# Sales & Inventory Management System

# For

# Mahanama Paint Center (PVT) LTD

**G.U.K GAMMANPILA**

**BIT Registration No: R141542**

**Index No: 1415425**

**Supervisor: Mr. Daminda Herath**

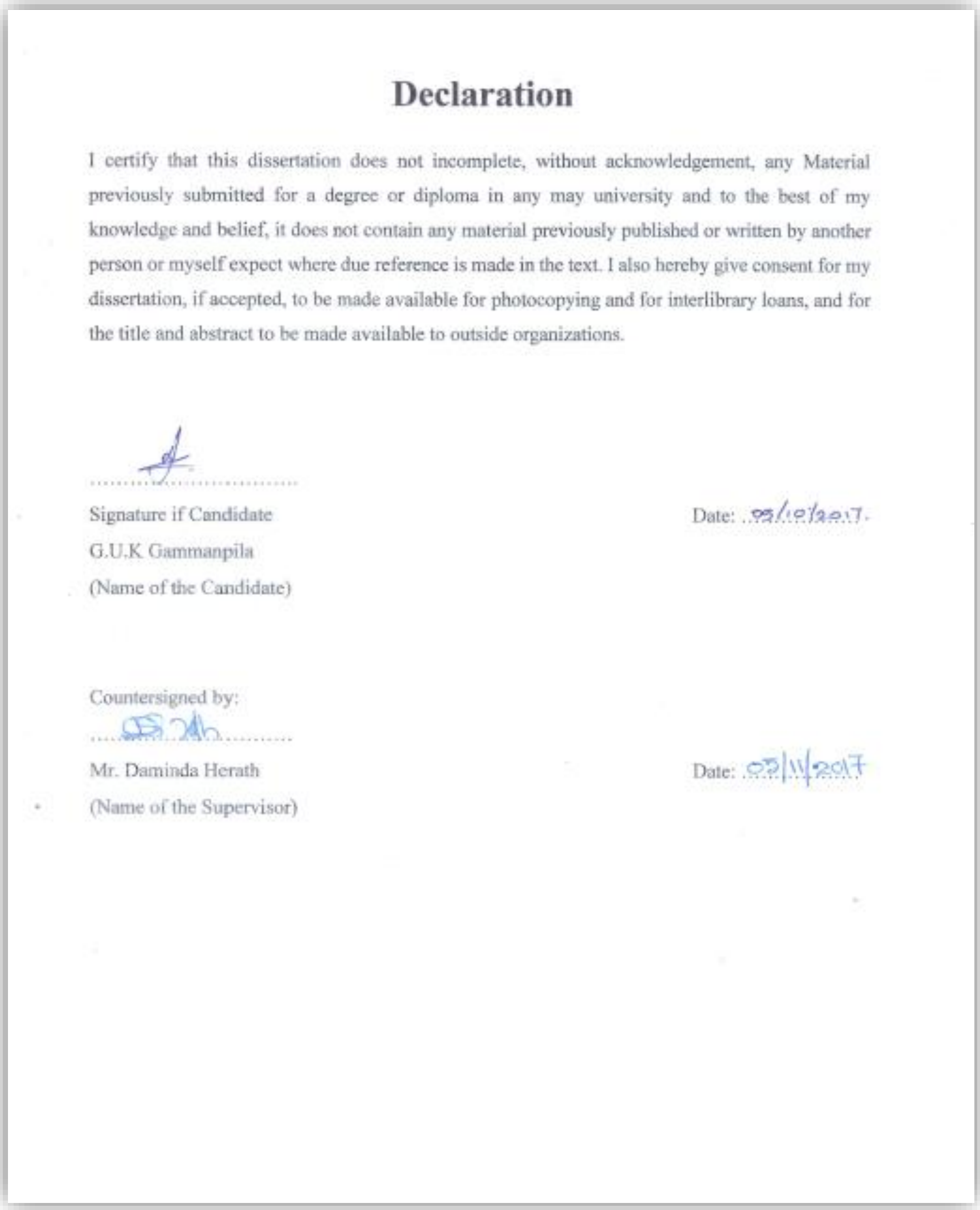
**November / 2017**

**This dissertation is submitted in partial fulfillment of the requirement of the**

**Degree of Bachelor of Information Technology (external) of the**

**University of Colombo School of Computing**

# Declaration



## Declaration

I certify that this dissertation does not incomplete, without acknowledgement, any Material previously submitted for a degree or diploma in any may university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself expect where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

.................................

Signature if Candidate
G.U.K Gammanpila
(Name of the Candidate)

Date: 03/10/2017.

Countersigned by:
.................................
Mr. Daminda Herath
(Name of the Supervisor)

Date: 03/11/2017

# Abstract

Mahanama paint center (PVT) Ltd is developing and well-known Paint reseller shop in kalubowila area. There should be a proper mechanism to manage their day to day activities, which has been handled with partly manual and systematic work currently which will be inefficient time wasting. As a solution we can get an aid from information technology. Sales & inventory management system is we proposed to reduce these problems and will increase the competency and efficiency of their work.

By the proposed web based system, it should be provided many facilities such as product management, customer management, staff management, stock management, supplier management, purchase management, return management, order management and report generation etc.

The system was developed using PHP (PHP hypertext Processor) open source server-side scripting language. MySQL was used as the database tool for designing the database. Apache web server was chosen to run the system. Additionally, Adobe Dreamweaver CS6 was chosen as the coding tool. According to the current Software Engineering theories and by studying the problem domain, the Rational Unified Process (RUP) was identified as the most suitable development methodology based on its flexible developing theory and related advantages in comparison to the other methodologies.

At the end of this development process, the web based system which was tested by many users of the company, was successfully handled over to the client to enhance their company business process as well as to help the paint shop to become a best paint shop in sri lanka.

# Acknowledgement

First of all, I would like to thankful to the BIT Coordinators and academic staff of University of Colombo School of Computing (UCSC) for giving us valuable degree programme and providing us a good guidance contiguously.

Specially thanks must go my project supervisor Mr Daminda Herath for his assistance and guidance.

I would like to thanks Mr. Isuru Mahanama and his staff of Mahanama Paint Center (PVT) LTD for giving me great support and dedicated their valuable time to complete my project.

# Table of Content

# List of Figures

# List of Tables

# List of Acronyms

**AJAX** – Asynchronous JavaScript and XML

**CD ROM** – Comact Disc Read Only Memory

**CSS** – Cascade Style Sheet

**DBMS** – Database Management System

**GB** – Gigabytes

**GHz** – Gigahertz

**HTML** – Hypertext Mark-up Language

**JSON** – JavaScript Object Notation

**MD5** – Message Digest Algorithm

**PDF** – Portable Document Format

**PHP** – Hypertext Pre-processor

**RUP** – Rational Unified Process

**SIMS –** Sales & Inventory Management System

**SQL** – Structured Query Language

**UML** – Unified Modeling Language

**URL** – Uniform Recourses Locator

**WWW** – World Wide Web

**XAMPP** – Windows/Linux Apache MySQL PHP Perl

# Chapter 1 : Introduction

## 1.1 Introduction

Paint is commonly using decorative to houses, building, and vehicles etc. Mahanama Paint Center (PVT) Ltd is a paint service provider in kalubowila area and some other areas.

## 1.2 Motivation of the Project

Currently client's shop is carrying their operations in a half of manual work and half of systematic way with paper work. This manual and systematic will be the very inefficient and very time wasted when provide good service to their customers. The following problems have been occurred in the existing manual and systematic work.

## 1.3 Scope of the Proposed Project

When deciding to develop a system, the scope is a most important fact which we need to consider about. The scope of this proposed project will be as follows.

- System has provided to store the product details, supplier details, customer details, and staff details of the shop.
- Customer Management has facilitated to increase customers loyalty points based on their orders.
- Purchase Management has provided to place purchase orders from suppliers.
- Stock Management has provided to manage stock levels efficiently such as check availability, increase stock levels, mixing stocks, mixing stock usage.
- Staff management can be handled all users of the system.
- System has provided to notification facility to all the low-level stocks.
- Order management can be handled customers' orders efficiently such as place order and view invoice and print.
- Users of each role can be accessed to the system relevant privileges.
- Return management can be handled all the purchase returns and orders returns.
- Report Management daily, monthly reports and other various reports can be generated very easily.
- Transaction management can be handled all the transaction did.
- Payment management can be handled all the payment history.

## 1.4 Objectives of the Project

The main objectives of this project are to,

- Reduced the wastage of time and resources.
- Increase the efficiency of all business operation.
- Centralized the data and enable managers to access information at any time in conventional manner.
- Motivate employees by displaying the progress reports on the working environment.
- Reduce carbon footprint.
- Increase customer satisfaction of shop.

## 1.5 Structure of the Dissertation

Six main chapters with the introduction chapter are comprised in the report. Six main chapters are briefly described as follows:

Requirements gathering technique such as interviews and question arias described in Analysis Chapter. How the current system work what the requirements are for the project (functional requirements and non-functional requirements) are also identified here. UML diagrams such as use case diagrams are drawn and provided to identify un-clear requirements and obtain idea about the system.

According to the analyzed requirements, the database design, user interface design and other design methods are carried out. Simple interface and database design diagram are included in the Design Chapter.

The system specification convert into real world is described in this chapter using different web supported languages. Implementation environment, using technology and reuse modules are discussed in Implementation Chapter.

how to implement system should be tested is described in this chapter. Such as test cases planning, applying the different testing methodologies to test the accuracy of the system. User acceptance testing is also carried out at Evaluation Chapter.

Lesson learnt by implementing this system with a brief description of the shop and future improvements of the systems are discussed in Conclusion Chapter.

Appendix is consisting of seven topics; each topic describes the system details. This section has been written in detail for the interest parties to learn about the system.

References all the URLs references and necessary quotation which helped to write this report are contained in Reference section.

# Chapter 2 : Analysis

## 2.1 Introduction

System analysis is one of the main phases in the software development life cycle (SDLC). System analysts will help to get an overall image of the system and will ale to produce a high-level description of the system through this phase. Main objectives of this phase are what facilities system should provide, required performance of the system. Before analyzing the system, first the requirements should be gathered by using the fact-finding techniques, such as interviews, observation, sample documentations.

## 2.2 Fact Gathering Techniques

Gathering client's requirements by using fact finding techniques are the most critical part of the analysis phase. When gathering the requirements, should be proper way to handle these techniques. There are some fact-finding techniques which can be using to collecting the accurate information's. In this project information's were gathered by using following techniques.

1. Samples and documentations
2. Interviews
3. observation

When we are gathering requirements using samplings and documentation, it would help to understanding about the system. We selected some documents such as daily reports, invoices, paint catalogs to use gather client requirements.

Interviews are most commonly used technique in analysis collects information from individuals face to face. We used both structured way and unstructured way. It was help to aid to verify the facts. We also gathered through these interviews to get feedback and questions ad its grate advantage to this project.

Observation is participating or watches a person perform the activities to learn system.

The existing system was look after to understand the complex areas of the system since the current system is a half manual system.

## 2.3 Analyzing Current Manual System

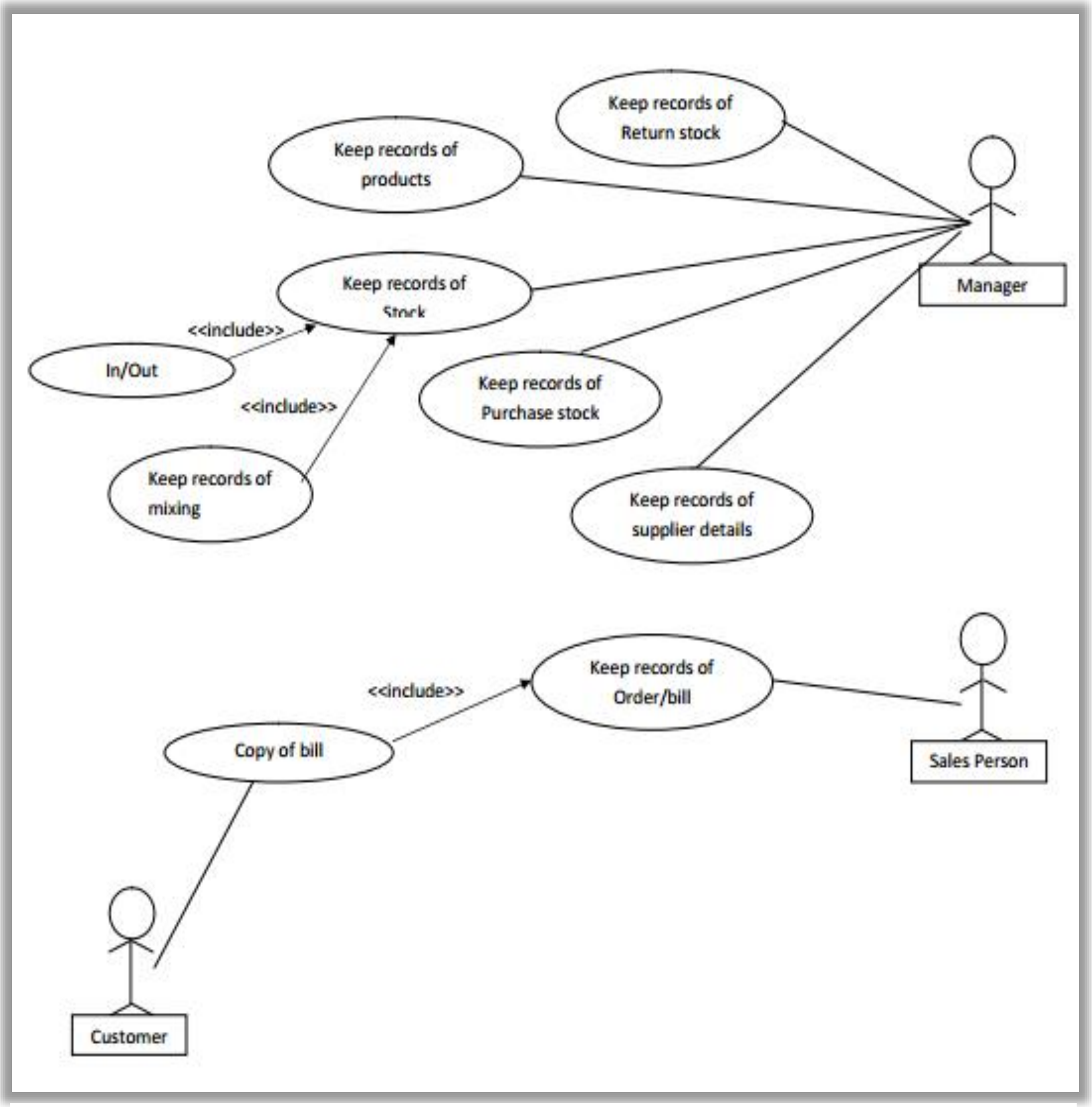


*Figure 2.1: Use Case Diagram for Manual System*

### 2.3.1 Product Management

Mainly shop focus on the products. Currently they have no systematic way to keep details about those products. They keep details about products paper based way. Manual system they keep product name, weight/capacity and category etc.

### 2.3.2 Supplier Management

Mainly they use a manual way to keep details about suppliers who are supply products to shop such as supplier name, supplier address, contact details and brand etc.

### 2.3.3 Purchase Products

When workers inform low products stocks, manager purchase from supplier and keep details current manual system. System has provided d facilities to manage these hire details effectively and has removed the heavy paper based work.

### 2.3.4 Purchase Return

After the purchase products checking all items are properly. Purchase return is based on purchased products. In this proposed system, it has provided a very efficient way to handle these.

### 2.3.5 Manage Stock Details

Stock manage is one of the critical parts of this paint shop. They manage several categories such as mixing management, stock in, stock out and stock availability. Since the currently they use manual system, they have to spend more time to check availability and other things. Proposed system has automated this overall stock management and reduces the time to process.

### 2.3.6 Order Management

Keep records of copy of customer orders bill. This is an only systematic part in the shop. But it is very time-consuming part in current manual system. Proposed system has provided good billing system for shop without any time wasting.

## 2.4 Functional Requirements

What the system is concerned to achieve, is known as functional requirements. Functional requirements can be a calculation, technical details, etc. following are the functional requirements that have been identified through the system analysis [1].

### 2.4.1 Product Management Module

- System should provide facilitate create new product profiles, edit, view, search and delete.
- Facilitate to search using product name, product brand and supplier.
- System should provide facilitate to assign category, sub category and brand based on product.
- Facilitate to create new categories, sub categories and brands, edit, view, search and delete.

### 2.4.2. Staff Management Module

- System should provide staff users can log into the system securely by validating given username and password.
- System should provide facilitate to create new staff members, edit, view and delete their profiles.
- System should provide facilitate to assign user privileges based on the user position.

### 2.4.3. Supplier Management Module

- System should provide facilitate to create new suppliers, edit, view, search and delete.

### 2.4.4. Customer Management Module

- System should provide facilitate to create new loyalty customers.
- Facilitate to edit, view, search and delete loyalty customers.
- System should provide adding points each and every order and buy products using points.

### 2.4.5. Purchase Management Module

- System should provide facilitate to purchase new stock is relevant product.
- System should provide facilitate to send purchase product list via email.
- System should provide to view, search and delete purchase records.

### 2.4.6. Stock Management Module

- System should provide, when considering particular products, it can identify kind of quantity is remaining in the stock and what is the quantity required to purchase.
- System should provide facilitate to store records about mixing product quantities.
- System should provide facilitate to show stock in and stock out details properly.

### 2.4.7. Return Management Module

- System should provide to return purchase product and customer ordered products.
- System should provide to edit, view, search and delete records.

### 2.4.8. Order Management Module

- System should provide to create bill, edit, view bills.
- System should provide to get sum of product prices.

### 2.4.9. Transaction Management Module

- System should provide to show transaction data and all orders details.

### 2.4.10. Payment Management Module

- System should provide to show financial details in shop.

### 2.4.11. Report Management Module

- System has provided facilitate to generate reports in the shop.
- Facilitated to generate daily and monthly reports of sales.
- Facilitated to generate staff reports and financial reports.
- Facilitated to generate reports such as products, supplier, purchase history, returns, stock info.

## 2.5 Non- Functional Requirements

Non-Functional requirements are also important facts which we need to consider when developing the system. Non-Functional requirements describe the system properties and constraints of the system. Those requirements are applied to the whole system, not only for individual parts of the system. If we do not consider about nonfunctional requirements system can be useless [2].

### 2.5.1. Security

System should be achieving the security requirements. Because most of the important personal details, customer details, financial details stored in the system. If that is sensitive information goes wrong person may be loss their competency.

### 2.5.2. Usability

Usability requirements has been achieved by using various techniques such as easy menu navigation, search options, attractive interfaces and great matching colors.

### 2.5.3. Accuracy

Accuracy is a one of another important nonfictional requirement which we need to consider. If we set their accuracy database problems may be occurred, by validation techniques.

## 2.6 Existing Similar Systems

Some of the existing open source systems were studied to get a full idea about the Sales & Inventory Management System functions.

**Right Control Lite** – is designed especially for the small business user. System with customer management, supplier management, sales management, purchase management, return management and other functions. This is available in below reference number [3].

**EREC'S Pharmacy** – is an open source sales & inventory management system for small and medium sized shops. This has an easy to use interface with interactive screens and many features of the sales & inventory management system [4].

# Chapter 3 : Design

## 3.1. Introduction

During software designing, system should have better design early implementation. System design provide information about architecture, interfaces components, modules and data. System designers using different system design approaches. Its include modern structured design, JAT, RAD, prototyping, information engineering and object-oriented design. Object oriented design is widely used approach in modern software design. Project can easily reuse than in the other approaches. Further coding is easier to write and understand less errors [5].

## 3.2. System Development Life Cycle

System development life cycle is process that explained the activities performed at each stage of software development project. It consists of a describe the how to plan, design, implement and test the system. The objective is design good quality software. There are several methodologies such as waterfall, prototype, agile, etc.

When developed the system used Iterative and Incremental Development type. Iterative and Incremental Software Development is a method of development that is modeled around a systematic increase in features addition and cycle release and upgrade pattern. Its begins with planning and continuous through user feedbacks and incremental feature addition with the deployment of completed software at the end of each cycle.

## 3.3. Methodology for the Proposed System

RUP (Rational Unified Process) is selected for proposed system. Because its iterative software development process framework and its suitable for object oriented development. Client does not have clear idea about system requirements properly so requirements may be changed. Further system has divide into modules such as stock management module, customer management module, etc. Each module has developed incrementally and iteratively. Inception, Elaboration, Construction, and Transition are the phases of RUP life cycle. Description given below [6].

