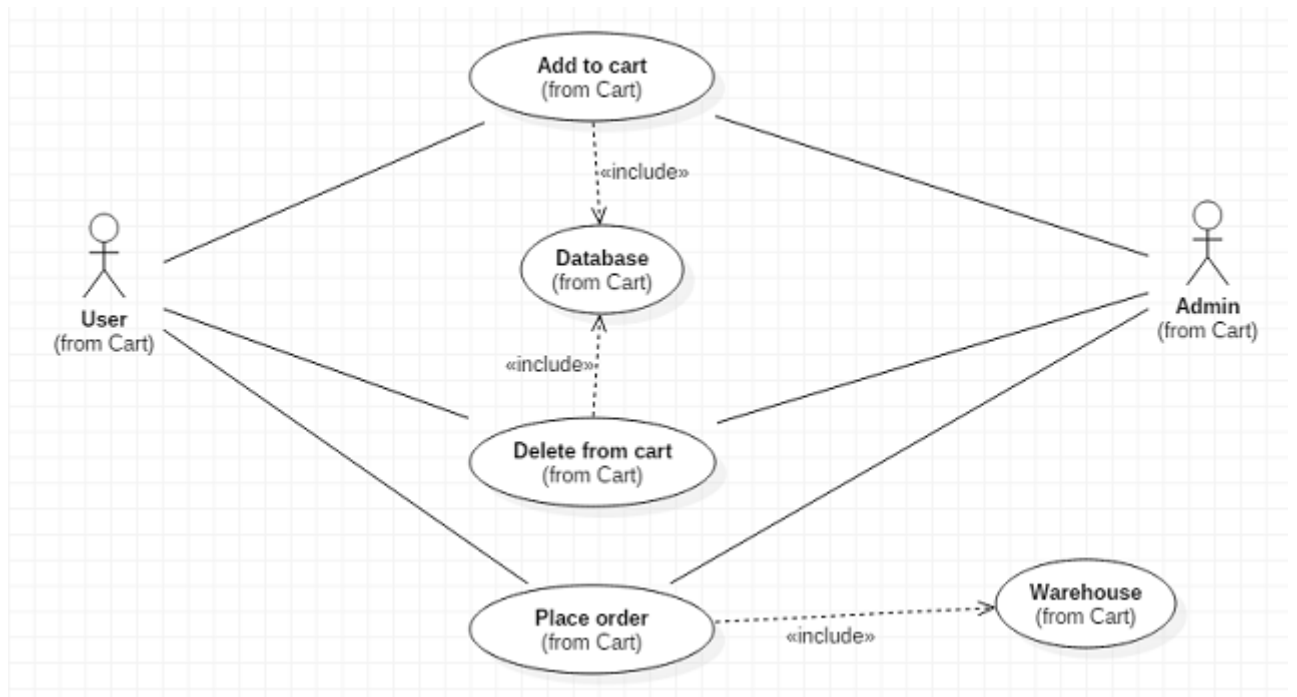
## Product Availability:



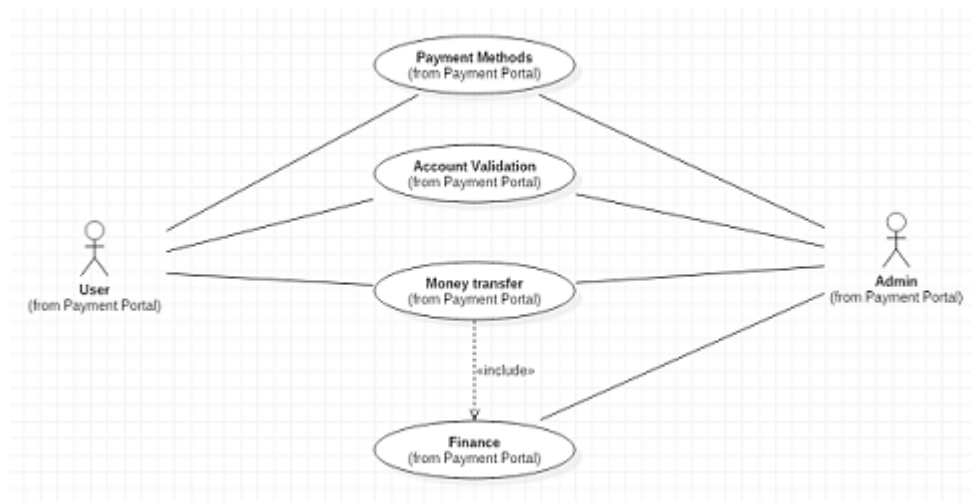
## Shop Products:



