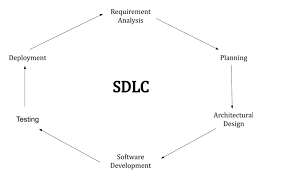
|  |  |
| --- | --- |
| **A picture of a winding road and trees**  **SOFTWARE DEVELOPMENT LIFE CYCLE (Online Shopping System)** | **M.N** |

# ABSTRACT

The Online Shopping system enables vendors to set up online shops, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. Also, on the agenda is designing an online shopping site to manage the items in the shop and also help customers purchase them online without having to visit the shop physically. Our online shopping mall will use the internet as the sole method for selling goods to its consumers. Shopping will be highly personalized and the mall will provide lower prices than most competitors.



# PHASE 1: REQUIREMENT ANALYSIS

This is a small-scale project for Online shopping System. The basic idea is that the candidates can buy products from anywhere during any time by using their card number and password provided to them. The database will maintain the product details information. Customers can view their product details using the card details. This Online shopping system involves two types of users:

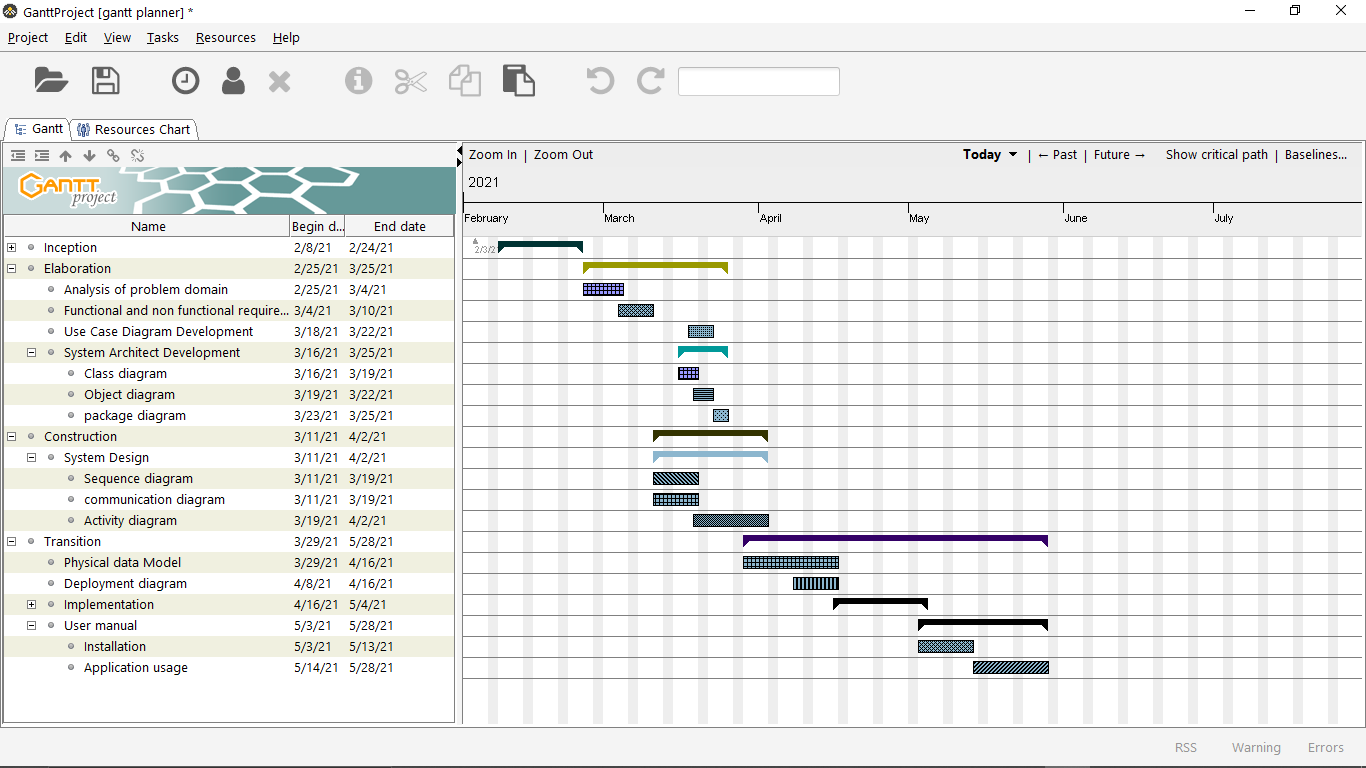
CUSTOMER ROLE: The customers can login/logout the System. His/her can view his/her product details and buy their product. The customer can just view the information whereas he/she could not make changes in the database.

ADMINISTRATOR ROLE: The administrator plays a vital role in the Online shopping system. The administrator controls the entire database. The report of the product is generated by the administrator itself. The main role of the administrator is to safeguard the database and can add/delete the products from the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tools | Version | Editor | Description | Size |
| WampServer | 2.5 |  | Application and database server | 138ko |
| Gantt Project | 2.0.10 | Free software foundation | To plan our Project | 9.22 Mo |
| Visual Paradigm | 12.1 |  | Modelling software | 2.10kb |
| Microsoft office | 2019 | Microsoft corporation | For typing our report |  |
| Netbeans | 8.0.2 |  | IDE | 260 Mo |
| Mozilla Firefox | 40 | Google | Web browser | 8687ko |

# PHASE 2: PLANNING

Project plan with Gantt project



# PHASE 3: MODELING THE REQUIREMENTS

## 3.1. Module Description

### 3.1.1. Login

It is the login session for the Administrator, customer and Exit.

### 3.1.2. Product Details

It is used to view the product details from the database.

### 3.1.3. Add New Products

It is used to add a product into the database and it includes searching for product details and removing a product from the database.

### 3.1.4. Updating the Quantity

It is used to update the product details from the database

## 3.2. UML Diagram

## 3.2. Use case Diagram

Use Case diagrams show the various activities the users can perform on the system. The System is something that performs a function. They model the dynamic aspects of the system. It provides a user’s perspective of the system.

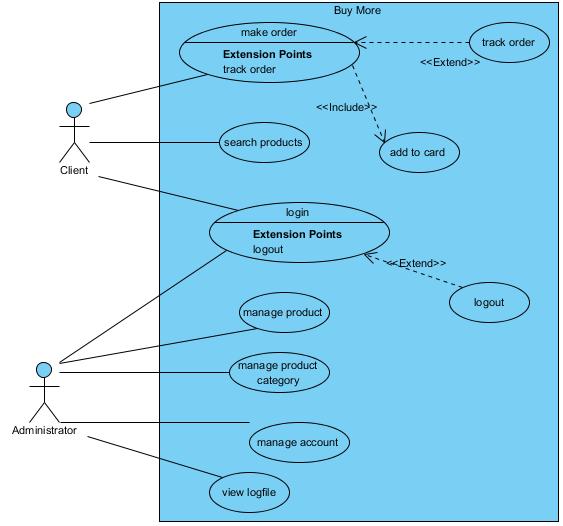
Actor:

An actor is a user of the system playing a particular role.