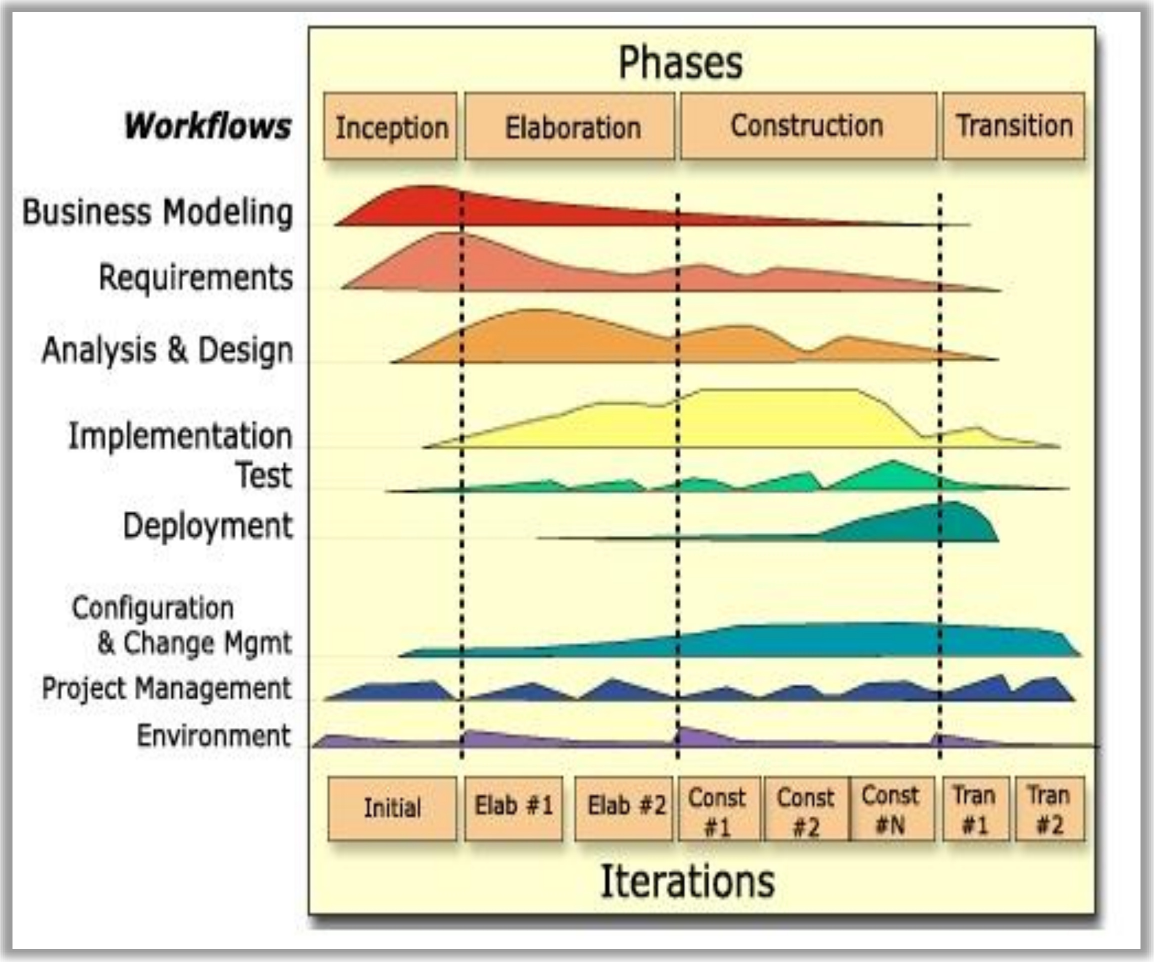


*Figure 3.1: Rational Unified Model [7]*

- **Inception Phase**

  Inception phase is to establish a business modal case for the system. You should identify all the external entities, that will interact with the system and define these interactions. You then use this information to assess the contribution that the system makes to the business.

- **Elaboration Phase**

  The goal of the elaboration phase is to develop an understanding of the problem domain, establish an architectural framework for the system. Develop the project plan, and identify key project risks. On completion of this phase you should have a requirement modal for the system.

- **Construction Phase**

  The construction phase involves system design, programming and testing. Part of the system developed in parallel and integrated during this phase. On completion of this phase, you should have working software system.

- **Transition Phase**

  The final phase of RUP is concern with moving the system from development community of the user community and making it work in a real environment.

## 3.4 Object Oriented Designing

An Object-Oriented Design is concerned with developing an object-oriented modal of the software system to implement the identified all requirements. The main goal is this approach is defining objects and establishing relationship between classes by sending and receiving messages.

Object oriented design is method which is used to collaborate object and attribute and methods of their objects. Unified Modeling Language is widely used designing tool for perform object oriented design. This design approach is very efficient way to implement designing phase [8].

### 3.4.1 High Level Use Case Diagram for Proposed System

We use mainly use diagraming technique is UML diagram, is use case diagram. This helps to identify the correct system requirements from system users and validate system requirements can be use by system analysis. The high-level use case diagram for the proposed system is showing in figure 3.2

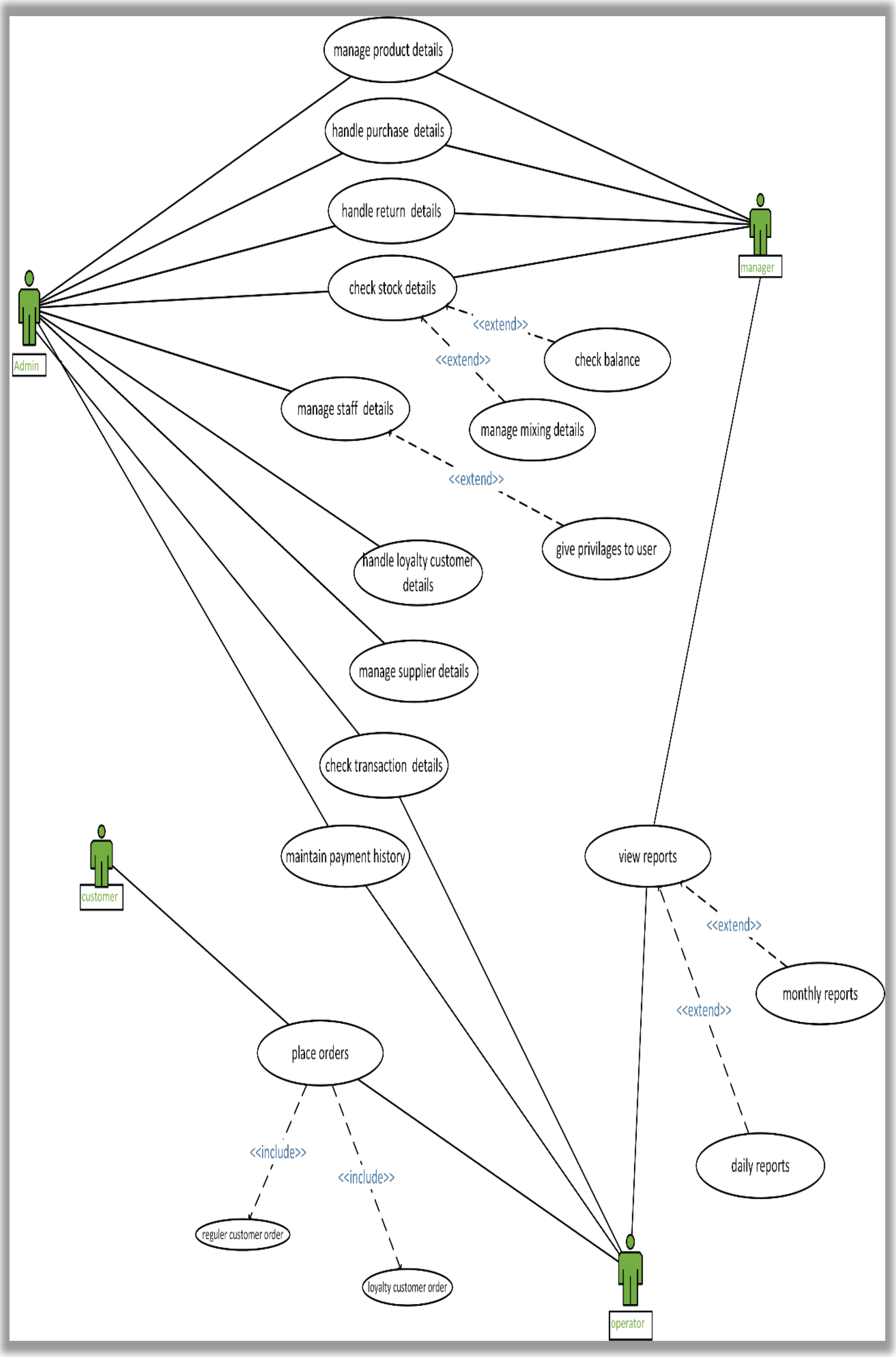


*Figure 3.2: High Level Use Case Diagram*

## 3.4.2 Class Diagram for Proposed System

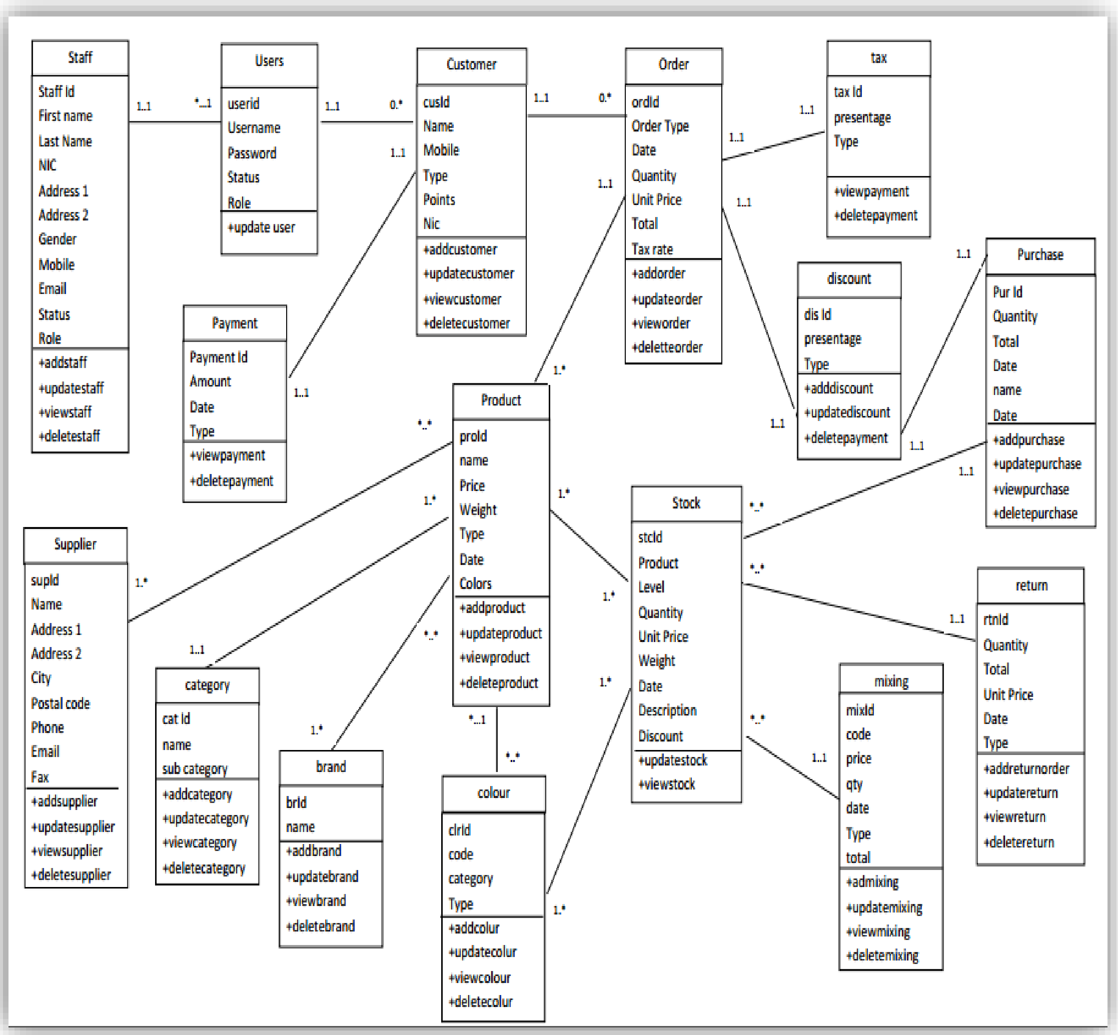


*Figure 3.3: Class Diagram*

“The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system” [8].

Class diagram are used when developing an object-oriented system modal to show the classes in a system and the associations between these classes. The class diagram shows figure 3.3.

## 3.5 Database Design

Database design is a process of producing a detailed model of a database. There are three main phases in the database design conceptual design, logical design and physical design.

The below database is normalized to the third normal form to avoid the redundancy of data. The minimal redundancy allows the performance of insertion, modification and deletion operation of the database without conflicts [9].

Figure 3.4 represent the table structure of the database for proposed system given below.

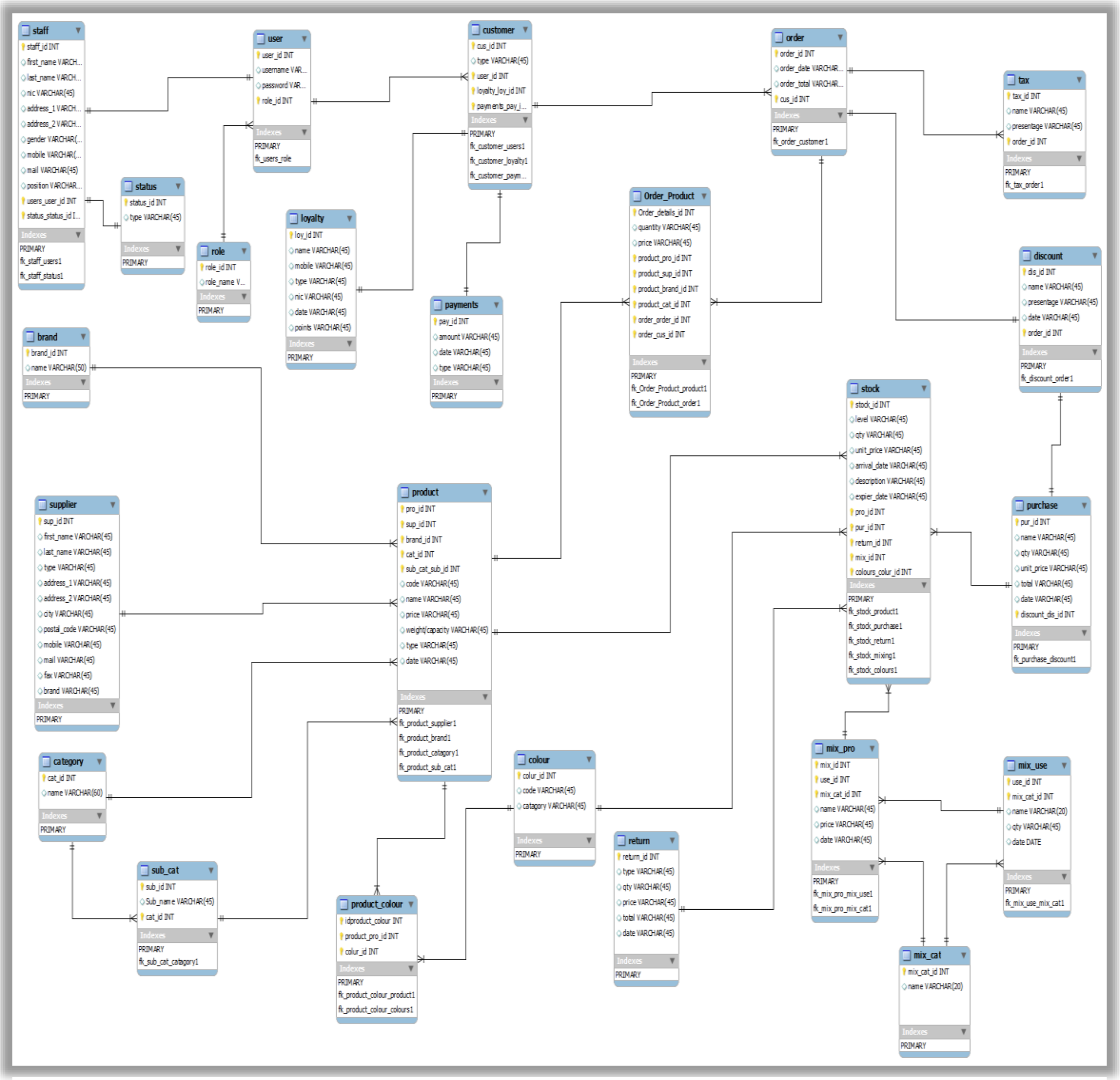


*Figure 3.4: ERD Diagram*

### 3.5.1 Database Normalization

Database Normalization is the process the organizing the attributes and table of a relational database to minimize and redundancy and dependency. Normalization usually involves dividing into large table to smaller tables defining between relationship them [10].

- First Normal Form (1NF)
  Main objective is eliminating the repeating groups and multi value columns and arranges them in single table, and define primary key for identifying each related attribute.

- Second Normal Form (2NF)
  Main objective is eliminating the partial dependencies and creating separate tables and related tables with the foreign keys.

- Third Normal Form (3NF)
  Main objective is eliminating the transitive dependencies.

## 3.6 Interface Design

Among the major component of the system which is user interfaces. Interactively in between system and the user is managed by the interface. User friendliness, integrated color combinations and the well-organized components are dependent on it. Without having user interface, interaction with the system becomes too hard.

Without having a user-friendly interface, interaction with the system becomes hard. Privileges must be set by the System Administrator to different users based on their role. Management of those privileges and presenting them effectively to users is helped by the interfaces.

### 3.6.1 Login Interface

The system login page which belongs to the developed system is displayed by figure 3.5. In any computerized system the first interface encountered by the user is the login page. Therefore, by designing and handling errors properly a pleasant feeling about the rest of the system can be created within the user.

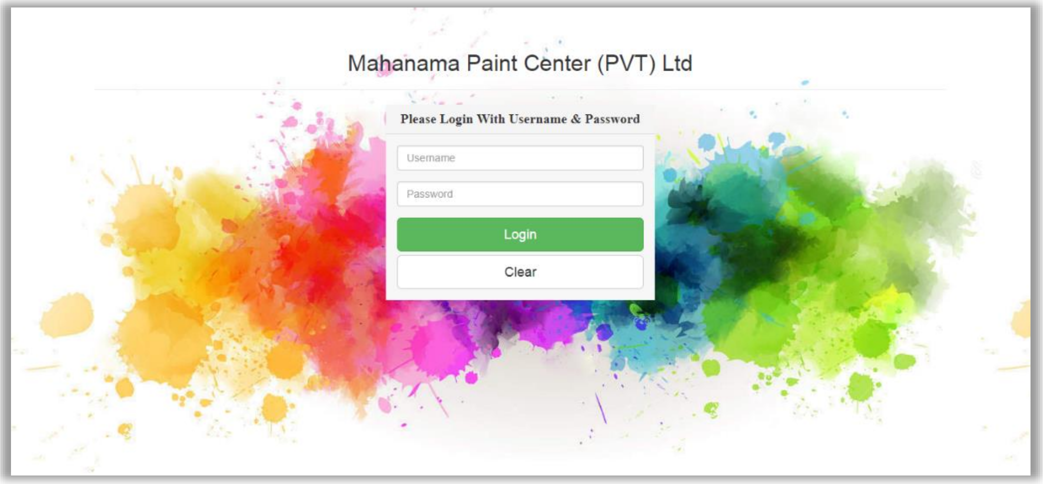
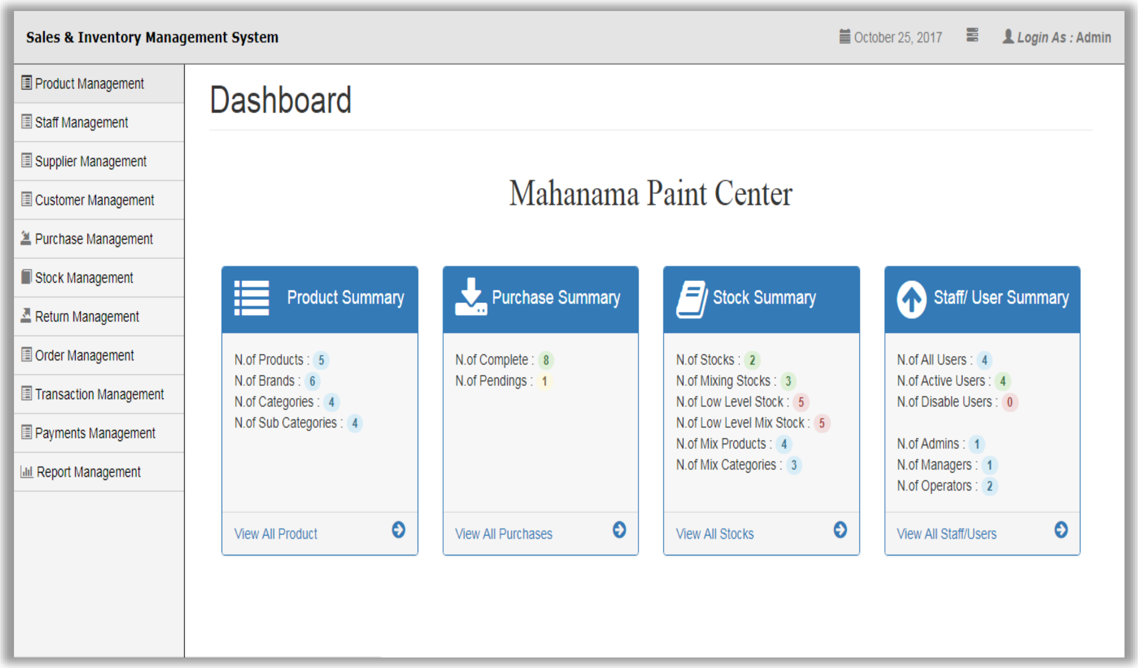

Mahanama Paint Center (PVT) Ltd
Please Login With Username & Password
Username
Password
Login
Clear

*Figure 3.5:Login Page*

## 3.6.2 Administrator Home Page

The home page of administrator's Home page is displayed by figure 3.6. According to the client's requirements as well as studying other similar web based system interfaces, the final output had been created. The information of module data counts, what user login, product summary, purchase summery, stock summary, staff summary and main menus and sub menus and also notifications.