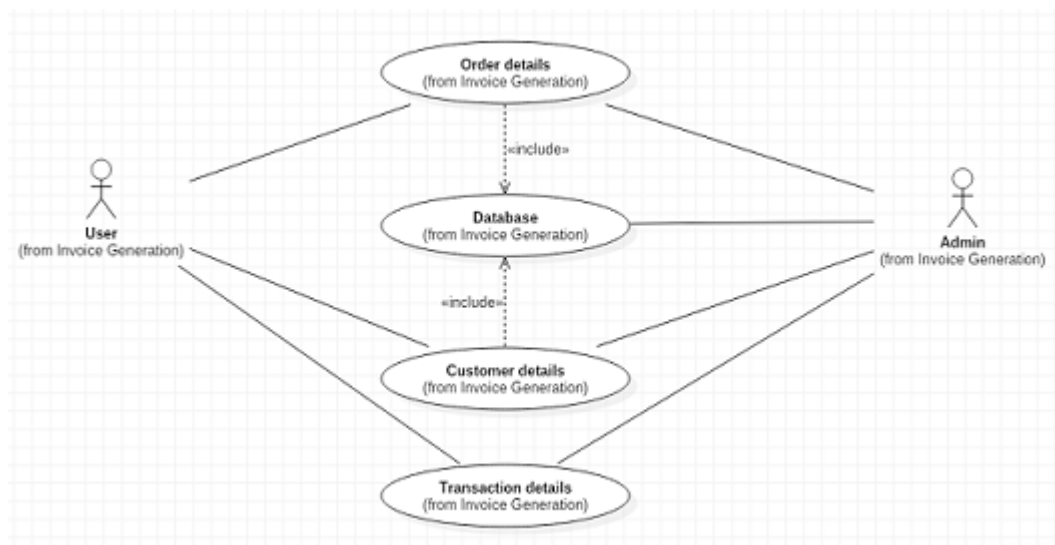
## Cart:



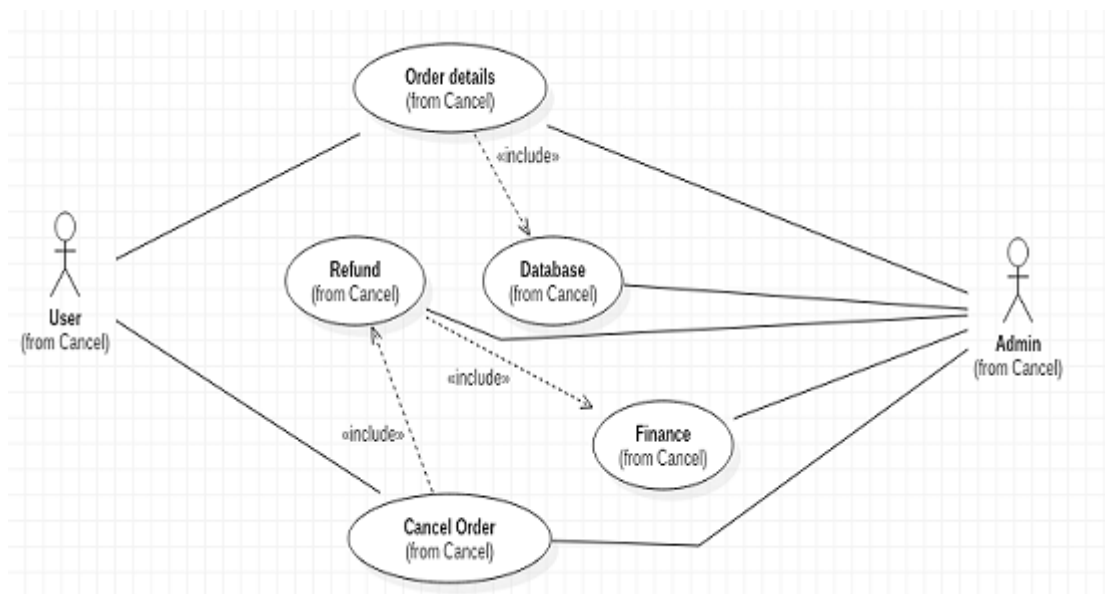
## Payment Portal:



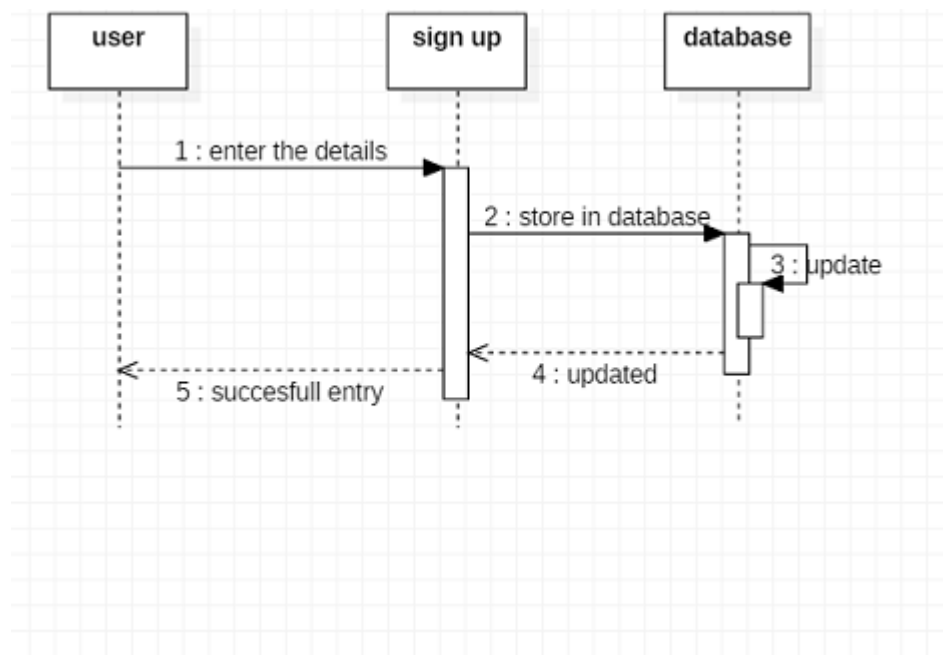
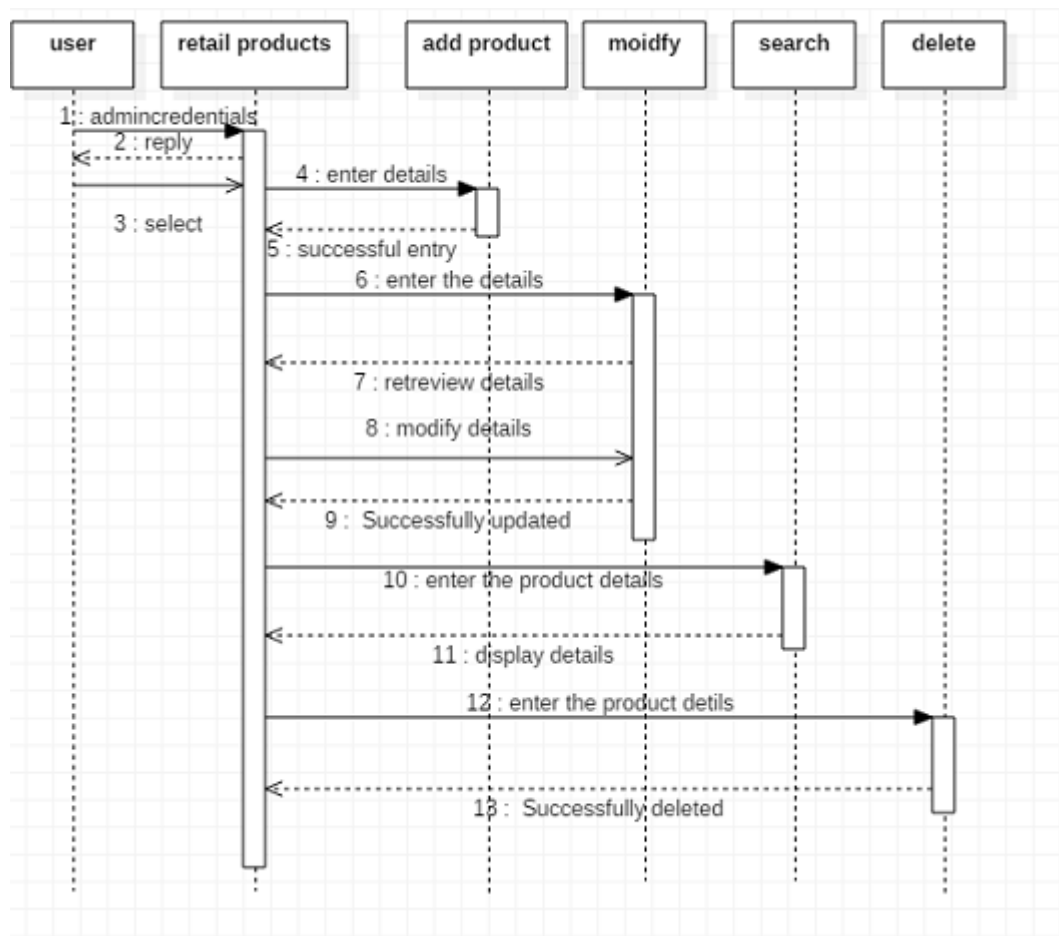
## Invoice Generation:

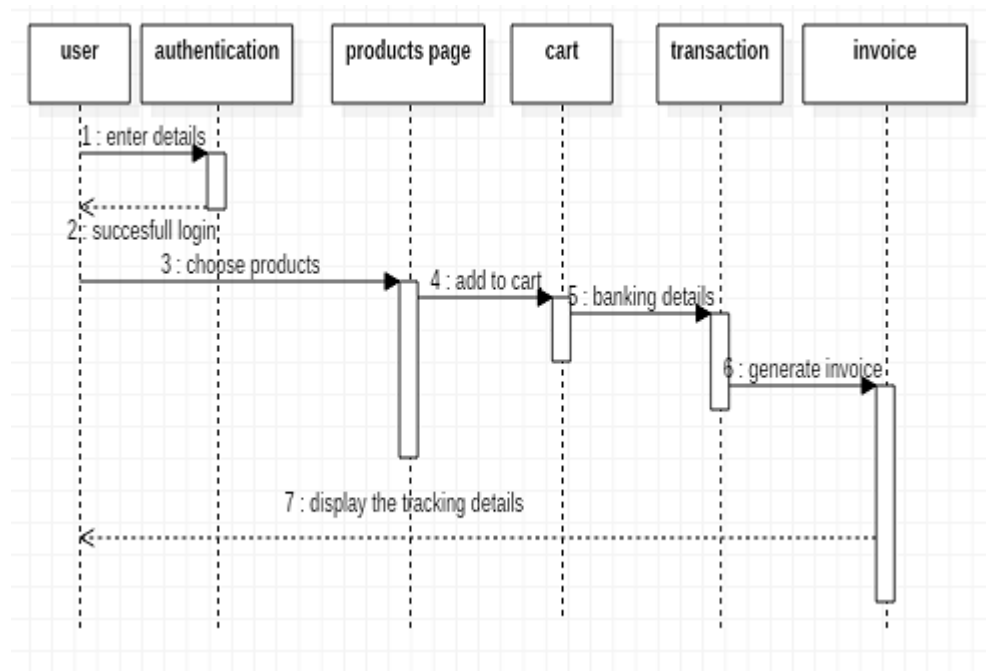


## Cancel:



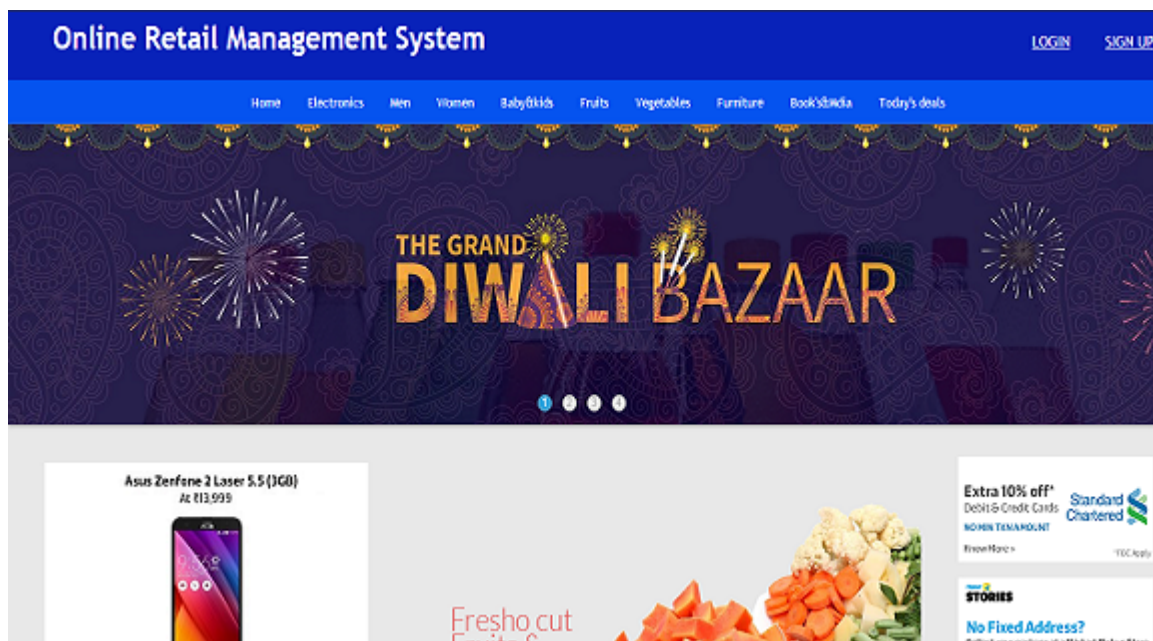
# SEQUENCE DIAGRAMS





## OUTCOME

## HOMEPAGE



## SIGN IN

Online Retail Management System

Username:

Password:

## ORDERING

Online Retail Management System

Logout

Welcome harsh!

Beverages

Product Name	Cost	Description	Wanna Buy??	Quantity
Sprite	25/-	Cold Drink, 200ml Tin	<input type="checkbox"/>	<input type="text"/>
Coke	25/-	CocaCola, 200ml Tin	<input type="checkbox"/>	<input type="text"/>
Fanta	25/-	Cold drink, CocaCola, 200ml Tin	<input type="checkbox"/>	<input type="text"/>
Coke	75/-	CocaCola, 2liter Bottle	<input type="checkbox"/>	<input type="text"/>
Red Bull	100/-	energy drink	<input type="checkbox"/>	<input type="text"/>

Snacks

Product Name	Cost	Description	Wanna Buy??	Quantity
kurkure	10/-	Fried And Spicy	<input type="checkbox"/>	<input type="text"/>