Use case:

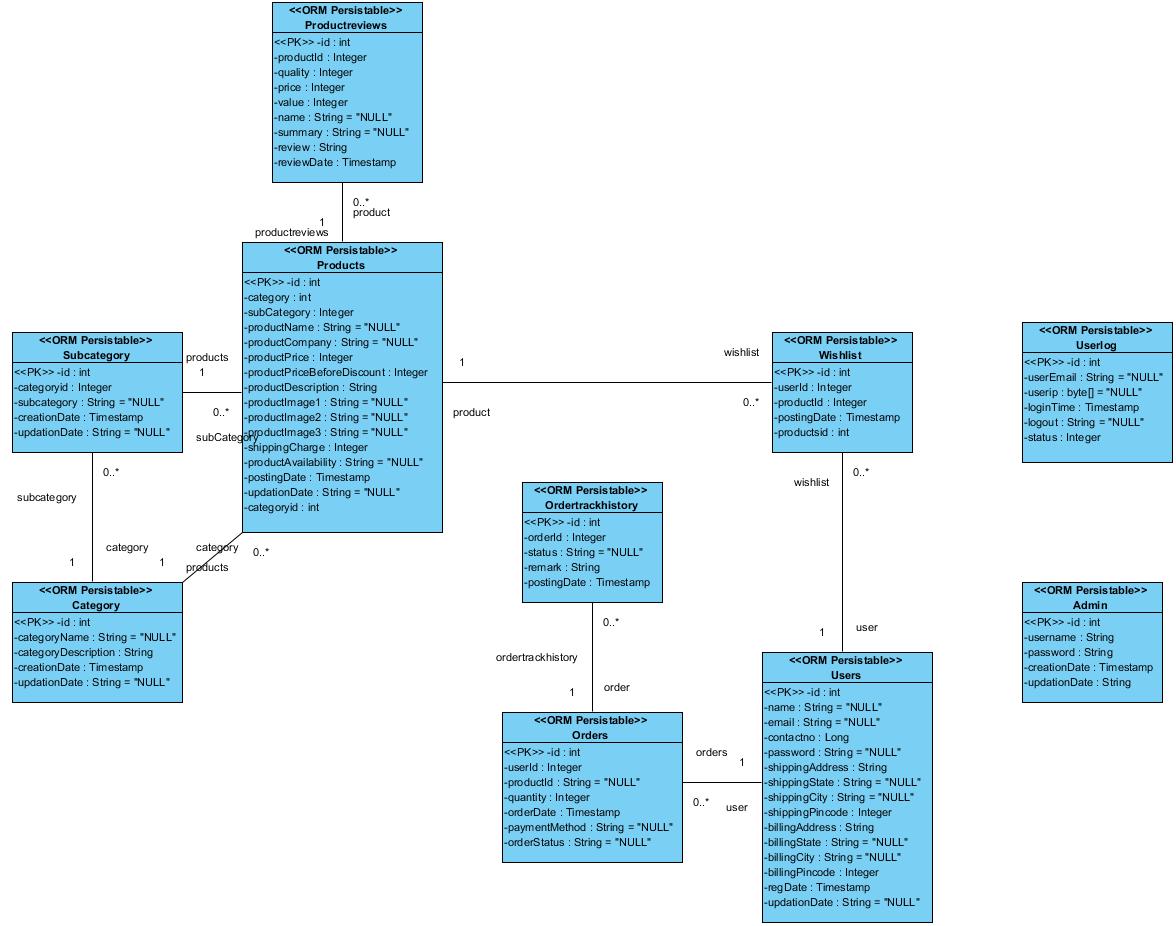
Use case is a particular activity a user can do on the system. Relationship:

Relationships are simply illustrated with a line connecting actors to use cases.



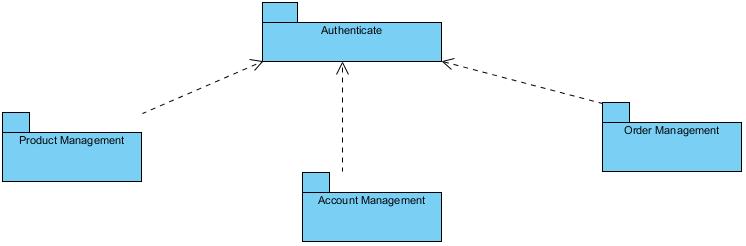
## 3.2. Class Diagram

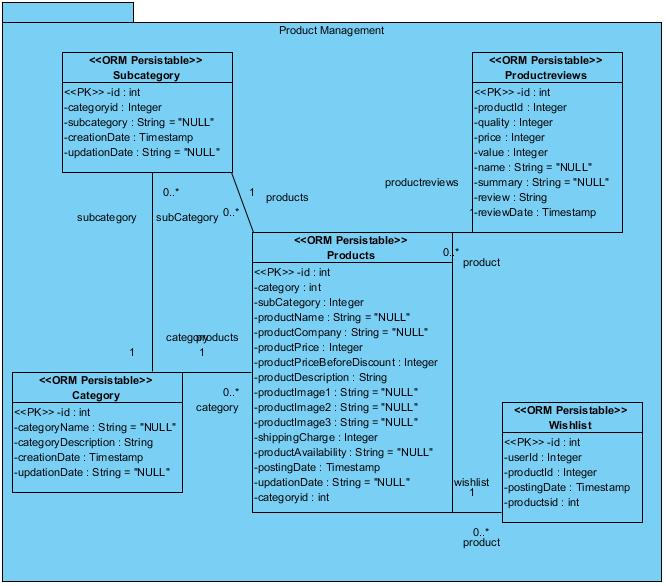
A class diagram shows the building blocks of any object-orientated system. Class diagrams portray a static view of the model, or part of the model, describing what attributes and behavior it has rather than detailing the methods for achieving operations. Class diagrams are most useful in illustrating relationships between classes and interfaces. Generalizations, aggregations, and associations are all valuable in reflecting inheritance, composition or usage, and connections respectively.



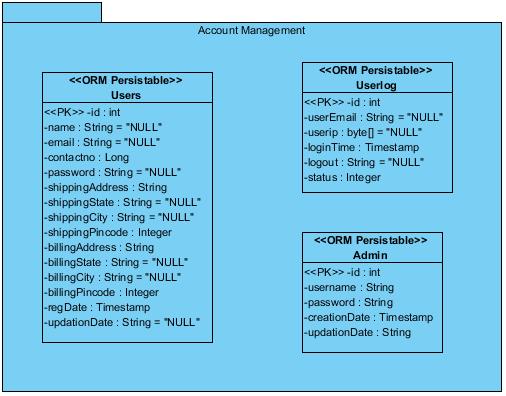
## 3.2. Package Diagram

**Package diagram** is UML structure **diagram** which shows structure of the designed system at the level of **packages**. The following elements are typically drawn in a **package diagram**: **package**, packageable element, dependency, element import, **package** import, **package** merge.