N.of Products : 5
N.of Brands : 6
N.of Categories : 4
N.of Sub Categories : 4

N.of Complete : 8
N.of Pendings : 1

N.of Stocks : 2
N.of Mixing Stocks : 3
N.of Low Level Stock : 5
N.of Low Level Mix Stock : 5
N.of Mix Products : 4
N.of Mix Categories : 3

N.of All Users : 4
N.of Active Users : 4
N.of Disable Users : 0
N.of Admins : 1
N.of Managers : 1
N.of Operators : 2

*Figure 3.6: Administrator Home Page*

### 3.6.3 Product Adding Page

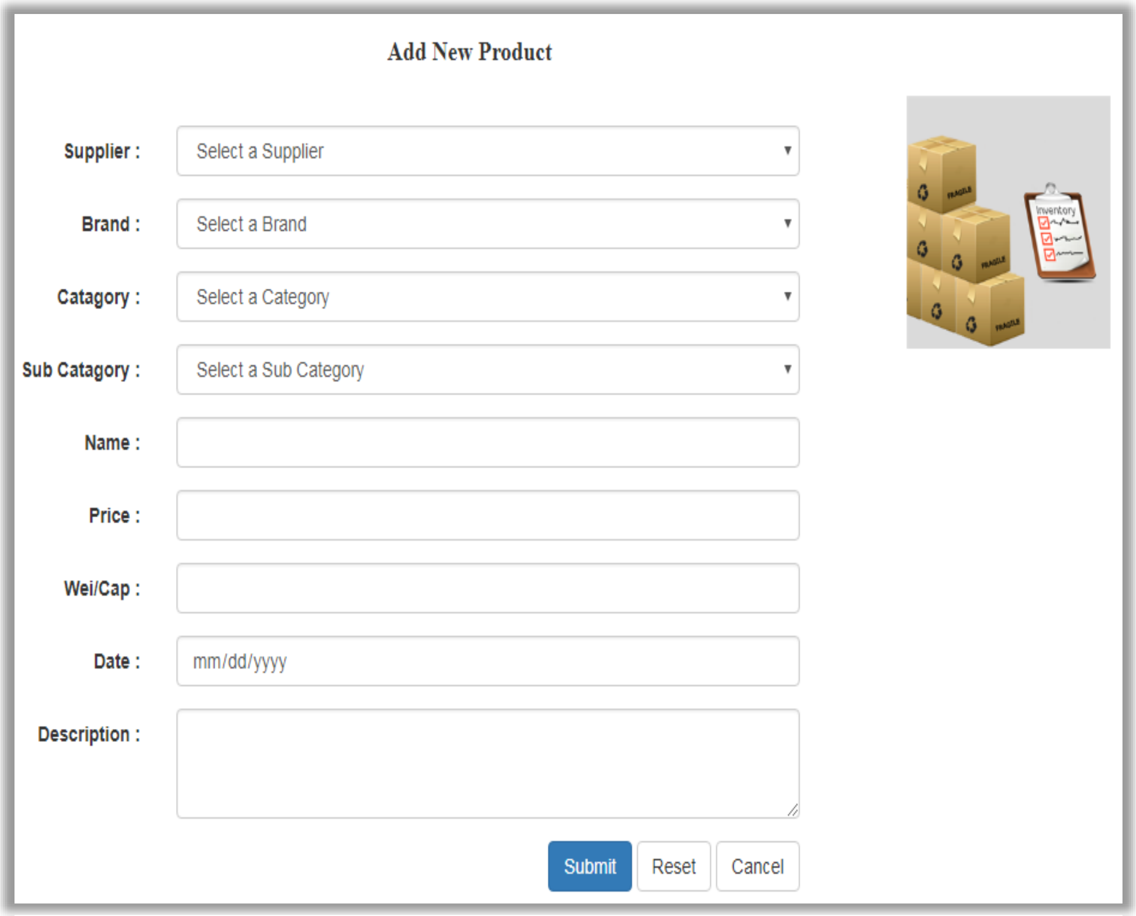
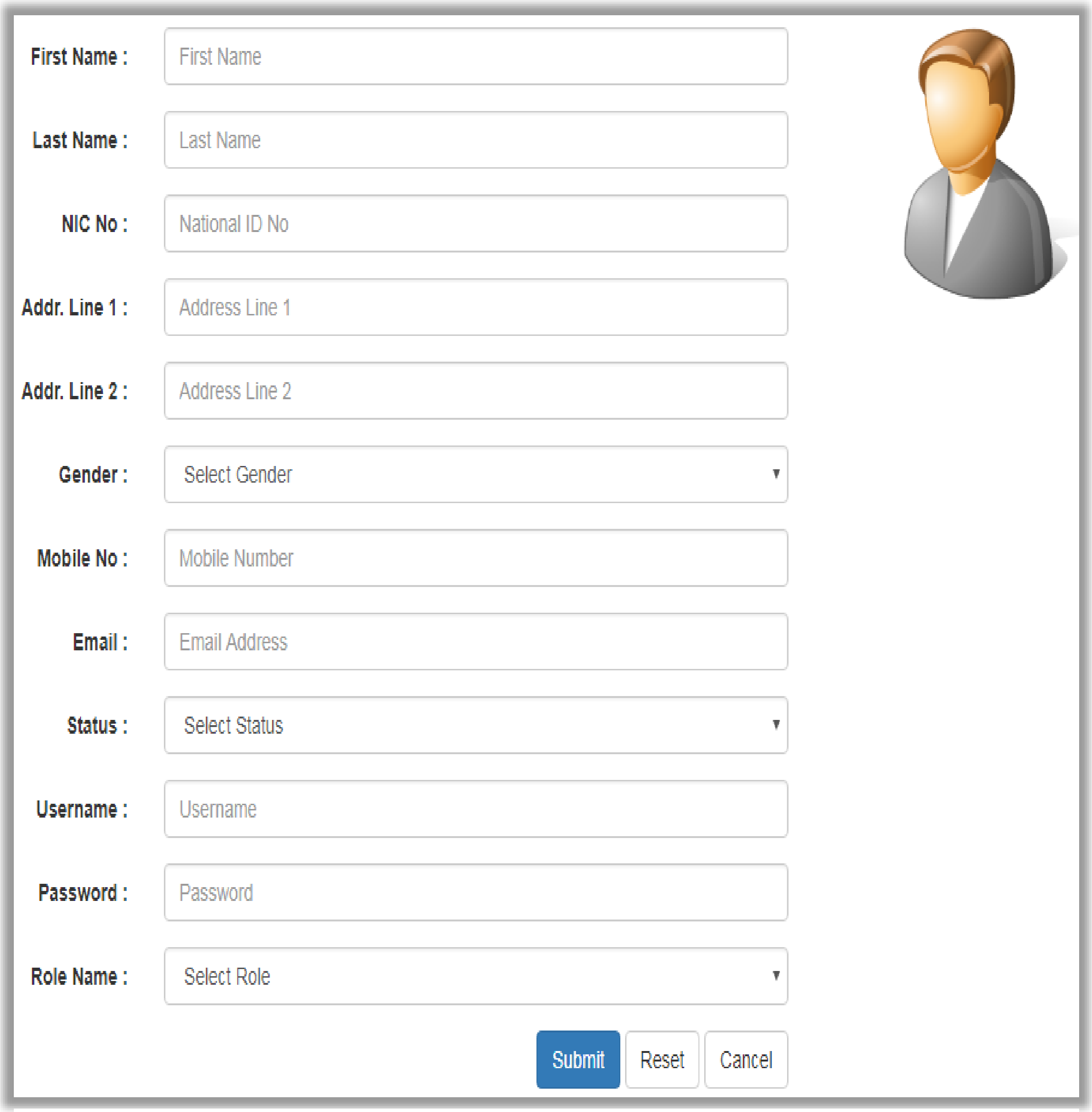


*Figure 3.7: Product Adding Page*

The interface helps to add new products information to the system is displayed in figure 3.7 Can add product based on different supplier, category, sub category including the data validation. Cannot submit form without any empty field.

### 3.6.4 Staff/ User Adding Page

The user's information based on their role. Can give username and password adding same time another information.  Adding information with data validations. The user profile creation belongs to staff management and cannot adding existing user. Only can create user by administrator. If administrator can deactivate user loggings.  Explained in below figure 3.8



First Name : First Name

Last Name : Last Name

NIC No : National ID No

Addr. Line 1 : Address Line 1

Addr. Line 2 : Address Line 2

Gender : Select Gender

Mobile No : Mobile Number

Email : Email Address

Status : Select Status

Username : Username

Password : Password

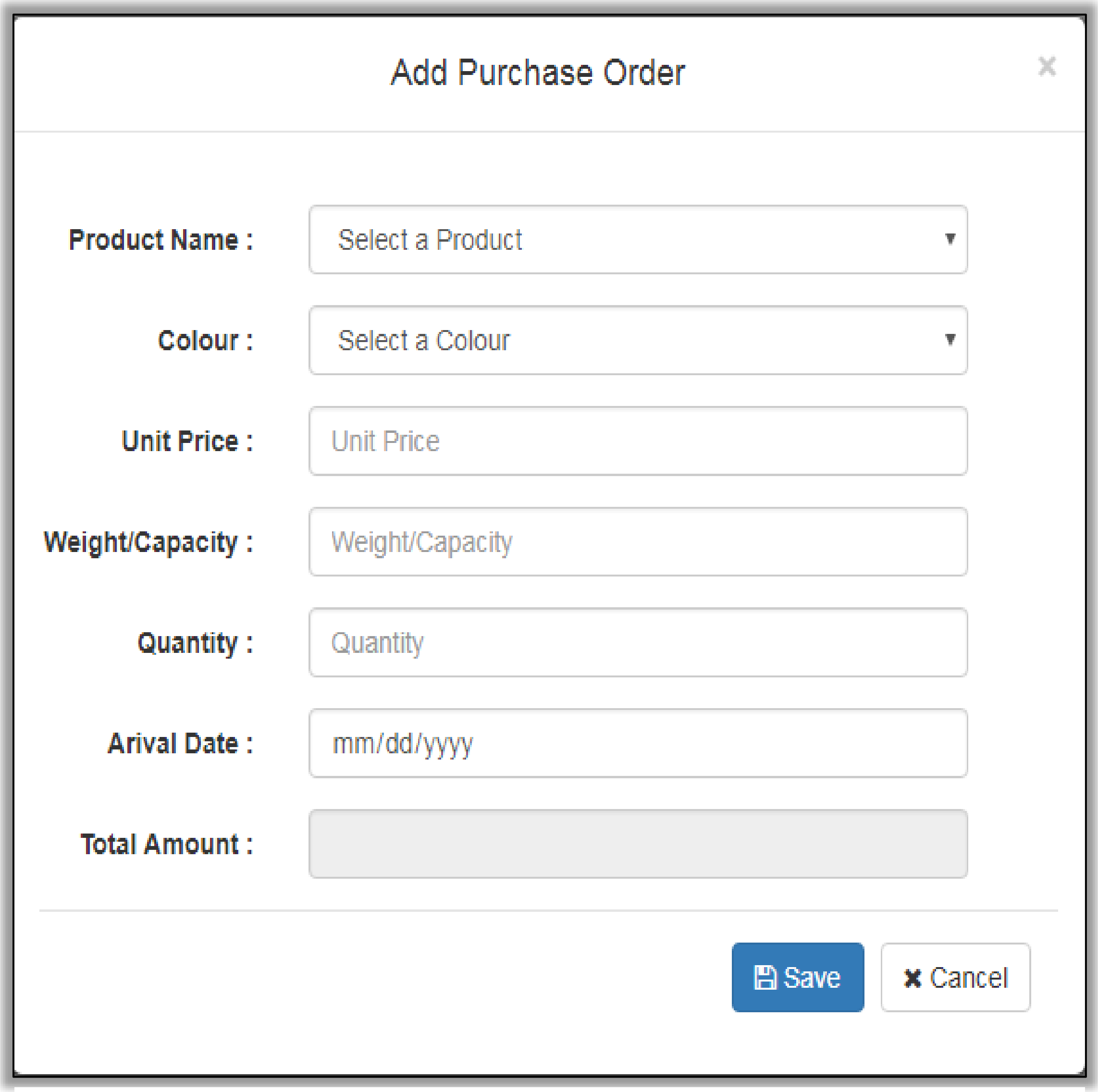
Role Name : Select Role

*Figure 3.8: Staff/User Adding Page*

### 3.6.5 Purchase Order Page

The add product purchase interface which belongs to purchase management in the system is displayed by Figure 3.9. If purchase order is not delivered yet system shows it’s still pending order then after delivered we can update status of the order it help to update stock quantity update purchases same time.

We can purchase products when stock level is low its showed in notification panel product name and status of quantity.



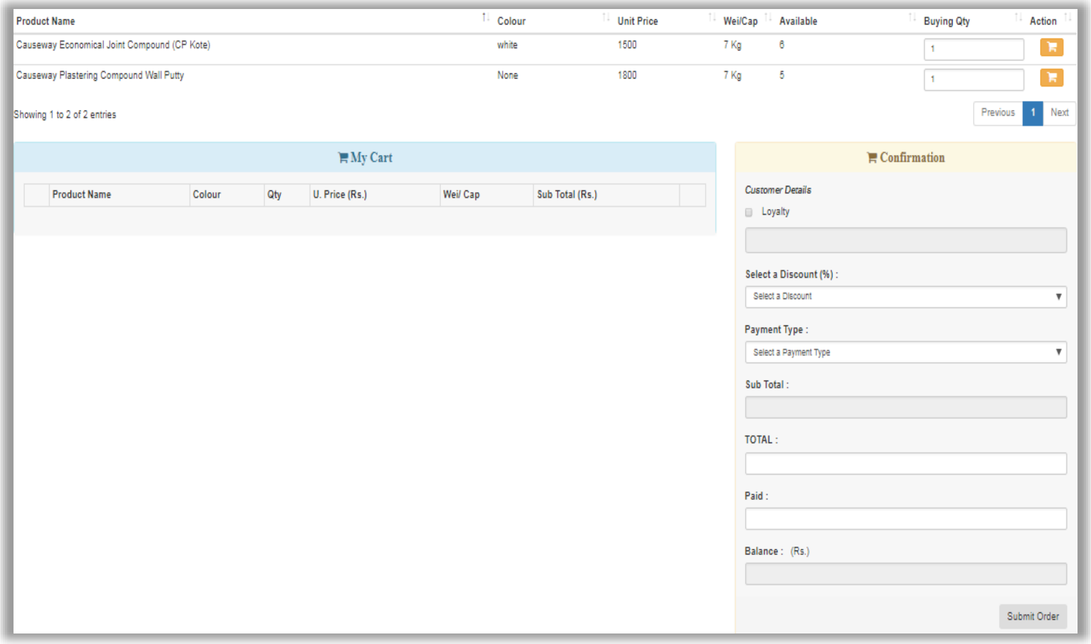
Add Purchase Order

| | |
|---|---|
| Product Name : | Select a Product |
| Colour : | Select a Colour |
| Unit Price : | Unit Price |
| Weight/Capacity : | Weight/Capacity |
| Quantity : | Quantity |
| Arival Date : | mm/dd/yyyy |
| Total Amount : | |

*Figure 3.9: Purchase Order Page*

### 3.6.6 Order Processing Page

Order processing page which belongs to order management. The order processing page display selection of products with quantity and ordered products showing as under cart table with customer selection, and fill confirmation form with loyalty customer mobile number, discount and customer paying amount and print invoice. Complication of validation and data processing explained in figure 3.10.



| Product Name | Colour | Unit Price | Wei/Cap | Available | Buying Qty | Action |
|---|---|---|---|---|---|---|
| Causeway Economical Joint Compound (CP Kote) | white | 1500 | 7 Kg | 6 | 1 | |
| Causeway Plastering Compound Wall Putty | None | 1800 | 7 Kg | 5 | 1 | |

Showing 1 to 2 of 2 entries

My Cart

| | Product Name | Colour | Qty | U. Price (Rs.) | Wei/ Cap | Sub Total (Rs.) | |
|---|---|---|---|---|---|---|---|

Confirmation

*Customer Details*

□ Loyalty

Select a Discount (%) :

Select a Discount ▾

Payment Type :

Select a Payment Type ▾

Sub Total :

TOTAL :

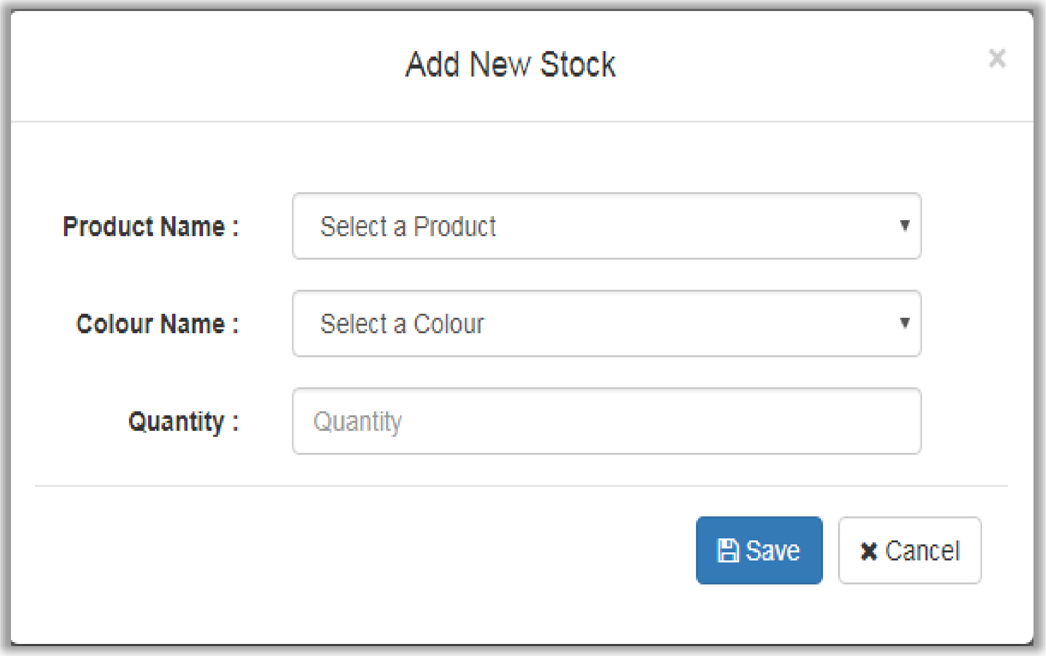
Paid :

Balance : (Rs.)

***Figure 3.10: Order Processing Page***

## 3.6.7 New Stock Adding Page

The stock availability page showing all the products stock quantity, highlighted low quantities. If already added product details then we can add new stock using this form. Cannot add existing stock into the system. Its properly explained in figure 3.11.



Add New Stock

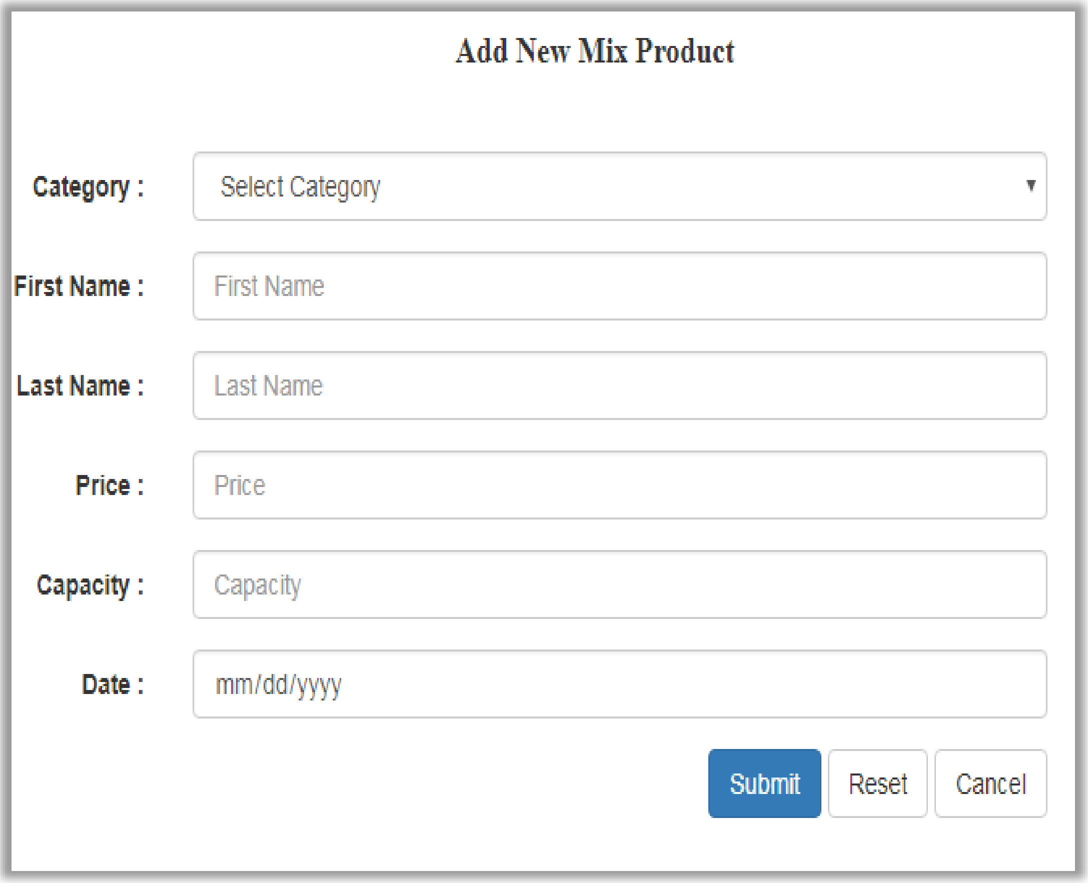
Product Name : Select a Product ▾

Colour Name : Select a Colour ▾

Quantity : Quantity

***Figure 3.11: New Stock Adding Page***

### 3.6.8 Mixing Product Adding Page


Add New Mix Product

Category : Select Category ▾

First Name : First Name

Last Name : Last Name

Price : Price

Capacity : Capacity

Date : mm/dd/yyyy

*Figure 3.12: Mixing Product Adding Page*

Mixing Product s which belongs to stock management module. The mixing products page showing the all paint mixing products in different categories. Its explained in figure 3.12

# Chapter 4 : Implementation

## 4.1 Introduction

After the completion of the design stage, implementation of the system begins according to what design stage planned by using relevant tools and techniques. This is an important stage in the software development life cycle, because this stage builds the system into a working system. This system has been developed as a modularized way and finally integrated all the module as a complete system.