## RECOMMENDATIONS

Pepe Jeans(Original)	1500/-	Jeans	<input type="checkbox"/>	<input type="text"/>
HikeAir Shoes225D	8799/-	Premium Shoe brand. White and yellow in color	<input type="checkbox"/>	<input type="text"/>
gold drop	1300/-	sunflower oil	<input type="checkbox"/>	<input type="text"/>
Iphone 5S	25000/-	smart phone	<input type="checkbox"/>	<input type="text"/>
Recommendations Based on Most Ordered				
Product Name	Cost	Description	Wanna Buy??	Quantity
Coke	25/-	CocaCola, 200ml Tin	<input type="checkbox"/>	<input type="text"/>
Coke	75/-	CocaCola, 2liter Bottle	<input type="checkbox"/>	<input type="text"/>
Recommendations Based on Product Mapping				
Product Name	Cost	Description	Wanna Buy??	Quantity
Lays blue	18/-	Potato Chips, Spicy	<input type="checkbox"/>	<input type="text"/>

## SPM Techniques in ORMS

### Recommendations:

The goal of a Recommender System is to generate meaningful recommendations to a collection of users for items or products that might interest them. The design of such recommendation engines depends on the domain and the particular characteristics of the data available. Such a data source records the quality of interactions between users and items. Additionally, the system may have access to user-specific and item-specific profile attributes such as demographics and product descriptions respectively. Recommender systems differ in the way they analyze these data sources to develop notions of affinity between users and items which can be used to identify well-matched pairs. The myriad approaches to Recommender Systems can be broadly categorized as

1. Collaborative Filtering (CF): In CF systems a user is recommended items based on the past ratings of all users collectively.
2. Content-based recommending: These approaches recommend items that are similar in content to items the user has liked in the past, or matched to attributes of the user.
3. Hybrid approaches: These methods combine both collaborative and content based approaches.

## Collaborative Filtering:

Collaborative filtering is most extensively used approach to design recommender system. Collaborative Filtering (CF) methods play an significant role in the recommendation process, although Collaborative filtering is often used along with other filtering techniques like content-based, knowledge based. Basically Collaborative filtering methods are established on gathering and examining a large amount of information which based on users demeanor, activities or preferences and anticipating taste of that particular user by using their similarity with other users. It does not depend on machine decomposable message and thus it is correctly recommending composite items and because of that it is a key benefit of the collaborative filtering approach. In collaborative filtering recommendation system recommended objects are selected on the basis of past evaluations of a large group of users.

### Advantages:

1. Memory-Based Collaborative filtering techniques makes implementation of recommendation system easier.
2. Using Memory-Based Collaborative filtering techniques one can add new data easily and in incremental manner.
3. Model-Based Collaborative filtering techniques improves prediction performance.

### Disadvantages:

1. Cold Start: CF systems often require a huge amount of existing data on which user can make exact recommendations.
2. Scalability: CF makes recommendations for various environments where billions of users and products exist. Therefore, a huge amount of computation power is often essential to compute recommendations.
3. Sparsity: On major e-commerce site the number of items sold are enormously large. Because of that only a small subset of the entire database is rated by most active users. Hence very few ratings are given to the most popular items.

## Content Based Filtering:

Content-based filtering (CBF) tries to recommend items to the active user based on similarity count which is rated by that user positively in the past.

Advantages:

1. Content-based recommender system provide user independence through exclusive ratings which are used by the active user to build their own profile.
2. Content-based recommender system provide Transparency to their active user by giving explanation how recommender system works.
3. Content-based recommenders system are adequate to recommend items not yet placed by any user. This will be advantageous for new user.

Disadvantages:

1. It is a difficult task to generate the attributes for items in certain areas.
2. CBF advocate the same types of items because of that it suffers from an overspecialization problem.
3. It is harder to acquire feedback from users in CBF because users do not typically rank the items (as in CF) and therefore, it is not possible to determine whether the recommendation is correct.

## Conclusion:

Classic Collaborative Filtering was proven to be the best performing algorithm. The smart non-personalized approach performed surprising well. Item-based Filtering displayed disappointing behavior. Collaborative Filtering algorithm focuses on similar users as judged by their ratings, collecting them in user neighborhoods, whereas Item-based Filtering focuses on similar items as judged by user ratings, collecting them in item neighborhoods.