Package <<Product Management >>



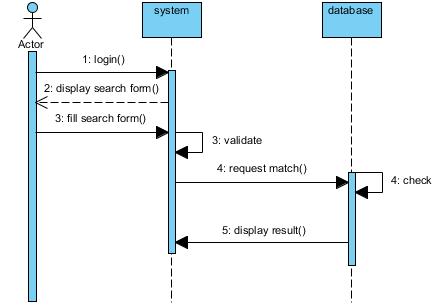
Package <<Account Management>>

## 3.2. Sequence Diagram

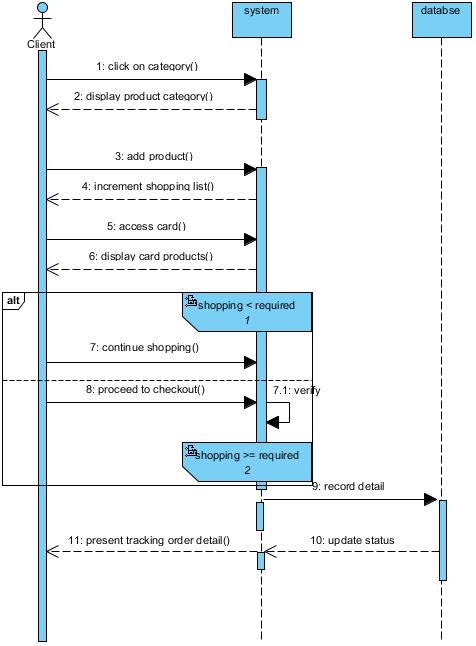
A sequence diagram in Unified Modeling Language (UML) is a kind of

interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

Sequence diagram <<Search product>>



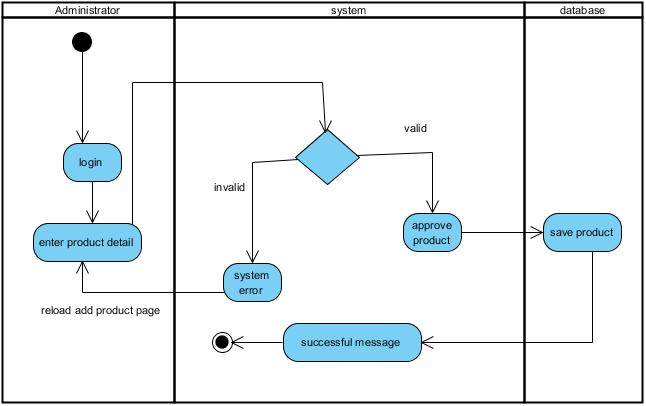
Sequence diagram <<add product>>



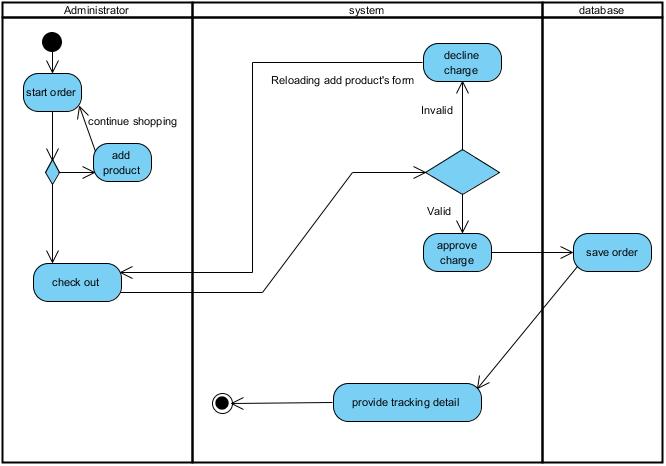
## 3.2. Activity Diagram

In UML, an activity diagram is used to display the sequence of activities. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities.

Activity diagram <<add product>>

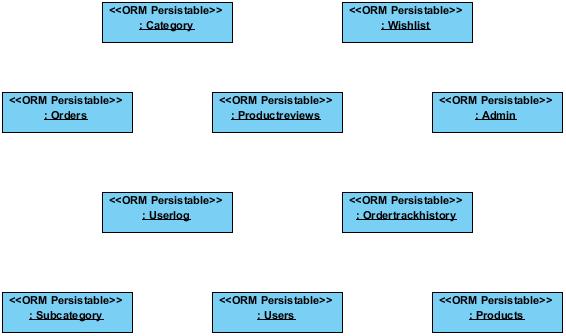


Activity diagram <<make order>>



## 3.2. Object Diagram

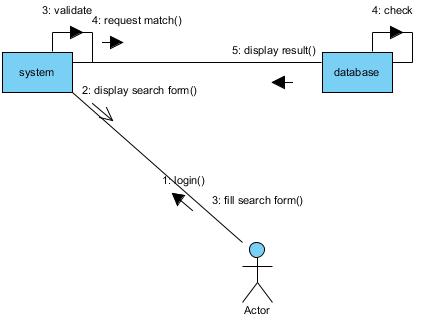
objects in a system and representing the associations between the objects as links. The interaction between the objects is denoted by arrows. To identify the sequence of invocation of these objects, a number is placed next to each of these arrows.



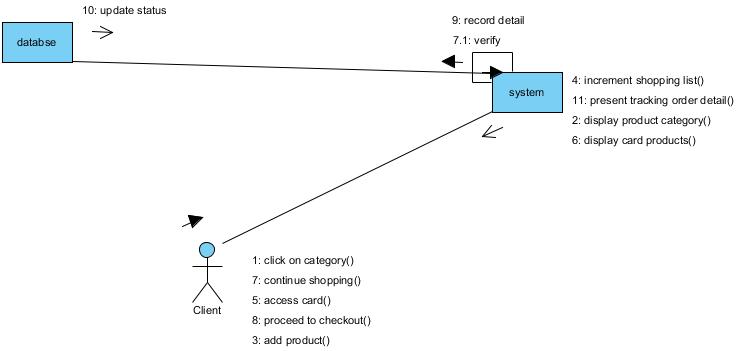
## 3.2. Communication Diagram

Communication diagram formerly called collaboration diagram is an interaction diagram that show similar information to the sequence diagram but it main focus is on object relationship. On communication diagram object are shown with association connectors between them, messages are added to the associations and show as short arrows pointing in the direction of the message flow. The sequence of message is shown through a numbering scheme.