This chapter mainly discuss about the implementation environment, the too land technique as well as the reusable components used to implement the system. Code segments which were included in the system with a comment are meant to be for future improvements.

## 4.2 Implementation Environment

The system implementation environment can be divided into two categories, hardware and software environment. These are listed below in table 4.1

| Hardware Environment | Software Environment |
|---|---|
| AMD A4-1250 APU @ CPU 1.00 GHz | Microsoft Windows 8.1 |
| Microsoft Windows 8.1 | XAMPP v3.2.2 |
| 500GB Hard Disk | Workbench |
| | MS Word, Visio |

***Table 4.1: Implementation Environment***

## 4.3 System Developed Tools and Technologies

When developing this system, the following tools and technologies were used and briefly explained below:

- Adobe Dreamweaver CS6
  This software used for the coding of the system. It contains more supportive background when doing implementation like popup code hints.

- PHP

  This is a server-side object oriented scripting language which used when developing the system.
- MYSQL

  This is database software which used to handle the database of the system.
- HTML

  This is the basic web related language and its helps to keep the system structure clearly.
- AJAX

  Ajax is a JavaScript based technology and its supports update the system without the refreshing.
- CSS

  This used for styling the system interfaced clearly.
- JavaScript

  This is scripting language and its helps to client-side validation and dynamic the system component.
- JSON

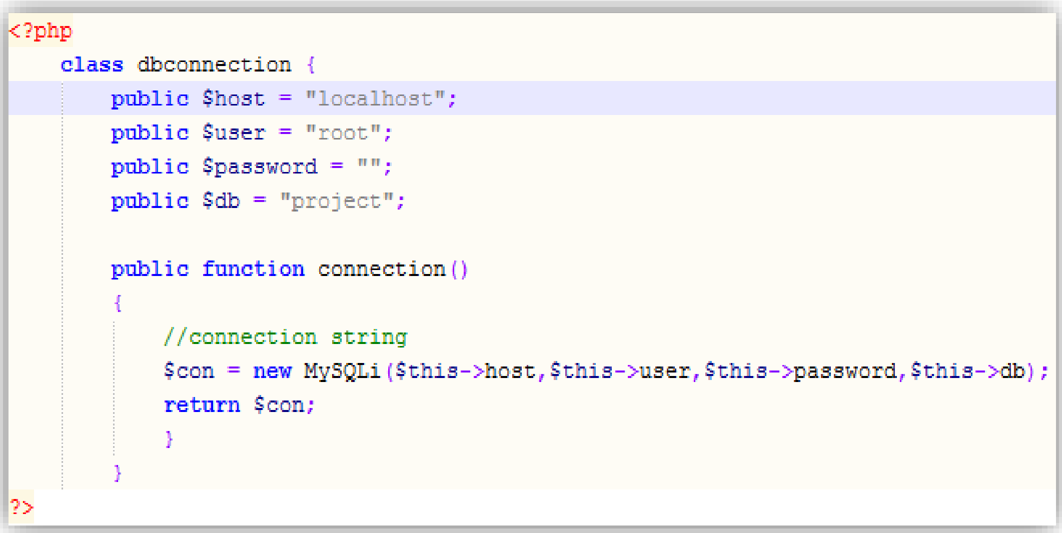
  JSON is syntax for storing and exchanging text information. It’s like XML and JSON is smaller than XML, a faster and easier to parse.

## 4.4 Major Code Segments

The main coding modules developed in the system have been mentioned below briefly explained their functionality. Code modules consists with comment to identify the reason of a particular code line.

- Database Connection Page

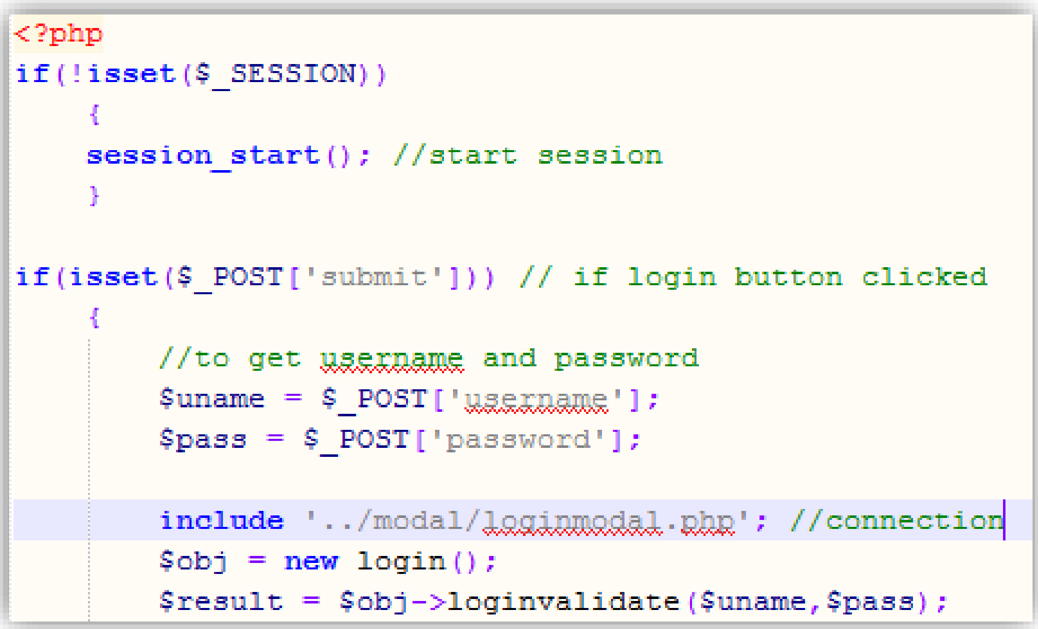
Every web system as well as stand-alone system always connected with database. If something is insert, delete or update in the system, the system always connects with the related database. Code segment given below links the system to its database.



```
<?php
    class dbconnection {
        public $host = "localhost";
        public $user = "root";
        public $password = "";
        public $db = "project";

        public function connection()
        {
            //connection string
            $con = new MySQLi($this->host,$this->user,$this->password,$this->db);
            return $con;
            }
        }
?>
```

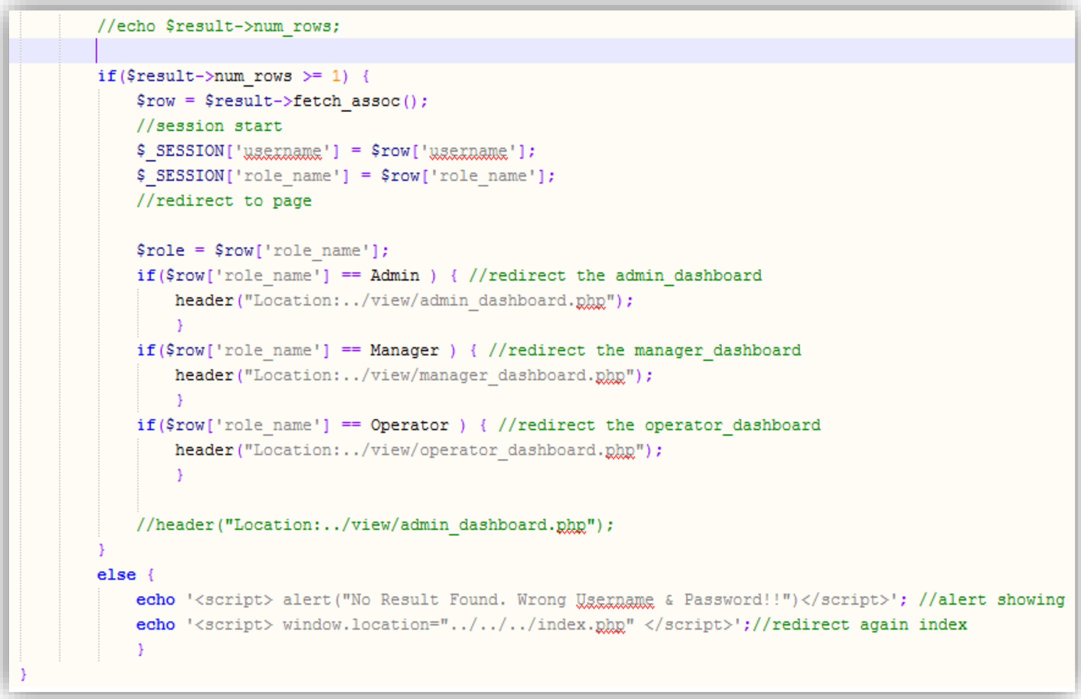
- Login Handling Page

This code segment showing all the required files, classes and start session and also caches the user input for the continuing process.



```
<?php
if(!isset($_SESSION))
    {
    session_start(); //start session
    }

if(isset($_POST['submit'])) // if login button clicked
    {
        //to get username and password
        $uname = $_POST['username'];
        $pass = $_POST['password'];

        include '../modal/loginmodal.php'; //connection
        $obj = new login();
        $result = $obj->loginvalidate($uname,$pass);
```

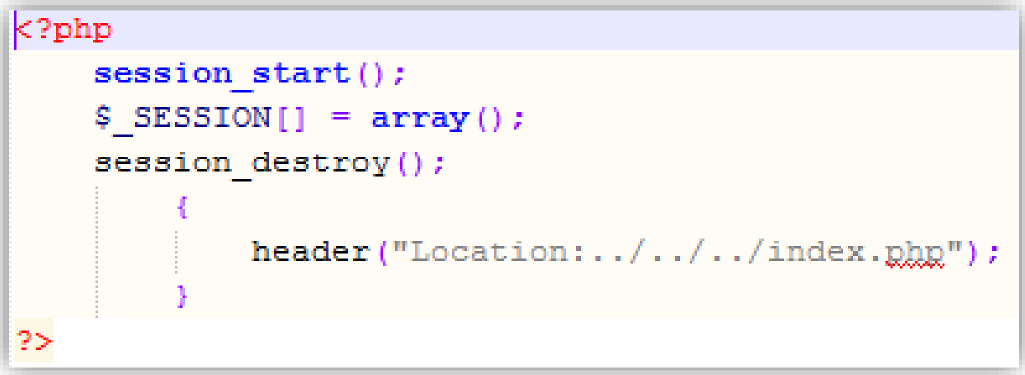
Code segment given below check that user exists according to the input. If exists, then check the password, then check the user status and check user role, whether the person’s account is active or disable. If enabled, continue the process; otherwise redirect to the logging page with error message.



```
        //echo $result->num_rows;

        if($result->num_rows >= 1) {
            $row = $result->fetch_assoc();
            //session start
            $_SESSION['username'] = $row['username'];
            $_SESSION['role_name'] = $row['role_name'];
            //redirect to page

            $role = $row['role_name'];
            if($row['role_name'] == Admin ) { //redirect the admin_dashboard
                header("Location:../view/admin_dashboard.php");
                }
            if($row['role_name'] == Manager ) { //redirect the manager_dashboard
                header("Location:../view/manager_dashboard.php");
                }
            if($row['role_name'] == Operator ) { //redirect the operator_dashboard
                header("Location:../view/operator_dashboard.php");
                }

            //header("Location:../view/admin_dashboard.php");
        }
        else {
            echo '<script> alert("No Result Found. Wrong Username & Password!!")</script>'; //alert showing
            echo '<script> window.location="../../../index.php" </script>';//redirect again index
            }
}
```

- Logout Page

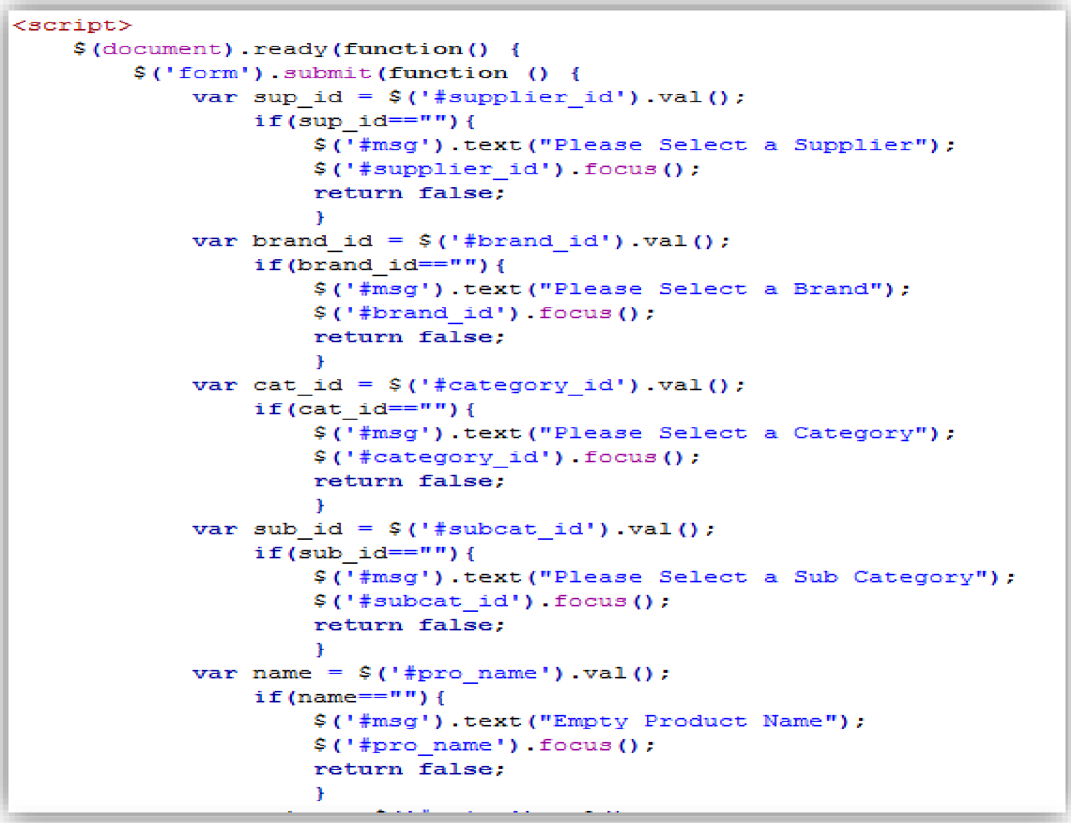
The function below describes about logout page. After user successfully logged in to the system this help to terminate his logged session from the system.



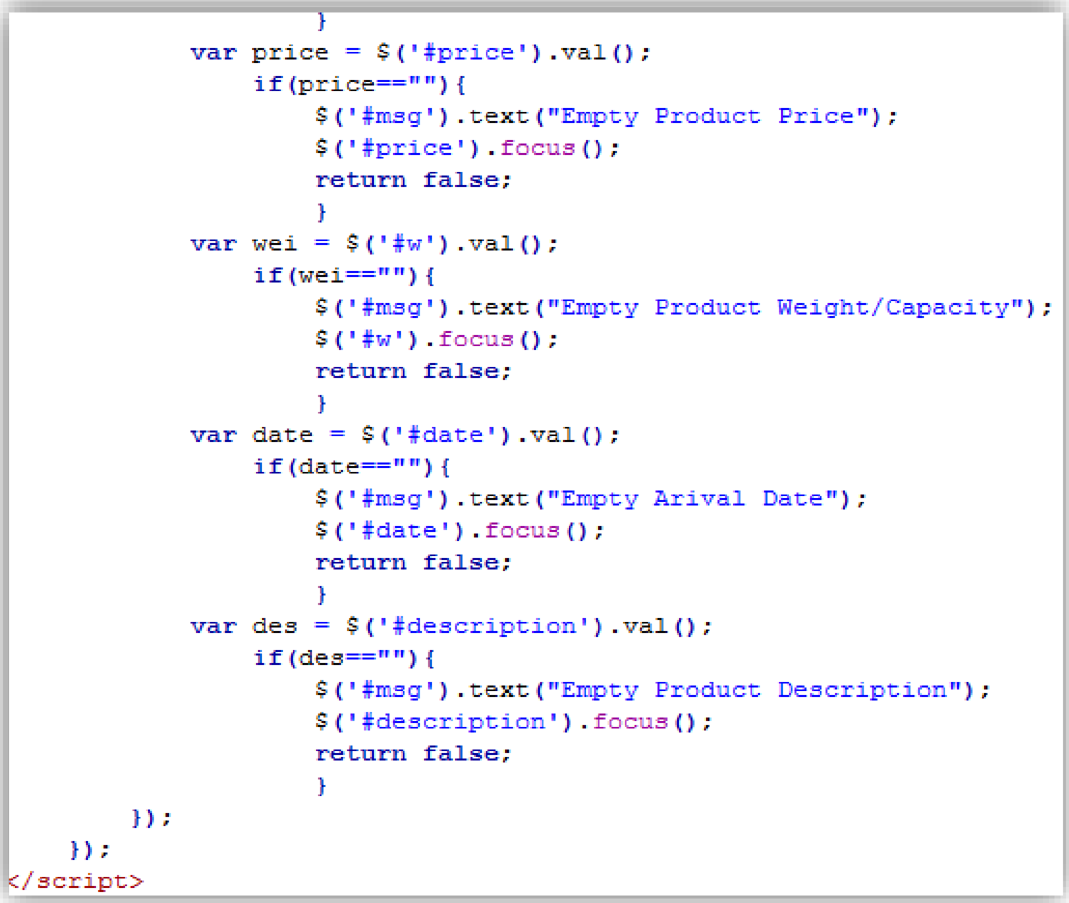
```
<?php
    session_start();
    $_SESSION[] = array();
    session_destroy();
        {
            header("Location:../../../index.php");
        }
?>
```

- Validate Input when Add Products

Adding o products to the system is a major requirement identified in requirement analysis stage. Code segment given below validate product inputs using JavaScript technology and handles errors before its transmitted to the server.



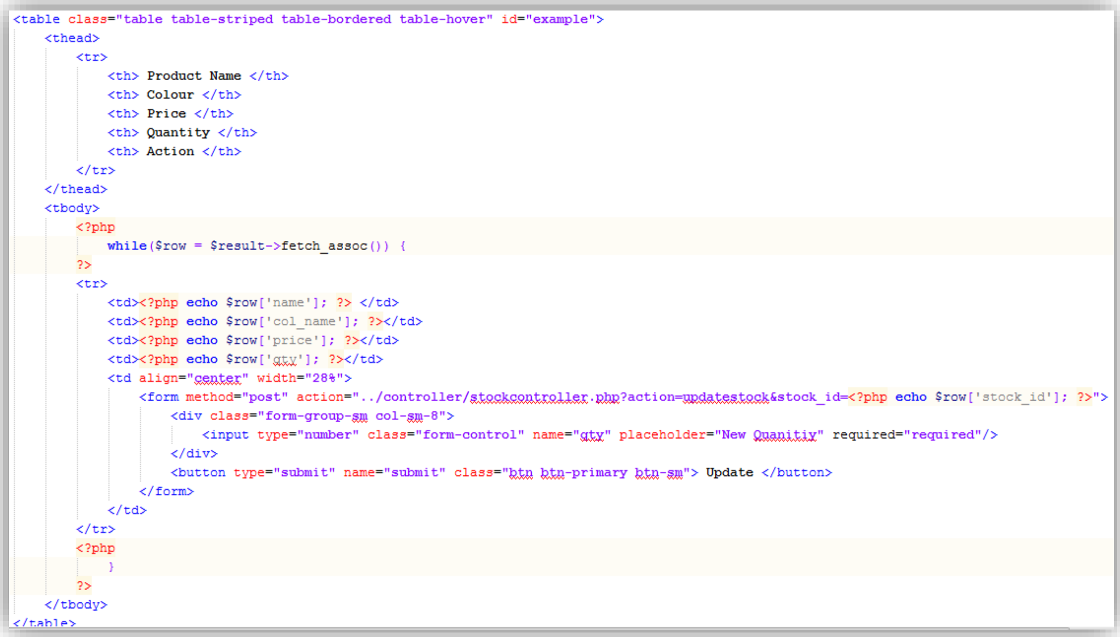
```
<script>
    $(document).ready(function() {
        $('form').submit(function () {
            var sup_id = $('#supplier_id').val();
                if(sup_id==""){
                    $('#msg').text("Please Select a Supplier");
                    $('#supplier_id').focus();
                    return false;
                    }
            var brand_id = $('#brand_id').val();
                if(brand_id==""){
                    $('#msg').text("Please Select a Brand");
                    $('#brand_id').focus();
                    return false;
                    }
            var cat_id = $('#category_id').val();
                if(cat_id==""){
                    $('#msg').text("Please Select a Category");
                    $('#category_id').focus();
                    return false;
                    }
            var sub_id = $('#subcat_id').val();
                if(sub_id==""){
                    $('#msg').text("Please Select a Sub Category");
                    $('#subcat_id').focus();
                    return false;
                    }
            var name = $('#pro_name').val();
                if(name==""){
                    $('#msg').text("Empty Product Name");
                    $('#pro_name').focus();
                    return false;
                    }
```



```
                    }
            var price = $('#price').val();
                if(price==""){
                    $('#msg').text("Empty Product Price");
                    $('#price').focus();
                    return false;
                    }
            var wei = $('#w').val();
                if(wei==""){
                    $('#msg').text("Empty Product Weight/Capacity");
                    $('#w').focus();
                    return false;
                    }
            var date = $('#date').val();
                if(date==""){
                    $('#msg').text("Empty Arival Date");
                    $('#date').focus();
                    return false;
                    }
            var des = $('#description').val();
                if(des==""){
                    $('#msg').text("Empty Product Description");
                    $('#description').focus();
                    return false;
                    }
        });
    });
</script>
```

- Stock Management Page

Stock management can be carried out using this function. Adding, update stock can also be done by following code segments.



```
<table class="table table-striped table-bordered table-hover" id="example">
    <thead>
        <tr>
            <th> Product Name </th>
            <th> Colour </th>
            <th> Price </th>
            <th> Quantity </th>
            <th> Action </th>
        </tr>
    </thead>
    <tbody>
        <?php
            while($row = $result->fetch_assoc()) {
        ?>
        <tr>
            <td><?php echo $row['name']; ?> </td>
            <td><?php echo $row['col_name']; ?></td>
            <td><?php echo $row['price']; ?></td>
            <td><?php echo $row['qty']; ?></td>
            <td align="center" width="28%">
                <form method="post" action="../controller/stockcontroller.php?action=updatestock&stock_id=<?php echo $row['stock_id']; ?>">
                    <div class="form-group-sm col-sm-8">
                        <input type="number" class="form-control" name="qty" placeholder="New Quanitiy" required="required"/>
                    </div>
                    <button type="submit" name="submit" class="btn btn-primary btn-sm"> Update </button>
                </form>
            </td>
        </tr>
        <?php
            }
        ?>
    </tbody>
</table>
```

## 4.5 Reusable Components

- Used bootstrap components
  - Form field styling has been used when styling form interface [11].
  - Tables has been used when styling the tables [12].
- Used jQuery reusable components
  - Data Tables - Pagination plugging's has been used to manage the data contain inside the system [13].
  - Sorting has been used to data to ascending or descending order [14].
- Other Re-Usable Components
  - Ajax functions has been used to pass data between server and the system efficiently and quickly [15].
  - JSON encode and decode function has been used when passing arrays from client side and server-side page [16].
  - Fusion Chart - Use the Fusion Charts for management reports and summarized view of data. [17].

# Chapter 5 : Evaluation

## 5.1 Introduction

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not further testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual requirements.

## 5.2 Software Testing

Software testing is a process that should be done during the development process. In order word software testing validation and verifying the process.

Validation is the process to make sure the system satisfies the specified requirements at the end of the development phase. In other words, make sure that the system is built as customer requirements.

Verification is the process to make sure the product satisfies imposed at the start of the development phase. In other words, make sure system behaves the way we want it to [18].

## 5.3 Techniques of Software Testing

There are two techniques of the software system.

- Black Box Testing

  Black box testing is a testing technique that ignores the internal mechanism of the system and forces on the output generated against any input and execution of the system. It is also called functional testing.

- White Box Testing

  White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation a white box testing is often used for verification.

## 5.4 Types of Testing

### 5.4.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

### 5.4.2 Integration Testing

Integration testing is the testing in which a group of components are combined to produced output and also the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

### 5.4.3 Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. If falls under the class of black box testing.

## 5.5 Test Plan and Test Cases

Test plan is detail about the scope and schedule of the testing, test deliverables etc. it given how the testing will proceed, who will do the testing, what will be tested, in how much time the test will take place, and to what quality level the test will be performed. Following tables show dome test cases major system modules. Follow: (Please refer the appendix E for all test cases and results.

### 5.5.1 Test Cases for User Authentication

| Test No | Test description | Steps to test | Expected result |
|---|---|---|---|
| 1 | Validate user input details | Enter correct username<br>Enter correct password<br>Enable account status | Successfully Login into the system. |
| 2 | Validate user input details | Enter correct username<br>Enter incorrect password<br>Enable account status | Display error message "No Record Found. Wrong Username or Password" |

| | | | |
|---|---|---|---|
| 3 | Validate user input details | Enter incorrect username<br>Enter correct password<br>Enable account status | Display error message “No Record Found. Wrong Username or Password” |
| 4 | Validate user input details | Enter both username & password incorrect<br>Enable account status | Display error message “No Record Found. Wrong Username or Password” |
| 5 | Validate user input details | Enter correct username<br>Enter correct password<br>Enable account status disable | Display error message “No Record Found. Wrong Username or Password” |

***Table 5.1: User Authentication Test Cases***

## 5.5.2 Test Cases for Product Management

| **Test No** | **Test description** | **Steps to test** | **Expected result** |
|---|---|---|---|
| 1 | Validate product information adding | Add information with relevant details | Display message “Record Insert Successfully” |
| 2 | Validate product information adding | Try to add information by keeping all fields in empty | Display message “Empty First Field Name” |
| 3 | Validate product information adding | Enter all fields without select supplier | Display message “Select a Supplier” |
| 4 | Validate product information adding | Add Product for already existing Product Name | Display message “Product Already added” |
| 5 | Update a product info | Choose Product from the table, click view details and update details | Display message “Record Update Successfully” |
| 6 | Delete a product | Choose Product from the table, click delete icon | Popup the confirm dialog box “Do you want to Delete product?” |

***Table 5.2: Manage Product Test Cases***

### 5.5.3 Test Cases for Supplier Management

| Test No | Test description | Steps to test | Expected result |
|---|---|---|---|
| 1 | Validate Supplier information adding | Add information with relevant details | Display message "Record Insert Successfully" |
| 2 | Validate Supplier information adding | Try to add information by keeping all fields in empty | Display message "Empty First Field Name" |
| 3 | Validate Supplier information adding | Enter all fields without First Field | Display message "Empty First Name" |
| 4 | Validate Supplier information adding | Add Supplier for already existing Supplier Name | Display message "Already added" |
| 5 | Update a Supplier info | Choose Supplier from the table, click view details and update details | Display message "Record Update Successfully" |
| 6 | Delete a Supplier | Choose Supplier from the table, click delete icon | Popup the confirm dialog box "Do you want to Delete Supplier?" |

***Table 5.3: Manage Supplier Test Cases***

### 5.5.4 Test Cases for Customer Management

| Test No | Test description | Steps to test | Expected result |
|---|---|---|---|
| 1 | Validate Loyalty Customer information adding | Add information with relevant details | Display message "Record Insert Successfully" |
| 2 | Validate Loyalty Customer information adding | Try to add information by keeping all fields in empty | Display message "Empty First Field Name" |
| 3 | Validate Loyalty Customer information adding | Enter all fields without First Field | Display message "Empty First Name" |
| 4 | Validate Loyalty Customer information adding | Add Loyalty Customer for already existing Loyalty Customer NIC | Display message "Already added" |

|  |  |  |  |
|---|---|---|---|
| 5 | Update a Loyalty Customer info | Choose Loyalty Customer from the table, click view details and update details | Display message “Record Update Successfully” |
| 6 | Delete a Loyalty Customer | Choose Loyalty Customer from the table, click delete icon | Popup the confirm dialog box “Do you want to Delete Loyalty Customer?” |

*Table 5.4: Manage Customer Test Cases*

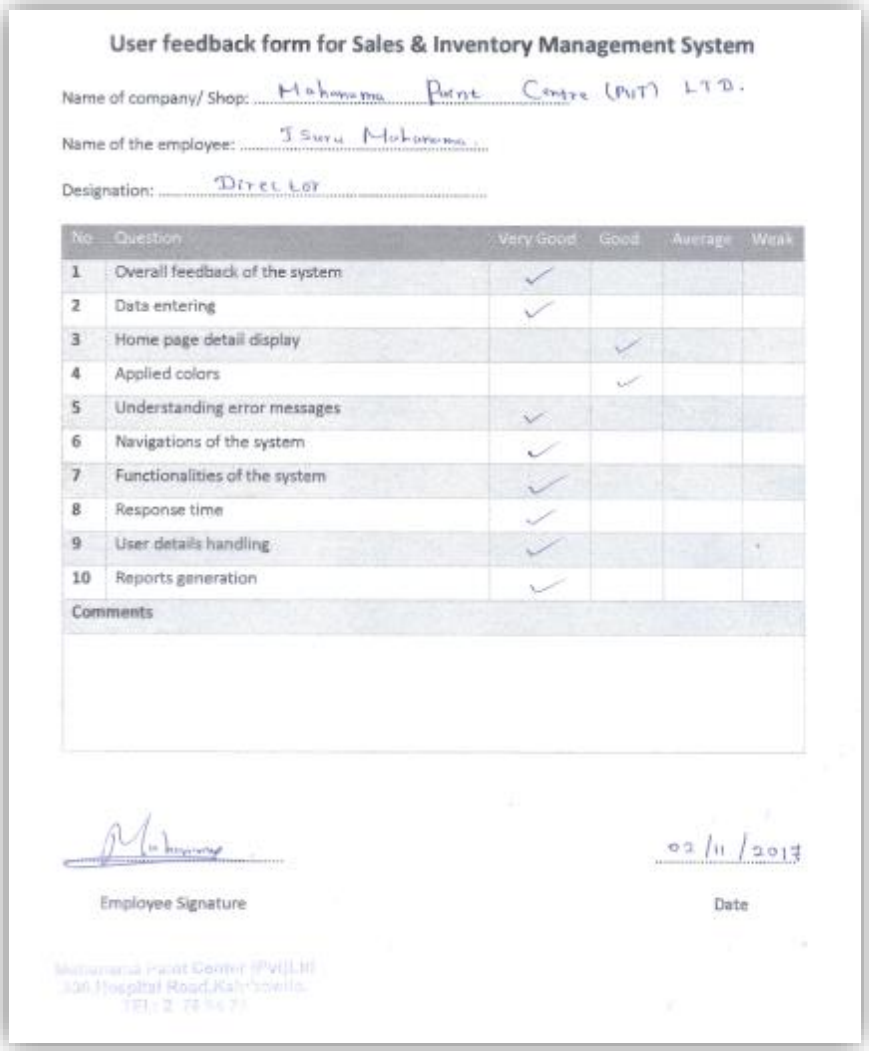
### 5.5.5 Test Cases for Purchase Order Management

| Test No | Test description | Steps to test | Expected result |
|---|---|---|---|
| 1 | Validate Purchase Order information adding | Add information with relevant details | Display message “Record Insert Successfully” |
| 2 | Validate Purchase Order information adding | Try to add information by keeping all fields in empty | Display message “Empty First Field Name” |
| 3 | Validate Purchase Order information adding | Enter all fields without select Product | Display message “Select a Product” |
| 4 | Validate Purchase Order information adding | Add unit price & quantity | Display Total last text field. |
| 5 | Update a Purchase Order info | Choose Purchase Order from the table, click view details and update details | Display message “Record Update Successfully” |
| 6 | Delete a Purchase Order | Choose Purchase Order from the table, click delete icon | Popup the confirm dialog box “Do you want to Delete Purchase Order?” |

*Table 5.5: Manage Purchase Orders Test Cases*

## 5.5 User Evaluation

In this Sales & Inventory Management System managing director taken as Administrator because other user has taken low privileges in this system. Following figure 5.1 shows user evaluation result sheet.



**User feedback form for Sales & Inventory Management System**

Name of company/ Shop: Maharagama Paint Centre (PVT) LTD.

Name of the employee: Isuru Maharagama

Designation: Director

| No | Question | Very Good | Good | Average | Weak |
|---|---|---|---|---|---|
| 1 | Overall feedback of the system | ✓ | | | |
| 2 | Data entering | ✓ | | | |
| 3 | Home page detail display | | ✓ | | |
| 4 | Applied colors | | ✓ | | |
| 5 | Understanding error messages | ✓ | | | |
| 6 | Navigations of the system | ✓ | | | |
| 7 | Functionalities of the system | ✓ | | | |
| 8 | Response time | ✓ | | | |
| 9 | User details handling | ✓ | | | |
| 10 | Reports generation | ✓ | | | |
| Comments | | | | | |

Employee Signature

02/11/2017

Date

Maharagama Paint Center (Pvt)Ltd
306 Hospital Road,Kalubowila.
TEL: 2 74 54 71

*Figure 5.1: User Evaluation Sheet*

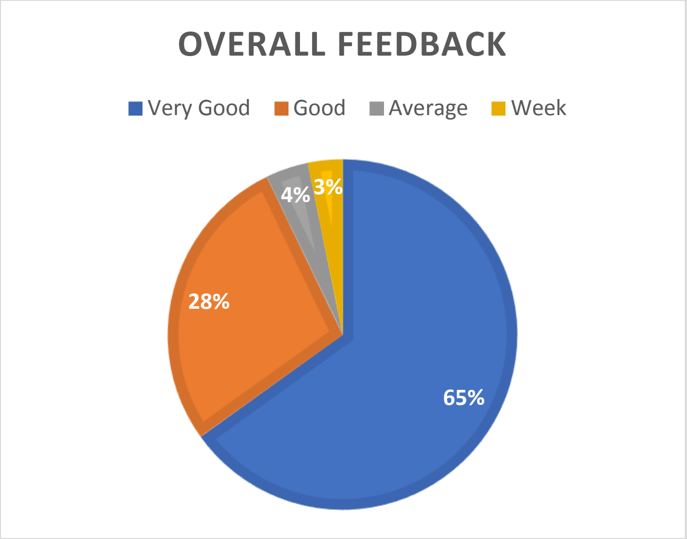


*Figure 5.2: User Evaluation Summery*

# Chapter 6 : Conclusion

## 6.1 Introduction

With today’s competitive market in sri lanka, among the good paint shops as well as many new paint shops, the Mahaanama Paint Center (PVT) Ltd success their business long time period with owner’s experience.

From the start of the shop its small shop and they have maintained the business process by using traditional manual system. With the manual system they faced lots of difficulties such as stock handling, order processing and product ordering. This sales & inventory management system was developed to solve this drawback as well as optimize their overall business process.

From the requirement analysis stage, the client requirement was carefully analyzed and system was developed by adding more functionalities based on client requirements. Automating some process, it reduced and help to saving time of the client efficiently.

## 6.2 Future Improvement

Developing a high-level system is huge task. Some suggestion for improvement’s in the future are as follows:

- Providing SMS facility to loyalty customers
  By implementing this suggestion, send offers and order details via SMS to loyalty customer.
- Provide place purchase order via email by implementing this suggestion, buy products from suppliers via email order.
- Add non-system user’s profiles by implementing this suggestion, administrator can add non-system user profiles.

## 6.3 Lesson Learnt

This project gave me a real life valuable programming experience and other skills. I learnt how to use programming theories in practically. Further when we delivered the project in a time constraint, and I learnt how to handle the pressure situations in project. It is good experience for my life.

Following are some lesson learnt things.

- While doing this project some database normalization issues were came since solving these problems, it helps to improve my database knowledge.
- Since doing the PHP programing codes, it helps to improve my programming language.
- During the evaluation phase I learnt about how to testing and creating test cases for the system.

## 6.4 Problems Encountered and Solutions

- Shop operators didn’t have no idea about the system requirements it’s hard to explained and gather the requirements.
- Some workers are not having proper IT knowledge to handle the system.

# References

[1] https://en.wikipedia.org/wiki/Functional_requirement (2017.10.05).

[2] https://en.wikipedia.org/wiki/Non-functional_requirement (2017.10.04).

[3] http://www.losoftware.co.uk/shop/rightcontrol-lite-free-inventory-management-software/ (2017.10.06).

[4] http://www.sourcecodester.com (2017.10.02).

[5] https://www.tutorialspoint.com/sdlc/sdlc (2017.10.06).

[6] [Ian Somerville]**,** I. Somerville, Software Engineering, 9th edition, Addison-Wesley.

[7**]**https://en.wikipedia.org/wiki/Rational_Unified_Process (2017.10.02).

[8] https://www.tutorialspoint.com/object_oriented_analysis_design/ooad_object_oriented_paradigm.htm (2017.10.01).

[9] https://en.wikipedia.org/wiki/Class_diagram (2017.10.10).

[10] https://en.wikipedia.org/wiki/Database_design (2017.10.05).

[11] https://en.wikipedia.org/wiki/Database_normalization (2017.10.03).

[12] https://getbootstrap.com (2017.10.31).

[13] https://www.w3schools.com/bootstrap/bootstrap_tables.asp (2010.11.01).

[14] https://datatables.net/ (201.11.03).

[15] https://www.w3schools.com/xml/ajax_examples.asp (2017.11.06).

[16] https://www.w3schools.com/js/js_json_intro.asp (2017.11.02).

[17] https://www.fusioncharts.com/ (2017.10.31).

[18] https://www.tutorialspoint.com/software_testing (2017.11.03).

# Appendix A - System Documentation

## System Manual

Tis documentation is describe steps for, how to install this Sales & Inventory Management System. These steps explained about the software and hardware environment clearly you need to installed. When install the system, this documentation helps to how to install it.

**Step 1: -** System Requirements can be verified according to Table A.1 and A.2

### Hardware Requirements

| Hardware | Minimum requirements |
|---|---|
| **Processor** | AMD A4-1250 APU @ CPU 1.00 GHz |
| **Memory** | 512MB memory Capacity |
| **Hard disk** | 40GB capacity |
| **Display** | 1366*768 resolution monitor |
| **Printer** | Inkjet Printer or other |

***Table A.1: Hardware Requirements***

### Software requirements

| Software | Minimum requirements |
|---|---|
| **Operating System** | Windows XP, Windows 7, Windows 8.1 |
| **XAMPP** | XAMPP v3.2.2 |
| **Web browser** | Google chrome (latest version),<br>Firefox (latest version) |
| **Image editor** | Adobe Photoshop CS6 (Need for future Development) |
| **Code editor** | Adobe Dreamweaver (Need for future Development) |
| **PDF convertor** | Foxit reader plugging or similar PDF convertor |

***Table A.2: Software Requirements***

**Step 2: -** Install XAMPP Server

- Download and install XAMPP for windows (refer table a.2 for minimum version) from http://www.apachefriends.org Give installing path to C:\XAMPP of the computer.

Install Web Browser

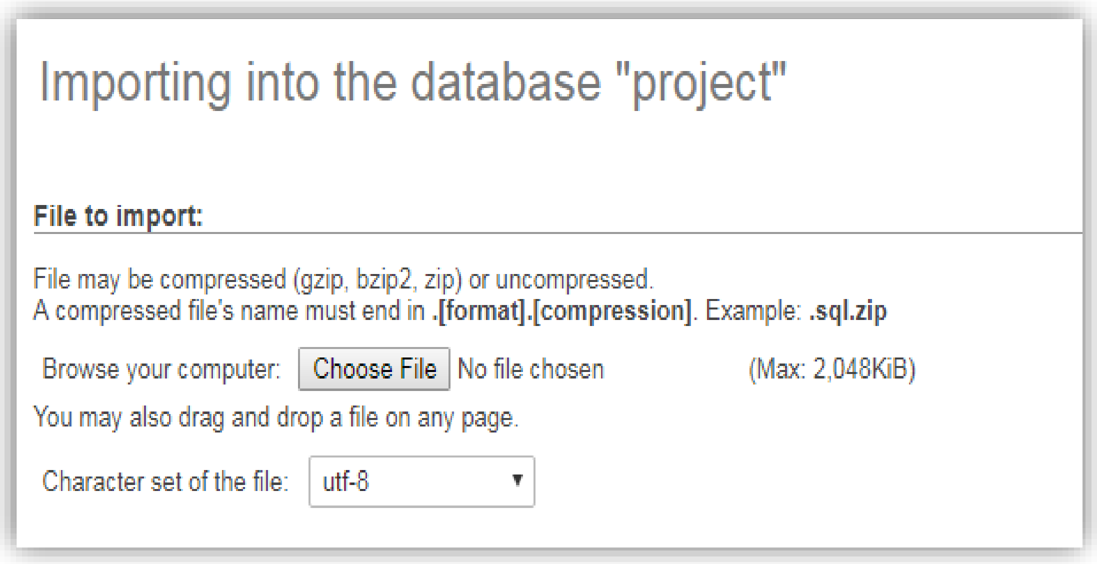
- Install Browser (refer table A.2 for the Version and Recommend Browser).