Communication diagram <<search product>>

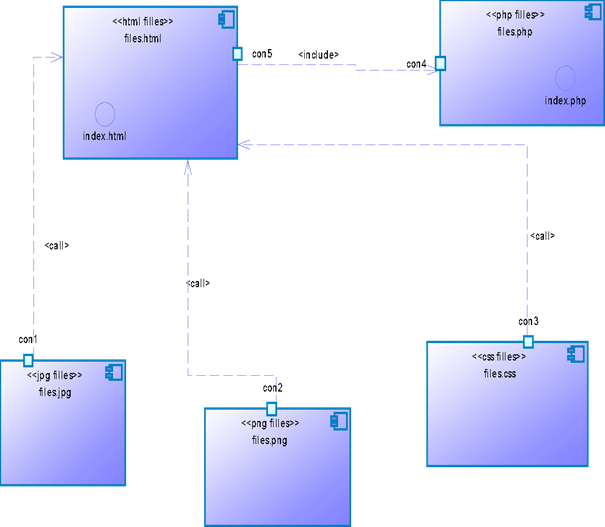


Communication diagram <<add product>>



## 3.2 Component Diagram

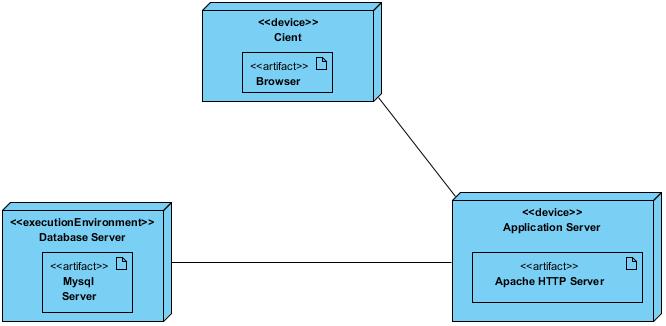
Component diagrams illustrate the pieces of software, embedded controllers, etc., that will make up a system. A component diagram has a higher level of abstraction than a Class Diagram - usually a component is implemented by one or more classes (or objects) at runtime. They are building blocks so a component can eventually encompass a large portion of a system



# PHASE 4: SOFTWARE DEPLOYMENT

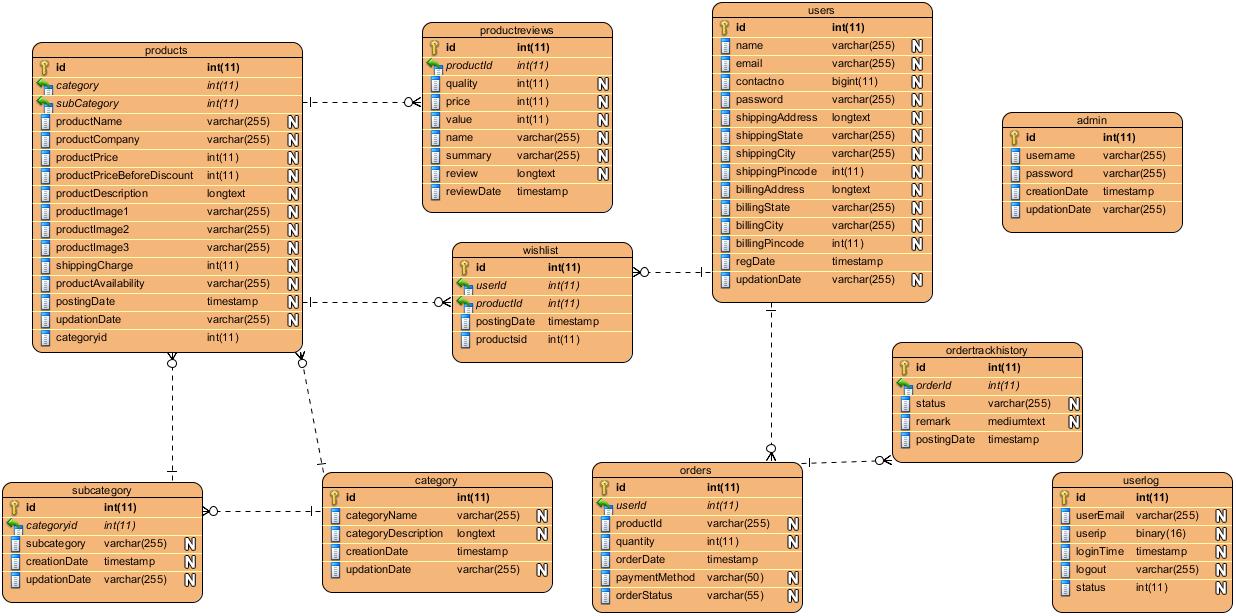
## 4.1. Deployment Diagram

A deployment diagram models the run-time architecture of a system. It shows the configuration of the hardware elements (nodes) and shows how software elements and artifacts are mapped onto those nodes.



## 4.2. Physical Data Model

The physical data model (PDM) presents all the necessary tables in our database. It provides a technical and concrete solution which enable the implementation of the web application. The Physical Data Model specifies the type of Database Management System (DBMS) to be used in managing the database. This model exposes a detail structure of the different database files.



## 4.3. Script

CREATE TABLE `admin` (

`id` int(11) NOT NULL,

`username` varchar(255) NOT NULL,

`password` varchar(255) NOT NULL,

`creationDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`updationDate` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `category` (

`id` int(11) NOT NULL,

`categoryName` varchar(255) DEFAULT NULL,

`categoryDescription` longtext,

`creationDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`updationDate` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `orders` (

`id` int(11) NOT NULL,

`userId` int(11) DEFAULT NULL,

`productId` varchar(255) DEFAULT NULL,

`quantity` int(11) DEFAULT NULL,

`orderDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`paymentMethod` varchar(50) DEFAULT NULL,

`orderStatus` varchar(55) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `ordertrackhistory` (

`id` int(11) NOT NULL,

`orderId` int(11) DEFAULT NULL,

`status` varchar(255) DEFAULT NULL,

`remark` mediumtext,

`postingDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `productreviews` (

`id` int(11) NOT NULL,

`productId` int(11) DEFAULT NULL,

`quality` int(11) DEFAULT NULL,

`price` int(11) DEFAULT NULL,

`value` int(11) DEFAULT NULL,

`name` varchar(255) DEFAULT NULL,

`summary` varchar(255) DEFAULT NULL,

`review` longtext,

`reviewDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `products` (

`id` int(11) NOT NULL,

`category` int(11) NOT NULL,

`subCategory` int(11) DEFAULT NULL,

`productName` varchar(255) DEFAULT NULL,

`productCompany` varchar(255) DEFAULT NULL,

`productPrice` int(11) DEFAULT NULL,

`productPriceBeforeDiscount` int(11) DEFAULT NULL,

`productDescription` longtext,

`productImage1` varchar(255) DEFAULT NULL,

`productImage2` varchar(255) DEFAULT NULL,

`productImage3` varchar(255) DEFAULT NULL,

`shippingCharge` int(11) DEFAULT NULL,

`productAvailability` varchar(255) DEFAULT NULL,

`postingDate` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,

`updationDate` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `subcategory` (