**Step 3: -** File Extraction

- Open the CD and copy the SIMS folder and paste it to the directory path "C:\XAMPP\htdocs"

**Step 4: -** Database Installation

- Open the web browser and type the URL http://localhost/phpmyadmin and enter Username and Password (if you set username & password).
- Create empty database by providing name as "**project**" and navigate to the "**Import**" tab and click "**choose file**" button. Then browse the CD and select the "**project.sql**" File by opening Database folder.
- Then press "**Go**" button located in the bottom of the page.



Importing into the database "project"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File No file chosen (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

## Final Step (Launching System)

- Verify the XAMPP is running, go to the "C:\XAMPP\" and open the control panel and verify whether Apache, PHP, MySQL running.
- Open the installed web browser and type the URL http://localhost/SIMS and press "Enter" button to access the system.
- Please refer the Appendix-C User Documentation to get the idea about the how to operate the system.

# Appendix B – Design Documentation

## Use case Diagram with Detailed Information

- **Product Management Module**

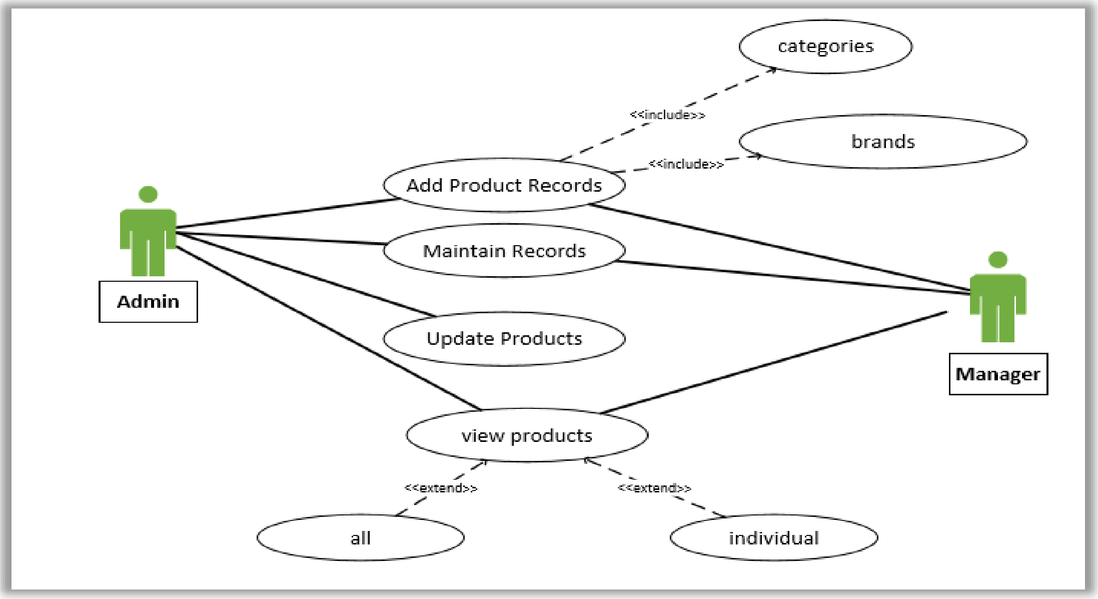


*Figure B.1: Use Case Diagram for Product Management Module*

| | | |
|---|---|---|
| **Use case name** | Adding product info | |
| **Actors** | Admin, Manager | |
| **Description** | admin/ manager adds products to system based on category, brands | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Select supplier, category, sub category and other details | |
| | | 2. Show message if already added product name |
| | 3. Click submit button | |
| | | 4. System display successfully inserted massage |
| **Alternative courses** | 5. System displays the error message | |
| **Conclusion** | This use case emphasizes the create new product | |
| **Post condition** | The data are saved in the database | |

*Table B.1: Use Case Description - Product Adding*

- **Supplier Management Module**

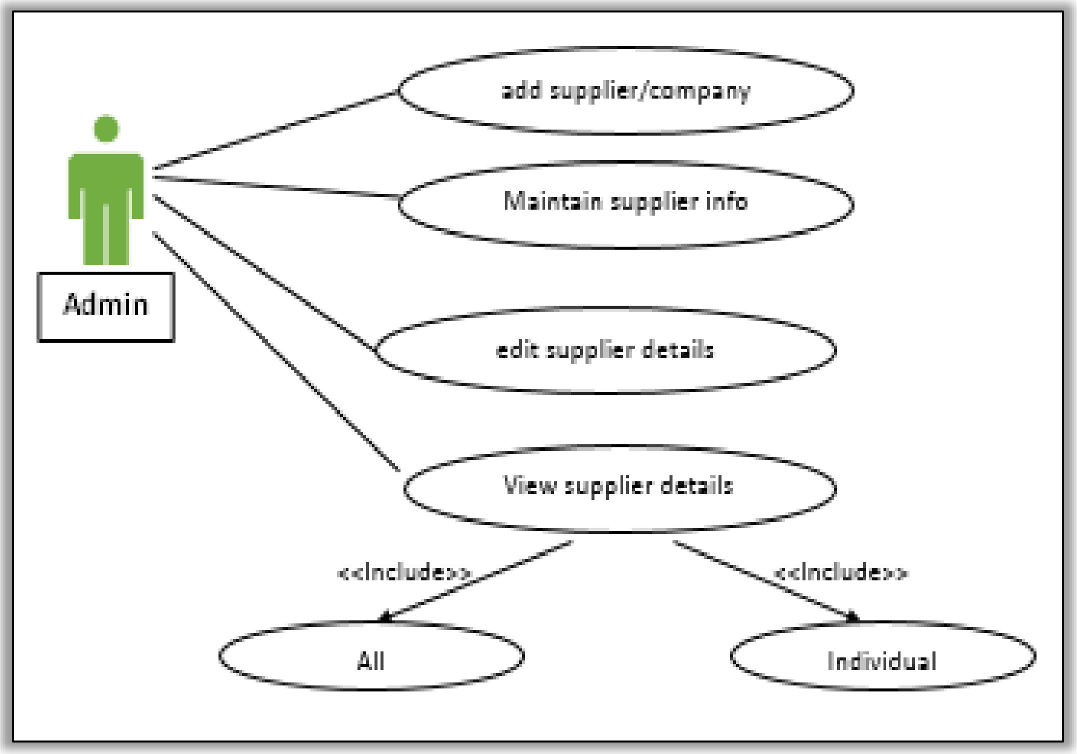


*Figure B.2: Use Case Diagram for Supplier Management Module*

| | | |
|---|---|---|
| **Use case name** | Adding supplier info | |
| **Actors** | Admin | |
| **Description** | admin adds suppliers to the system | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the supplier details | |
| | | 2. Show message if already added supplier name |
| | 3. Click submit button | |
| | | 4. System display successfully inserted massage |
| **Alternative courses** | 5. System displays the error message | |
| **Conclusion** | This use case emphasizes the create new supplier | |
| **Post condition** | The data are saved in the database | |

*Table B.2: Use Case Description - Adding Supplier*

- **Customer Management Module**

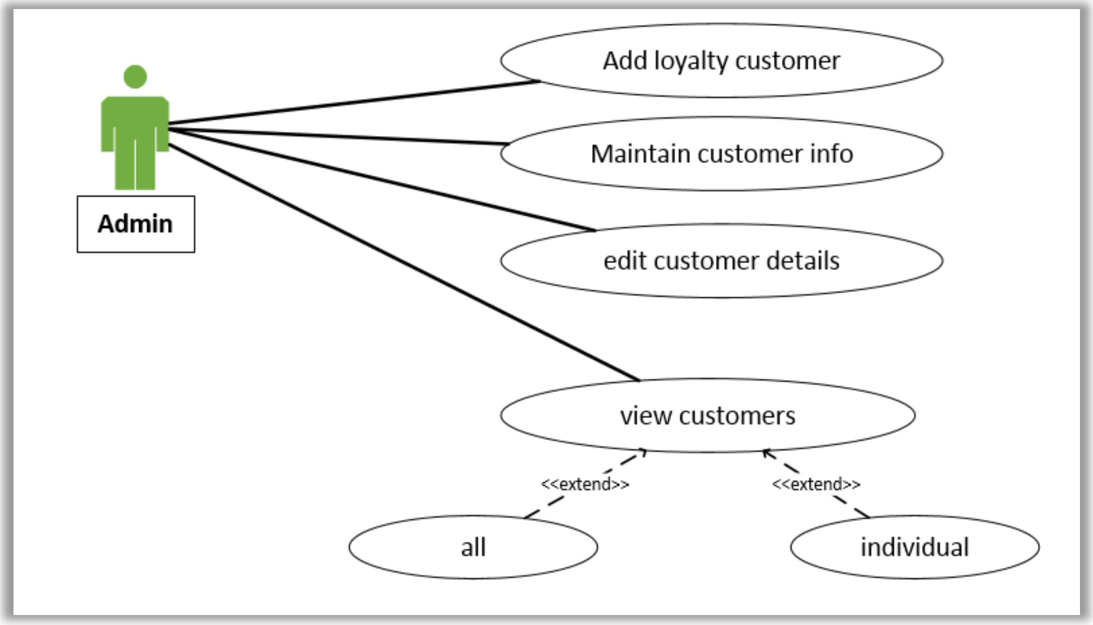


*Figure B.3: Use Case Diagram for Customer Management Module*

| **Use case name** | Adding loyalty customer info | |
|---|---|---|
| **Actors** | Admin | |
| **Description** | admin adds loyalty customers to the system | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the loyalty customer details | |
| | | 2. Show message if already added customer NIC |
| | 3. Click submit button | |
| | | 4. System display successfully inserted massage |
| **Alternative courses** | 5. System displays the error message | |
| **Conclusion** | This use case emphasizes the create new Loyalty customers | |
| **Post condition** | The data are saved in the database | |

*Table B.3: Use Case Description - Adding Customers*

- **Staff/user Management Module**

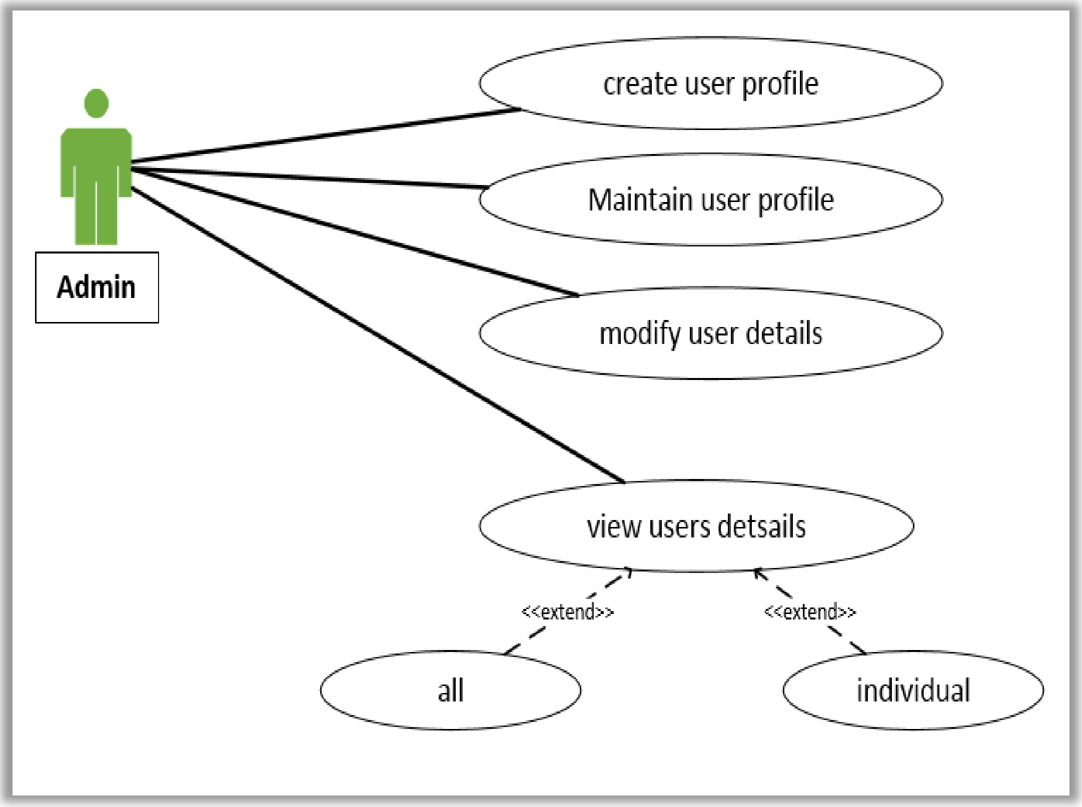


*Figure B.4: Use Case Diagram for Staff/User Management Module*

| | | |
|---|---|---|
| **Use case name** | Adding user/staff info | |
| **Actors** | Admin | |
| **Description** | Admin create user/staff accounts to the system | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the user details | |
| | 2. Click submit button | |
| | | 3. System display successfully inserted massage |
| **Alternative courses** | 4. System displays the error message | |
| **Conclusion** | This use case emphasizes the create new user/staff | |
| **Post condition** | The data are saved in the database | |

*Table B.4: Use Case Description - Adding Staff/ User*

- **Purchase Management Module**

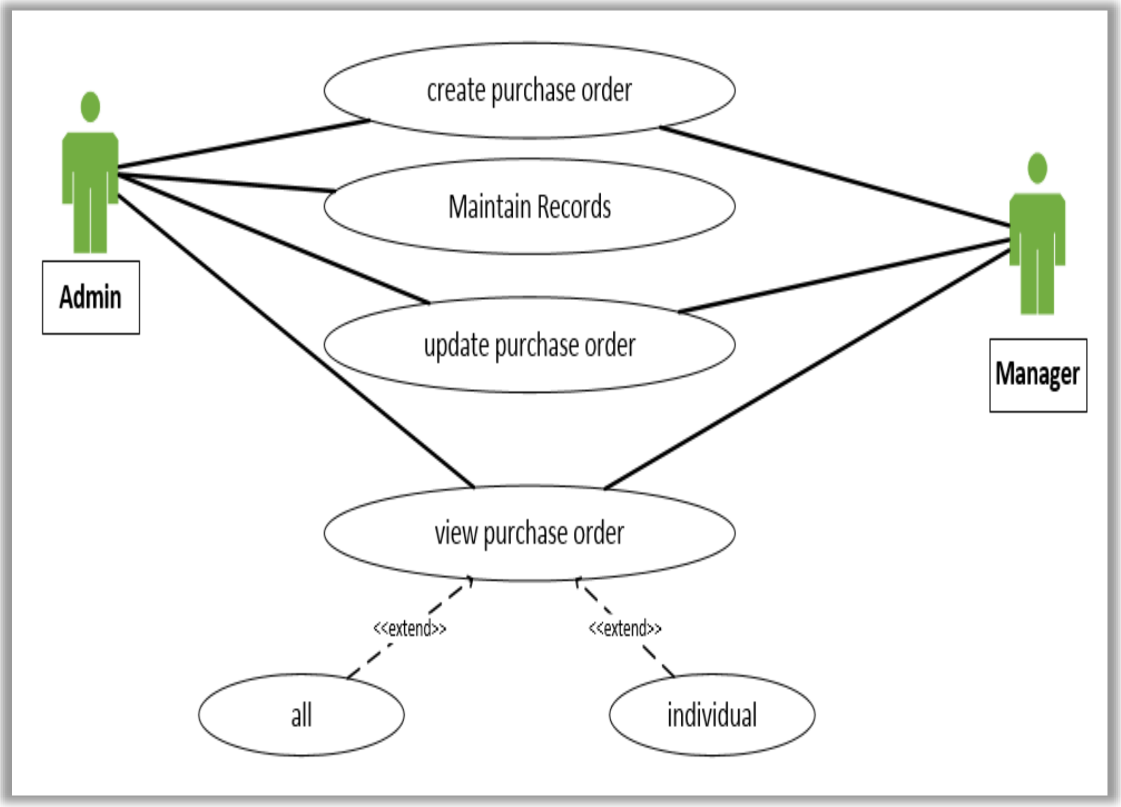


*Figure B.5: Use Case Diagram for Purchase Management Module*

| | | |
|---|---|---|
| **Use case name** | Place purchase order | |
| **Actors** | Admin/ Manager | |
| **Description** | Admin/ Manager place purchase orders | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the purchase order details | |
| | 2. Click submit button | |
| | | 3. System display successfully inserted massage |
| **Alternative courses** | 4. System displays the error message | |
| **Conclusion** | This use case emphasizes the place purchase orders | |
| **Post condition** | The data are saved in the database | |

*Table B.5: Use Case Description - Place the Purchase Order*

| | | |
|---|---|---|
| **Use case name** | Update purchase order | |
| **Actors** | Admin/ Manager | |
| **Description** | Admin/ Manager Update purchase orders | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Check the purchase order details with delivered | |
| | 2. Click submit button | |
| | | 3. Update status as complete |
| | | 4. Update stock quantity |
| | | 5. System display successfully update massage |
| **Alternative courses** | | |
| **Conclusion** | This use case emphasizes the update purchase orders | |
| **Post condition** | The data are saved in the database and update quantity | |

*Table B.6: Use Case Description - Update Placed Order*

- **Stock Management Module**

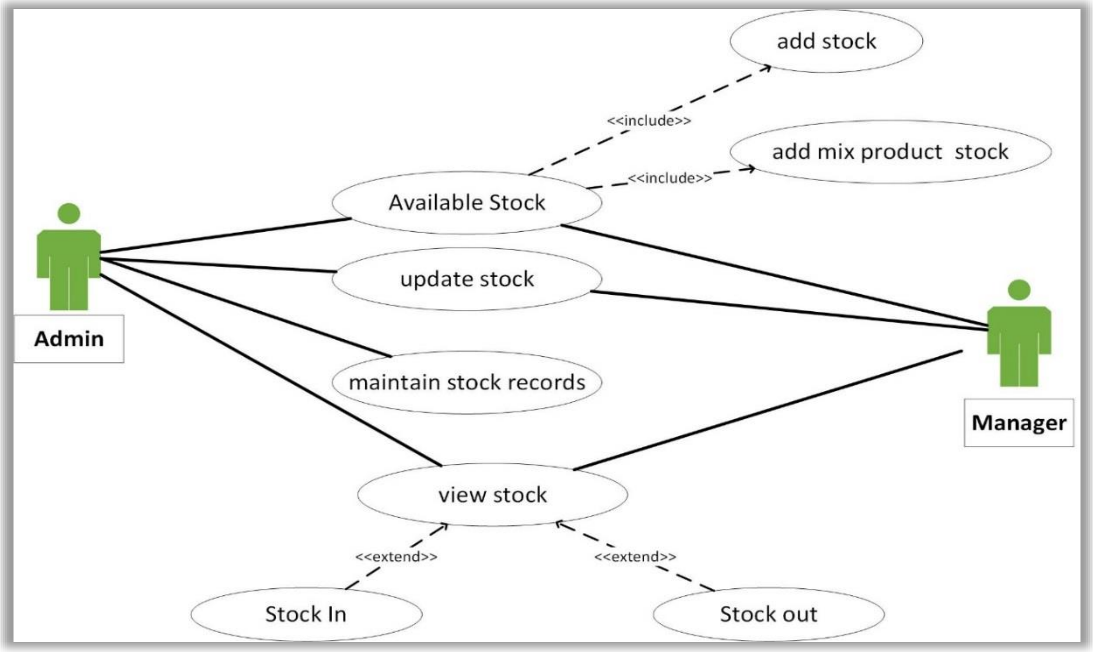


*Figure B.6: Use Case Diagram for Stock Management Module*

| | | |
|---|---|---|
| **Use case name** | Update Stock | |
| **Actors** | Admin/ Manager | |
| **Description** | Admin/ Manager can add/update stock levels | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the Quantity | |
| | 2. Click Update button | |
| | | 3. System display successfully inserted massage |
| **Alternative courses** | 4. System displays the error message | |
| **Conclusion** | This use case emphasizes the Update Stock | |
| **Post condition** | The data are saved in the database | |

*Table B.7: Use Case Description - Update Stock Levels*

- **Order Management Module**

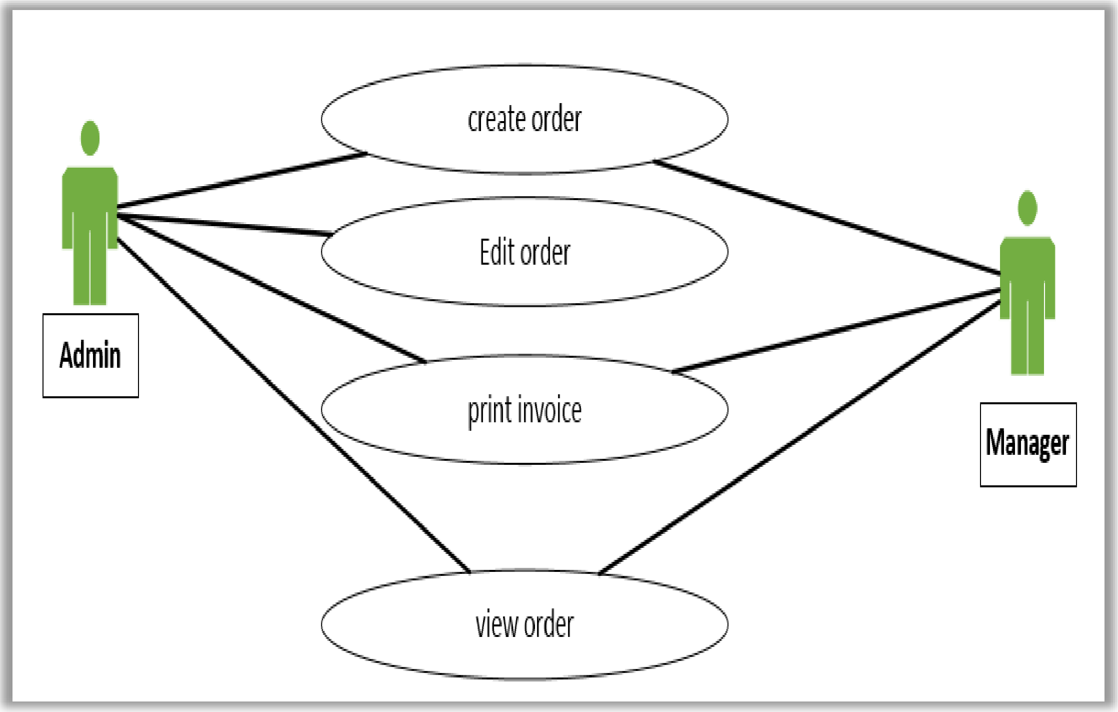


*Figure B.7: Use Case Diagram for Order Management Module*

| | | |
|---|---|---|
| **Use case name** | Order handling | |
| **Actors** | Admin/ sales operator | |
| **Description** | Admin/ sales operator place orders | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the order details | |
| | 2. Select customer type | |
| | 3. Select discount | |
| | 4. Select payment method | |
| | 5. Click checkout button | |
| | | 6. Update stock quantity |
| | | 7. View invoice |
| | | 8. System display successfully place order massage |
| **Alternative courses** | 9. System displays the error message | |
| **Conclusion** | This use case emphasizes the place orders | |
| **Post condition** | The data are saved in the database | |

*Table B.8: Use Case Description - Order Handling*

- **Return Management Module**

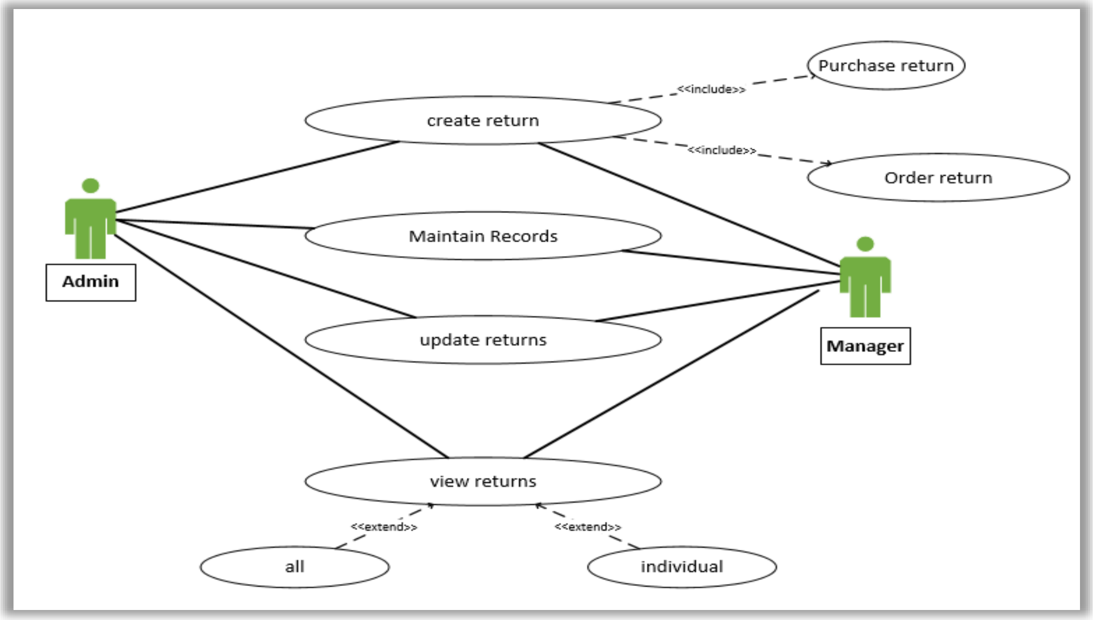


*Figure B.8: Use Case Diagram for Return Management Module*

| | | |
|---|---|---|
| **Use case name** | Purchase Return handling | |
| **Actors** | Admin/ Manager | |
| **Description** | Admin/ Manager Return the Damage Products | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the Purchase return details | \ |
| | 2. Click submit button | |
| | | 3. Update stock |
| | | 4. System display successfully return massage |
| **Alternative courses** | 5. System displays the error message | |
| **Conclusion** | This use case emphasizes the return purchases | |
| **Post condition** | The data are saved in the database | |

***Table B.9:Use Case Description - Purchase Return Handling***

| | | |
|---|---|---|
| **Use case name** | Order Return handling | |
| **Actors** | Admin/ Manager | |
| **Description** | Admin/ Manager Return the orders | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Enter the return order details | |
| | 2. Click submit button | |
| | | 3. Update stock |
| | | 4. System display successfully return order massage |
| **Alternative courses** | 5. System displays the error message | |
| **Conclusion** | This use case emphasizes the Return orders | |
| **Post condition** | The data are saved in the database | |

***Table B.10: Use Case Description - Order Return Handling***

- **Reports Management**

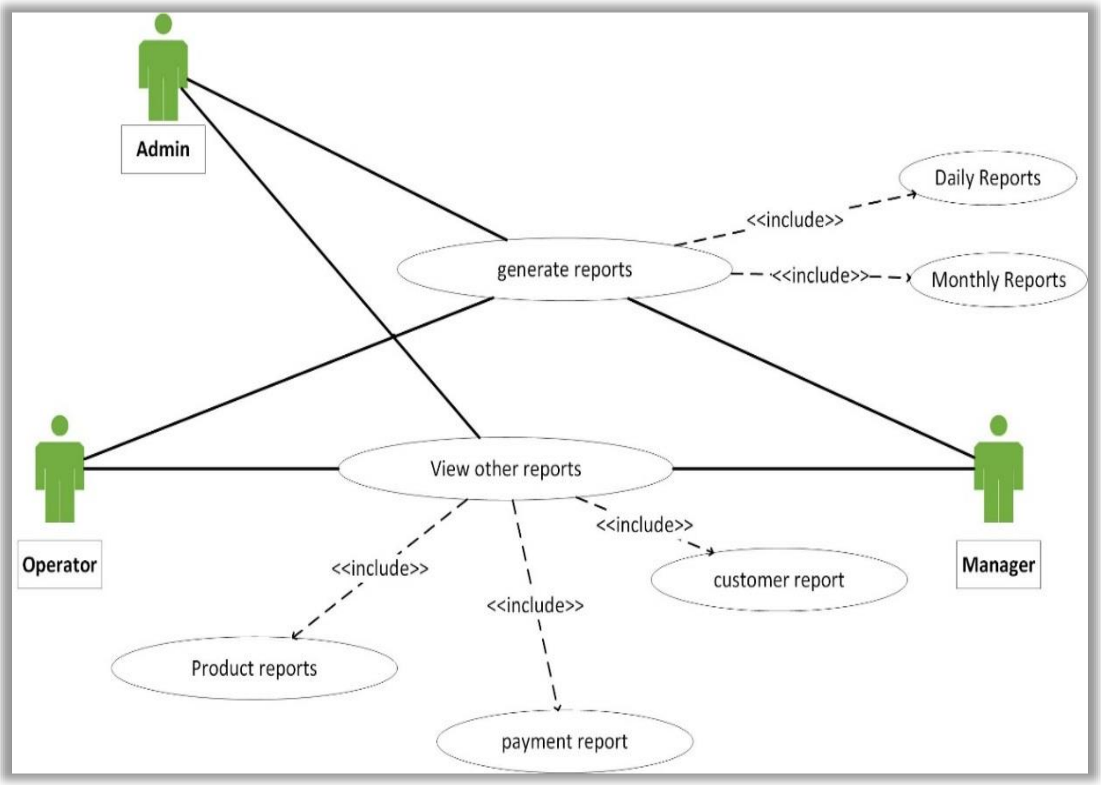


*Figure B.9: Use Case Diagram for Report Management Module*

| | | |
|---|---|---|
| **Use case name** | Generate Reports | |
| **Actors** | Admin/ Manager | |
| **Description** | Generate the reports of the system | |
| **Pre-condition** | System users should be logged into the system | |
| **Typical course of events** | **Actor action** | **System response** |
| | 1. Select the report type | |
| | | 2. System display report generation page |
| | 3. Select date or month | |
| | 4. Click the search button | |
| | | 5. Generate report |
| **Alternative courses** | | |
| **Conclusion** | This use case emphasizes generate the reports | |
| **Post condition** | View report | |

*Table B.11: Use Case Description - Generate Reports*

# Activity Diagrams

## ➢ Activity Diagram for Stock Handling

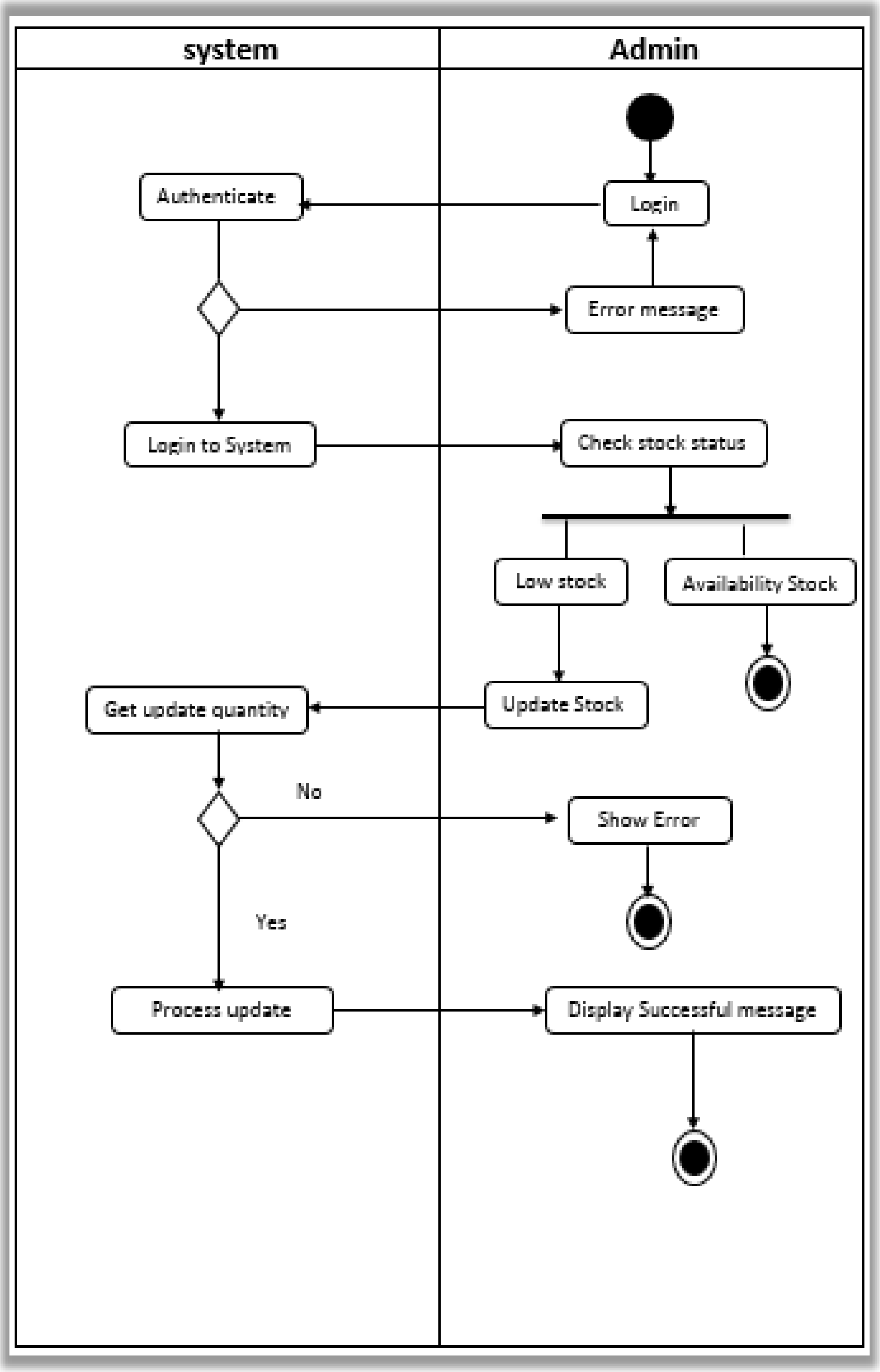


*Figure B.10: Activity Diagram for Stock Handling*

## ➢ Activity Diagram for Product Management

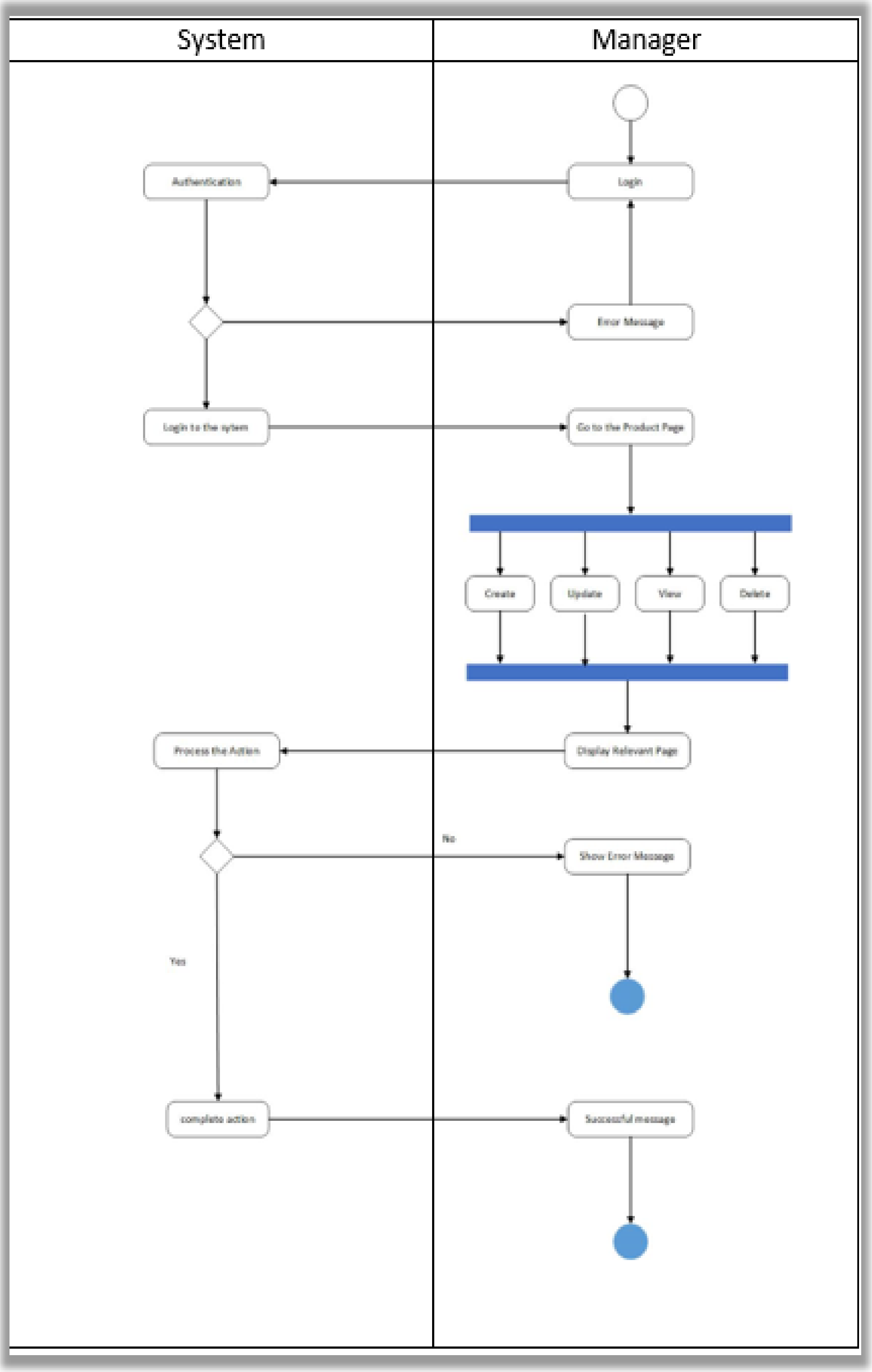


*Figure B.11: Activity Diagram for Product Management Module*

# Sequence Diagrams

## ➢ Sequence Diagram for Order Processing

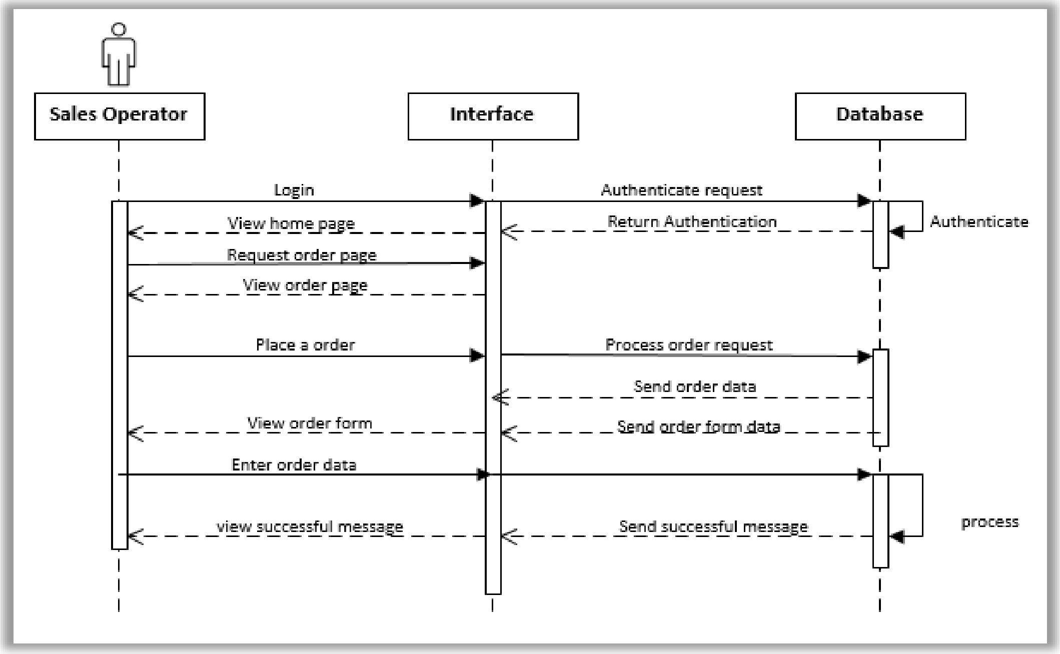


*Figure B.12: Sequence Diagram for Order Processing*

## ➢ Sequence Diagram for Purchase Order

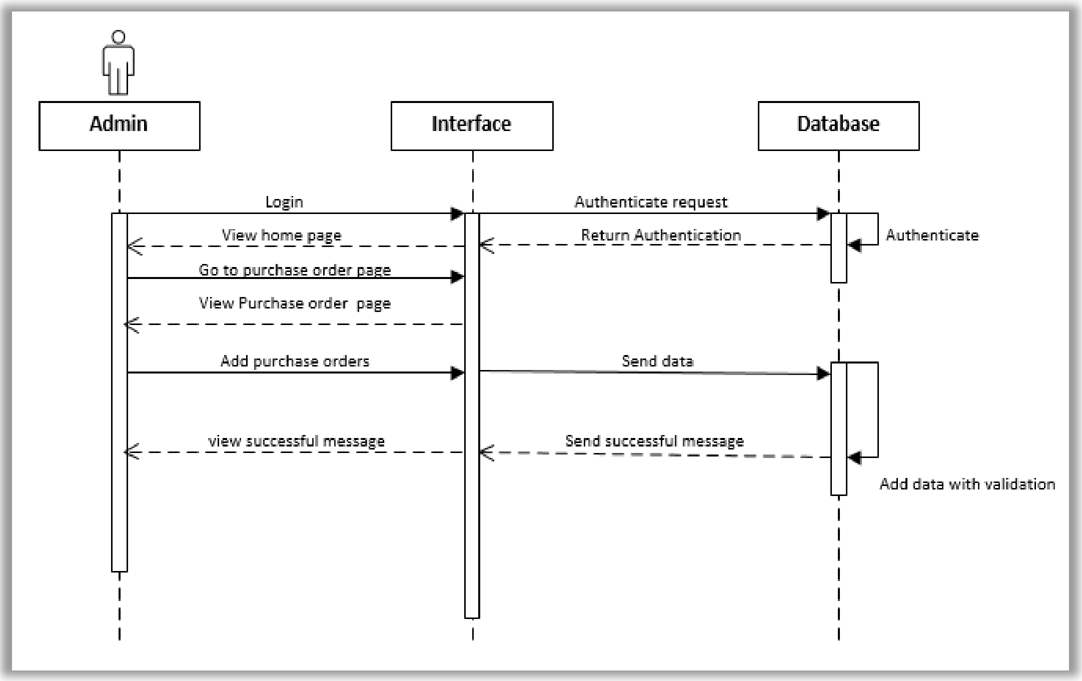
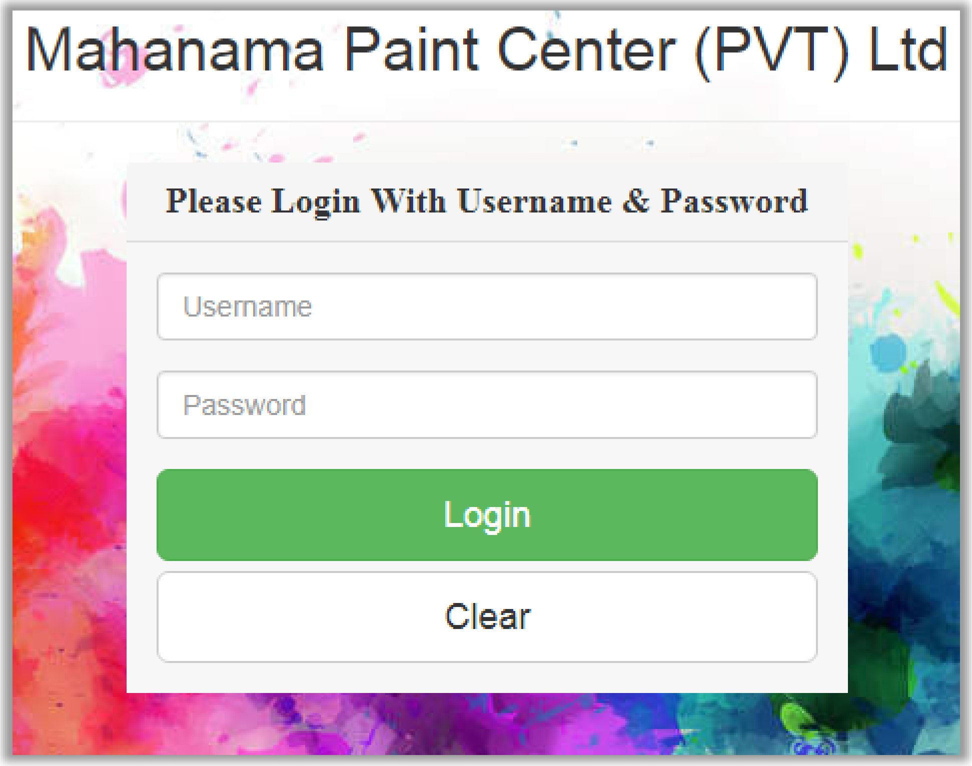