`id` int(11) NOT NULL,

`categoryid` int(11) DEFAULT NULL,

`subcategory` varchar(255) DEFAULT NULL,

`creationDate` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,

`updationDate` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `userlog` (

`id` int(11) NOT NULL,

`userEmail` varchar(255) DEFAULT NULL,

`userip` binary(16) DEFAULT NULL,

`loginTime` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,

`logout` varchar(255) DEFAULT NULL,

`status` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `users` (

`id` int(11) NOT NULL,

`name` varchar(255) DEFAULT NULL,

`email` varchar(255) DEFAULT NULL,

`contactno` bigint(11) DEFAULT NULL,

`password` varchar(255) DEFAULT NULL,

`shippingAddress` longtext,

`shippingState` varchar(255) DEFAULT NULL,

`shippingCity` varchar(255) DEFAULT NULL,

`shippingPincode` int(11) DEFAULT NULL,

`billingAddress` longtext,

`billingState` varchar(255) DEFAULT NULL,

`billingCity` varchar(255) DEFAULT NULL,

`billingPincode` int(11) DEFAULT NULL,

`regDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`updationDate` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `wishlist` (

`id` int(11) NOT NULL,

`userId` int(11) DEFAULT NULL,

`productId` int(11) DEFAULT NULL,

`postingDate` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

# PHASE 5: SOFTWARE TESTING

## 5.1. Preparing Test Plan

Preparing the test plan is the first step in the last phase of the software development cycle .The test plan consists of all the activities that had to be done in the software testing phase. This test plan has been documented using the rational test manager software.

## 5.2. Perform Validation Testing

Software is completely assembled as a package interfacing errors have been uncovered and a final series of software test validation testing may begin. Validation successive when the customer is satisfied.

## 5.3. Validation Test Criteria

Software validation is achieved through a series of black box test that

demonstrates conformity with requirements.

## 5.4. Coverage Analysis

Coverage analysis is used to identify untested code. Using rational pure

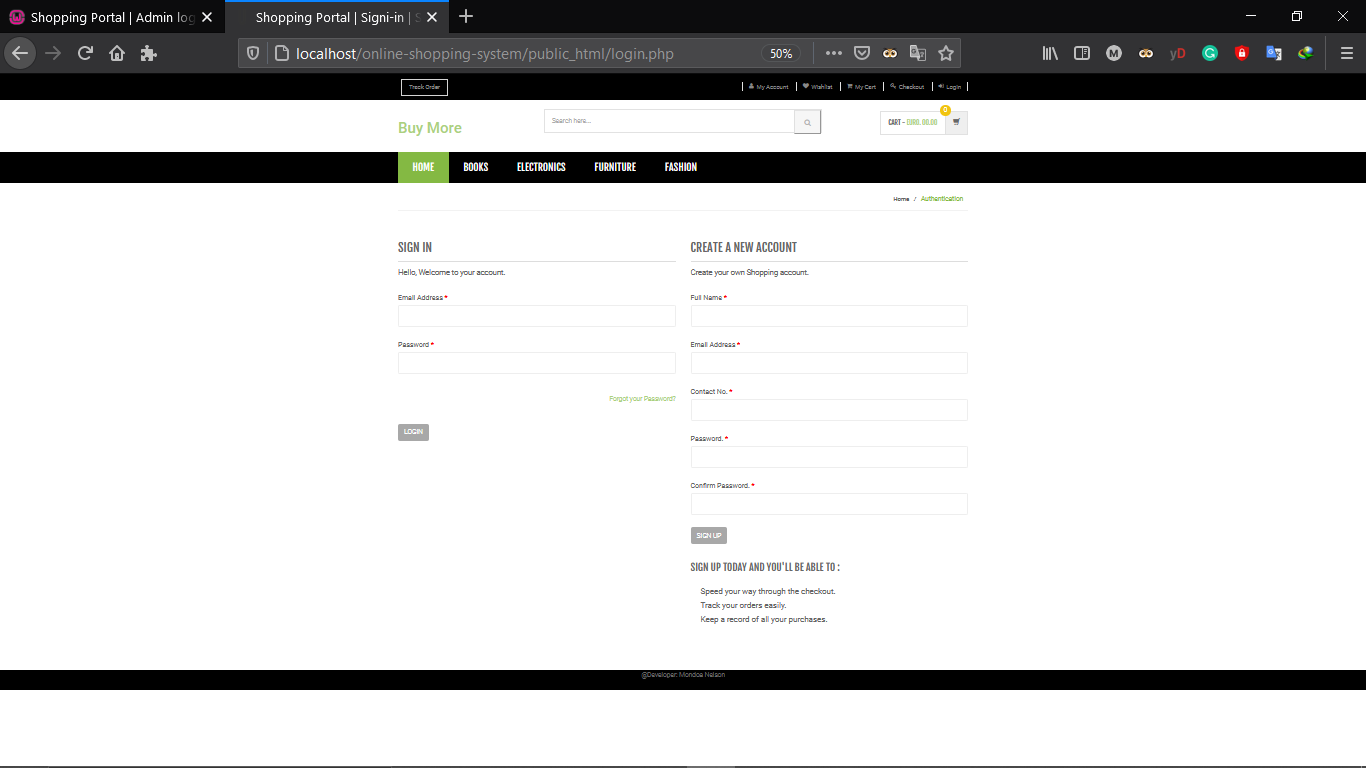
coverage, untested code can easily be identified.

## 5.5. Memory Leaks

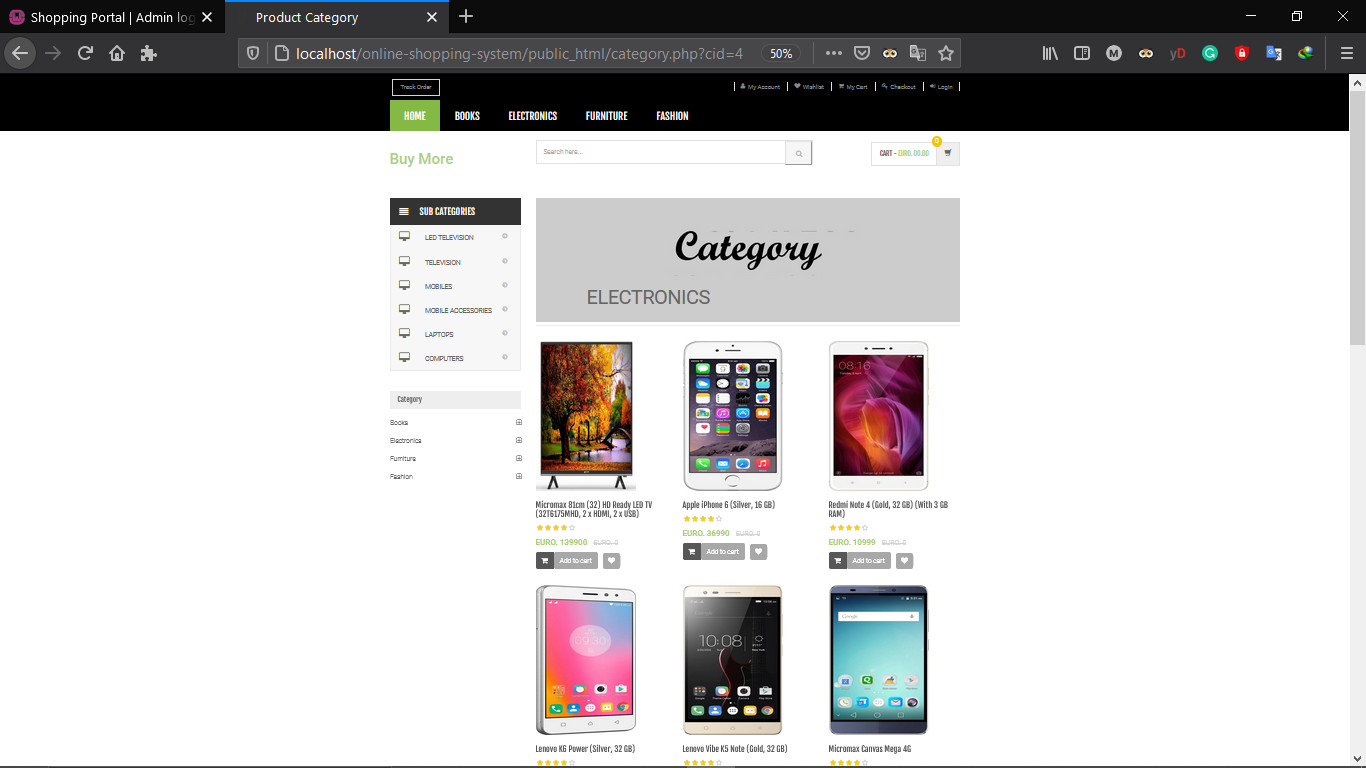
Memory leak testing has been done using rational purity software.

# PHASE 6: USER MANUAL

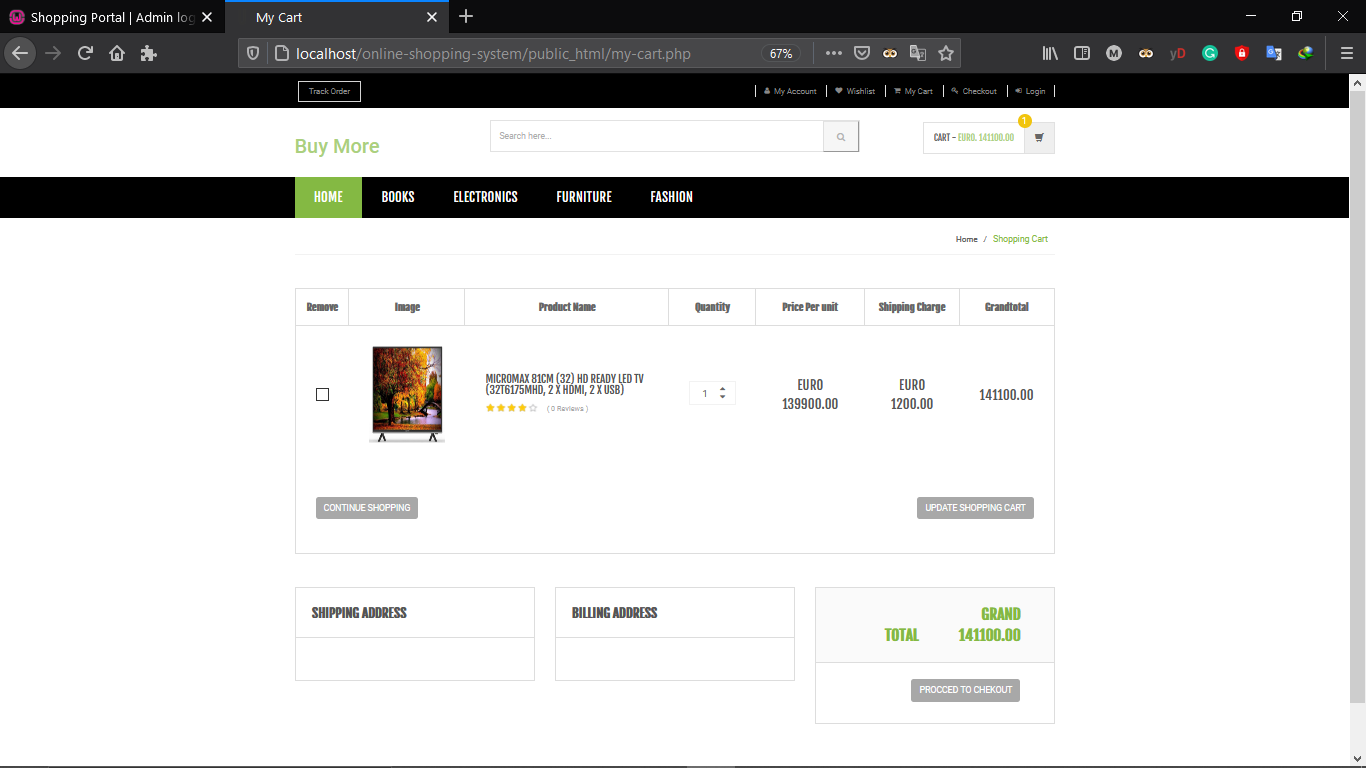
## 6.1. User Account



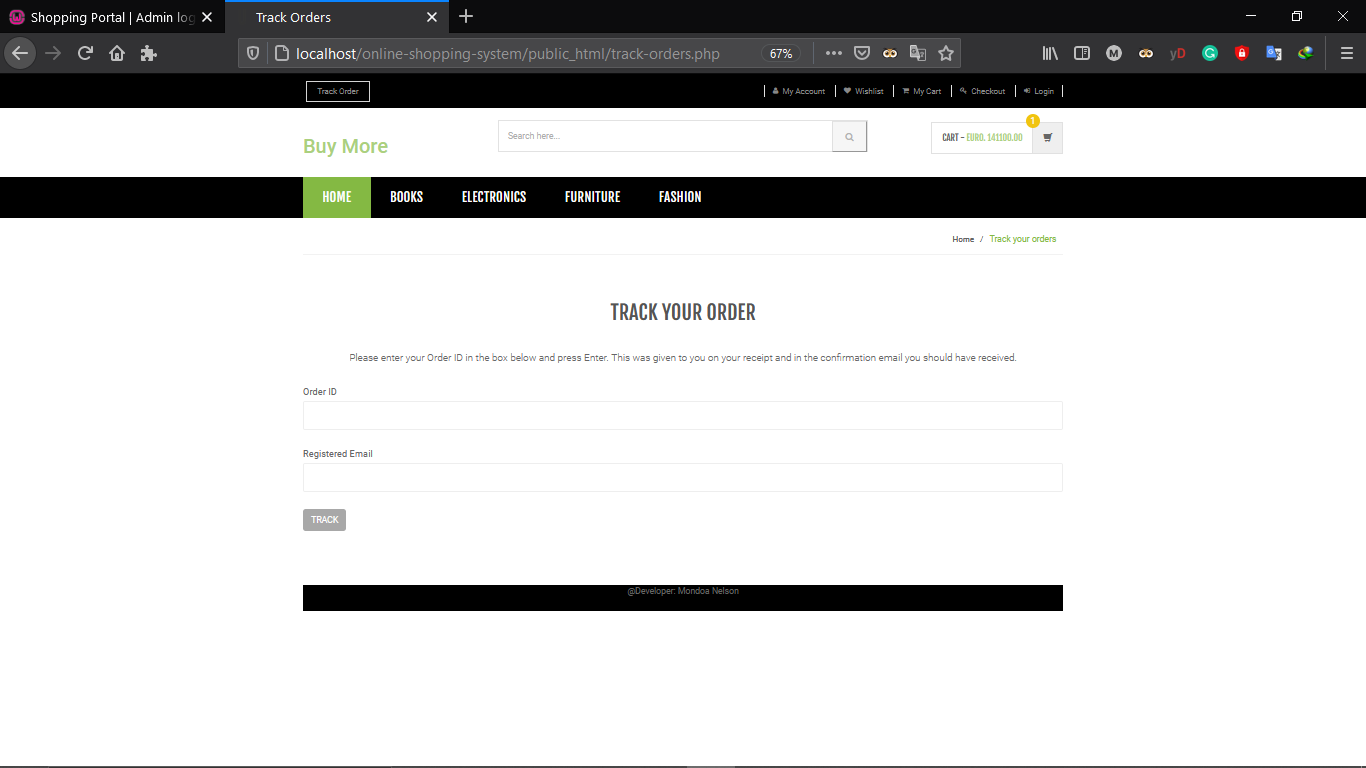
## 6.2. Products



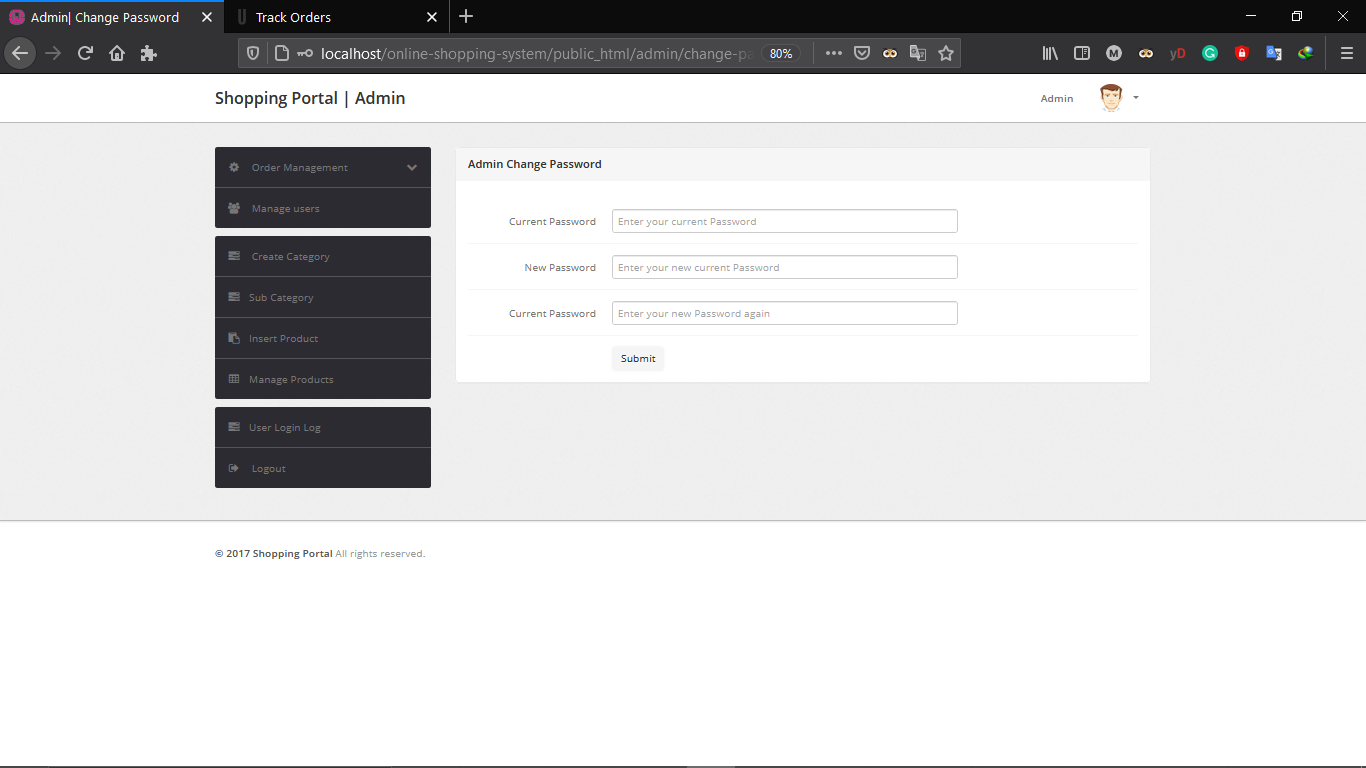
## 6.3. Shopping cart



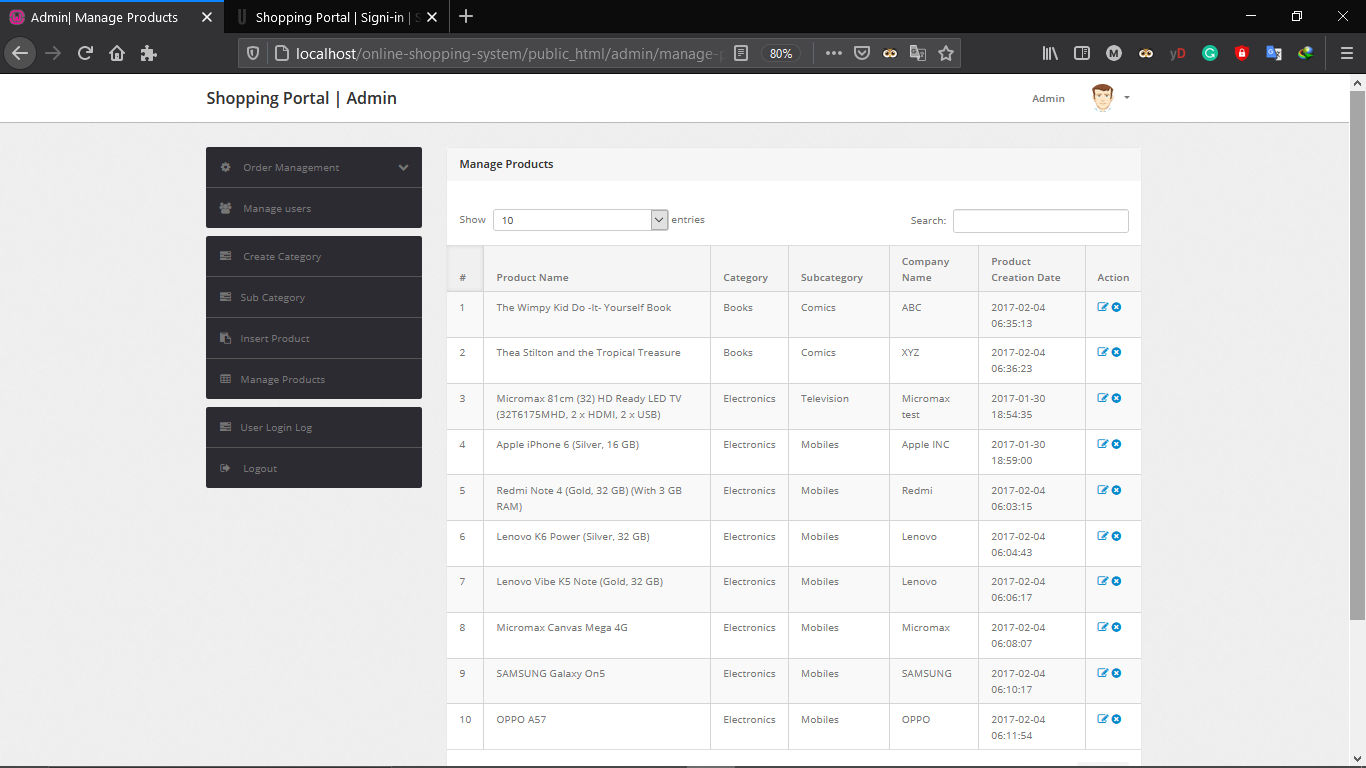
## 6.4. Track order



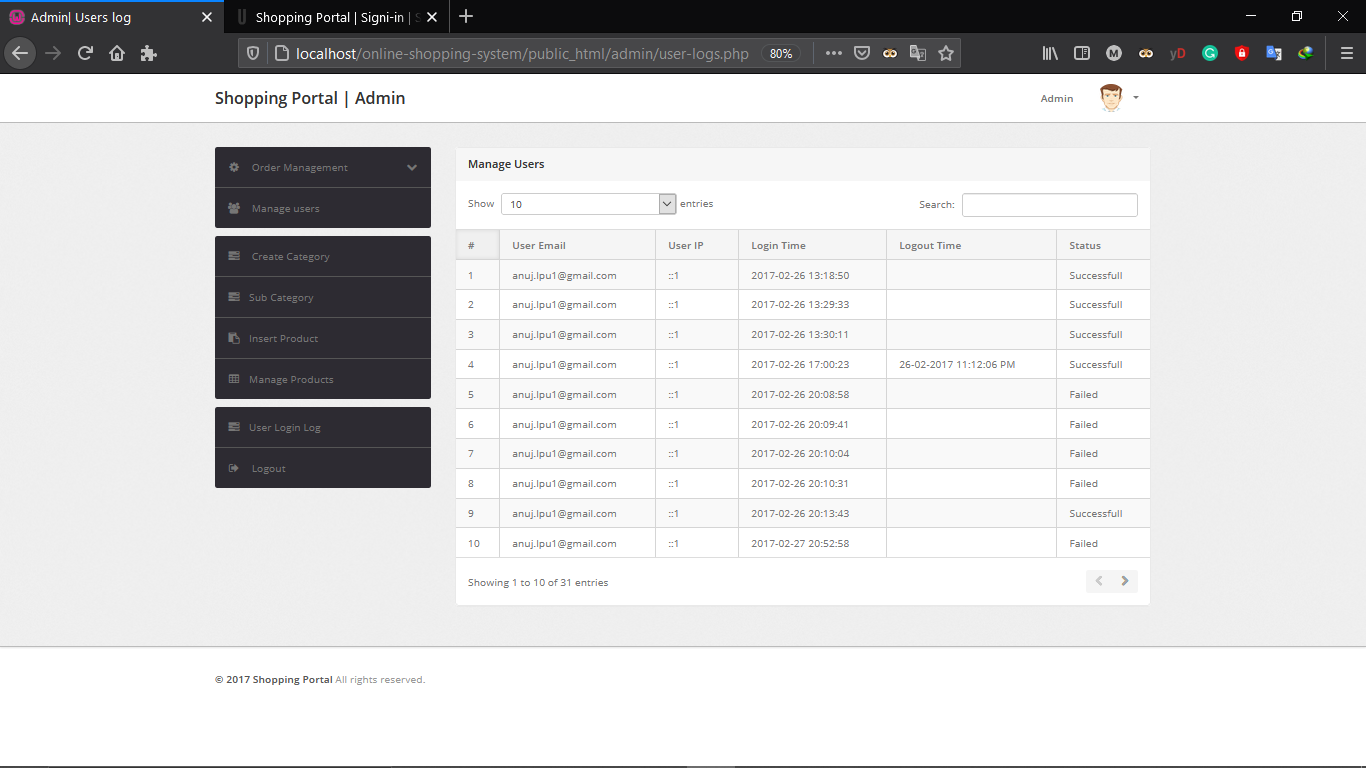
## 6.5. Admin page



## 6.6. Product Management



## 6.7. Log file management



# BIBIOGRAPHY

❖ PHP5 and MYSQL BIBLE by Tim converse and Joyce park with Clark Morgan Wiley publishing, ink, copyright 2004

❖ SQL tutorial pdf by simply easy learning by tutorial point.com

❖ Benning PHP 5.3 by matt Doyle pdf

❖ Relational Database tutorial pdf

❖ Lecturer 6-project Delivery method pdf

❖ PHP and MySQL web development

# WEBOGRAPHY

❖  [**https://www.w3schools.com**](https://www.w3schools.com/): which is the web site that helped us to learn more about PHP, MySQL html, CSS, JavaScript, jQuery and all the web design tool for the programming of the application. Accessed on the 12 May 2021

❖  [**https://www.sql.sh**](https://www.sql.sh/): this website provided us with SQL commands so that we could better manipulate data found in our database and get more knowledge on SQL. Accessed on the 28 May 2021.

❖  [**https://www.visaul-paradigm.com/vp**](https://www.visaul-paradigm.com/vp) **gallery diagrams**: this website helped us to know more about 2.0 and also helped us get all the relationship description of the diagrams. Accessed from the 23th July to the 6 April, 2021.

❖  [**https://www.openclassroom.com**](https://www.openclassroom.com/): this website permitted us to get more knowledge in an object-oriented approach of programming. Accessed on the 3 April 2021.

❖  [**https://www.spark**](https://www.spark/) **system.com au/uml-tutorials.html**: this website permitted us to have more information about UML2.0 and more study on how to draw the diagrams. Accessed on the 18 April 2021.