*Figure B.13: Sequence Diagram for Purchase Order*

# Appendix C – User Documentation

## ➢ Login Form

Following figure C.1 shows user login from which allows users to log into the system. All levels of users can log into the system in one form. When user try to log into the system check whether this user is valid user or not otherwise system redirect to again login page.



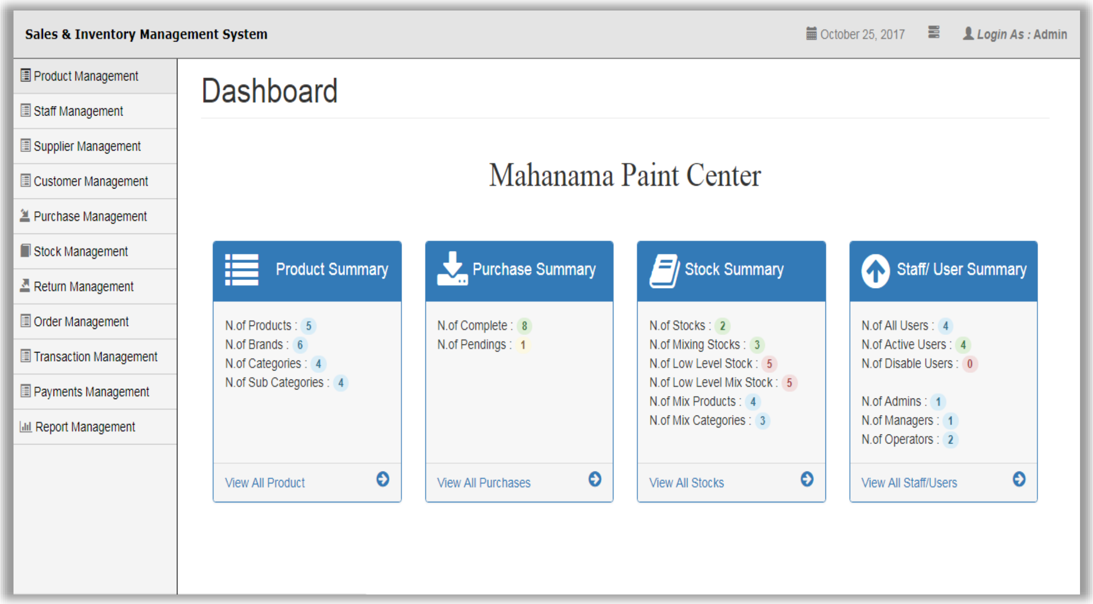
Mahanama Paint Center (PVT) Ltd

Please Login With Username & Password

*Figure C.1: Login Form*

## ➢ Administrator Home Page

All the functions of each and every module can be accessible by users through this main form. Home page view summary of product, purchase, stock and staff/user management modules discretionally.  Following figure C.2 shows how main menu looks like.



Sales & Inventory Management System

October 25, 2017 Login As : Admin

Dashboard

Mahanama Paint Center

Product Summary

N.of Products : 5
N.of Brands : 6
N.of Categories : 4
N.of Sub Categories : 4

Purchase Summary

N.of Complete : 8
N.of Pendings : 1

Stock Summary

N.of Stocks : 2
N.of Mixing Stocks : 3
N.of Low Level Stock : 5
N.of Low Level Mix Stock : 5
N.of Mix Products : 4
N.of Mix Categories : 3

Staff/ User Summary

N.of All Users : 4
N.of Active Users : 4
N.of Disable Users : 0

N.of Admins : 1
N.of Managers : 1
N.of Operators : 2

***Figure C.2: Main Home Page***

- ## Main Menu Items

Figure C.3 shows the main menu item of the main home page

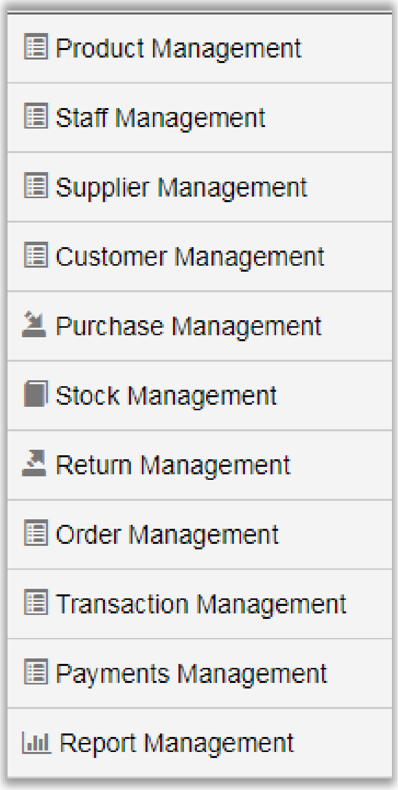
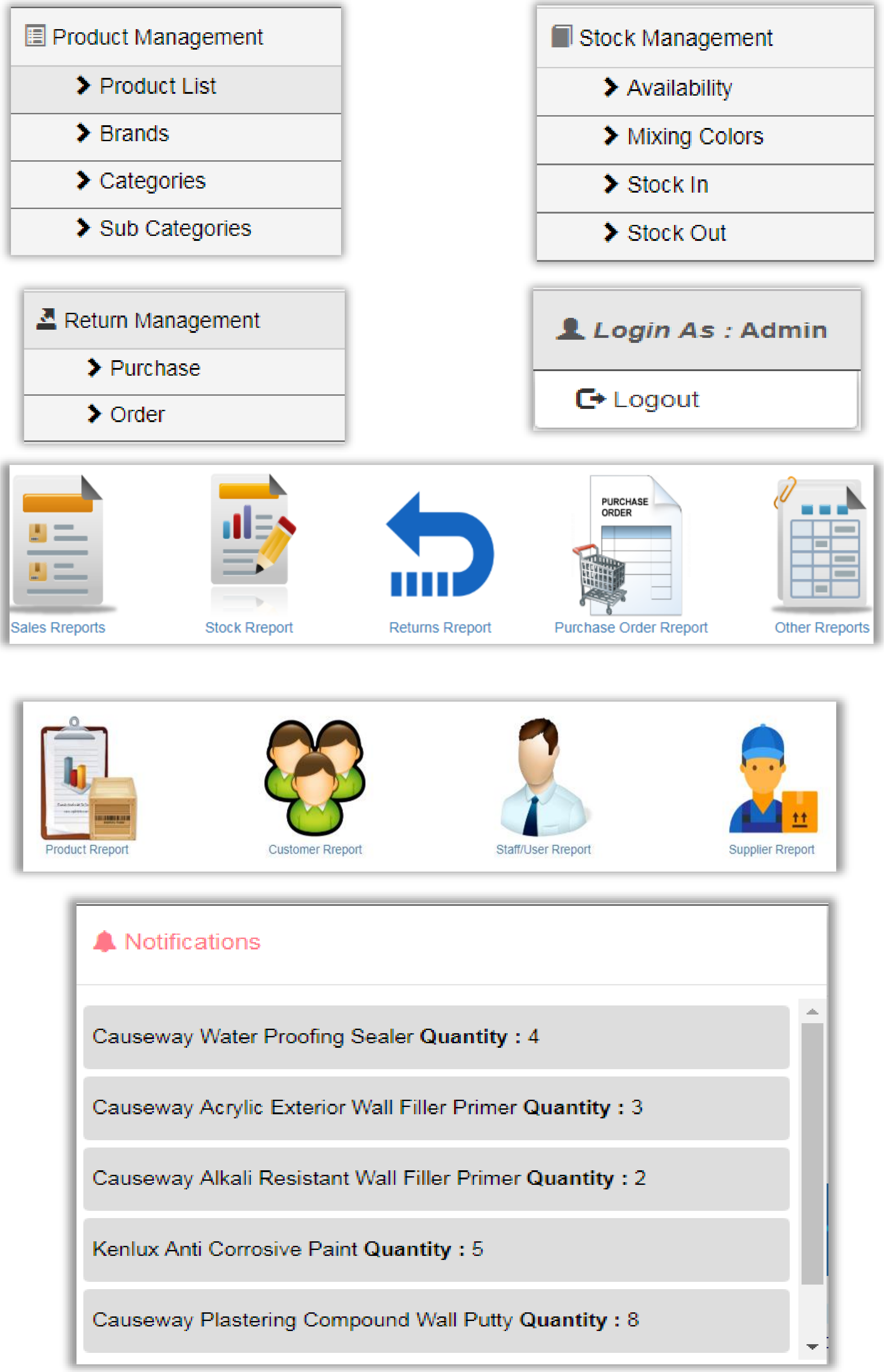


***Figure C.3: Main Menu Item***

- **Submenu Item**

Figure C.4 shows the sub menu items of the home page.



Causeway Water Proofing Sealer **Quantity :** 4

Causeway Acrylic Exterior Wall Filler Primer **Quantity :** 3

Causeway Alkali Resistant Wall Filler Primer **Quantity :** 2

Kenlux Anti Corrosive Paint **Quantity :** 5

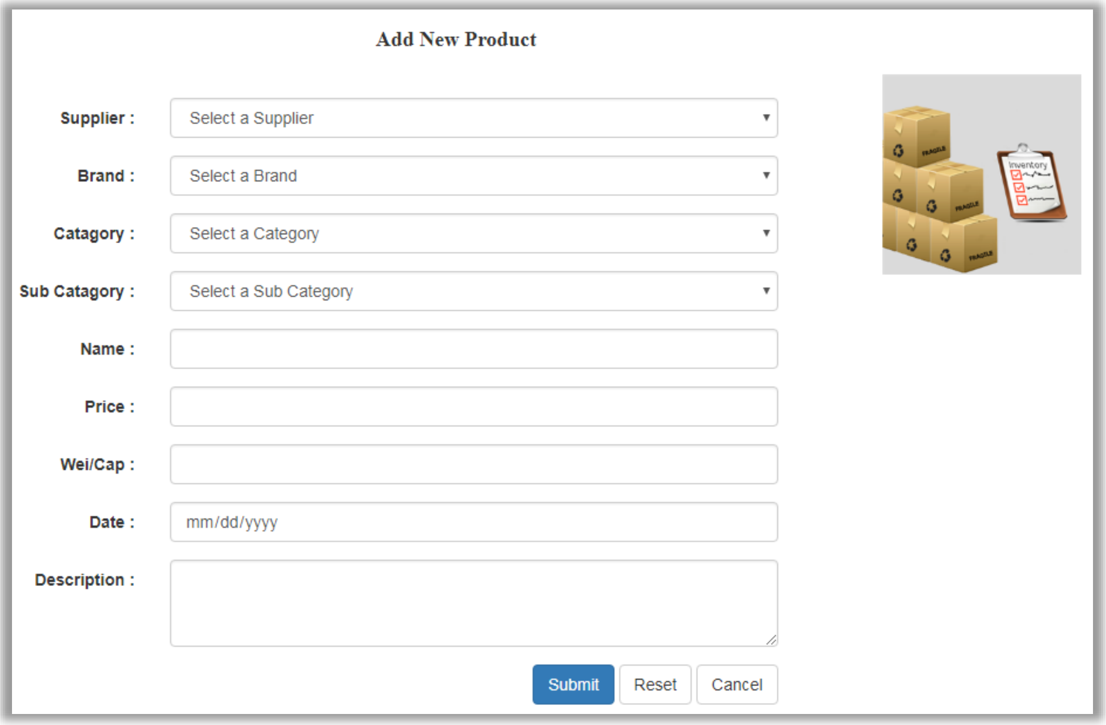
Causeway Plastering Compound Wall Putty **Quantity :** 8

***Figure C.4: Sub Menu Items***

## ➢ Product Registration Form

All the product details can be stored via this form. User can insert, update, delete and view each record individually. Further all the inserted data displayed the table. If user enters invalid data system displays the error message. If user trying to save with empty field system checks and display error message and if user enters existing product name, then system showing its already add or not.

Figure C.5 show how product registration form looks like it.



Add New Product

Supplier : Select a Supplier

Brand : Select a Brand

Catagory : Select a Category

Sub Catagory : Select a Sub Category

Name :

Price :

Wei/Cap :

Date : mm/dd/yyyy

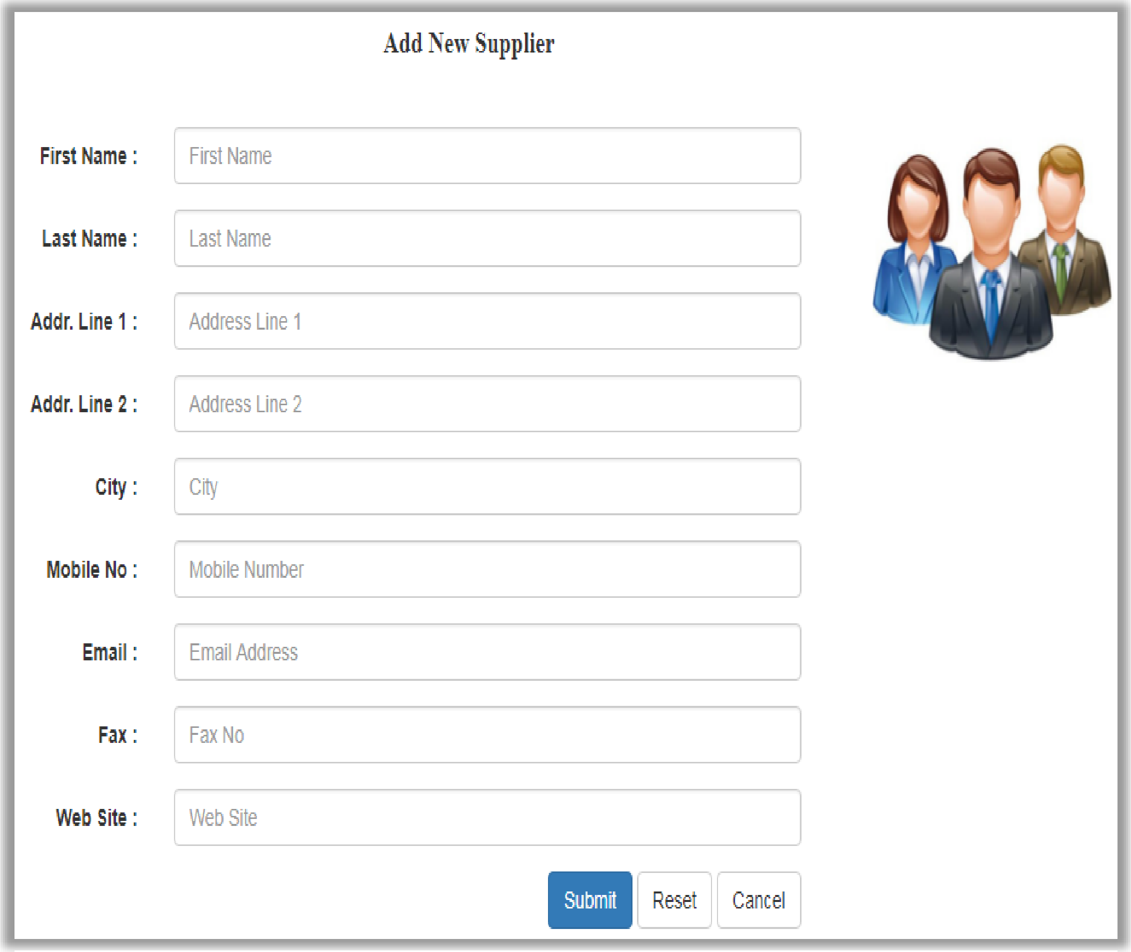
Description :

*Figure C.5: Product Adding Page*

## ➢ Supplier Registration Form

All the supplier details can be stored via this form. User can insert, update, delete and view each record individually. Further all the inserted data displayed the table. If user enters invalid data system displays the error message. If user trying to save with empty field system checks and display error message and if user enters existing supplier name, then system showing its already add or not.

Figure C.6 show how supplier registration form looks like it.



Add New Supplier

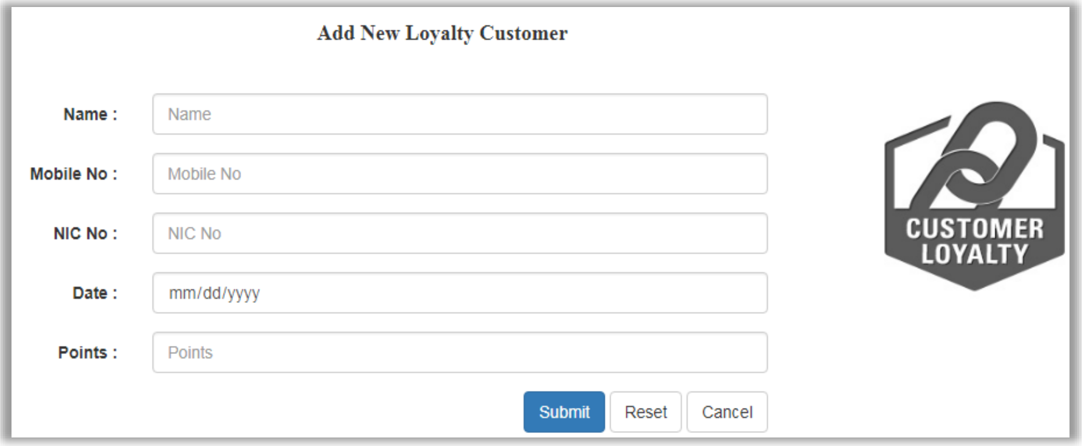
| | |
|---|---|
| First Name : | First Name |
| Last Name : | Last Name |
| Addr. Line 1 : | Address Line 1 |
| Addr. Line 2 : | Address Line 2 |
| City : | City |
| Mobile No : | Mobile Number |
| Email : | Email Address |
| Fax : | Fax No |
| Web Site : | Web Site |

*Figure C.6: Supplier Adding Page*

## ➢ Customer Registration Form

All the loyalty customer details can be stored via this form. User can insert, update, delete and view each record individually. Further all the inserted data displayed the table. If user enters invalid data system displays the error message. If user trying to save with empty field system checks and display error message and if user enters existing NIC number, then system showing its already add or not.

Figure C.7 show how loyalty customer register form looks like it.



Add New Loyalty Customer

Name : Name

Mobile No : Mobile No

NIC No : NIC No

Date : mm/dd/yyyy

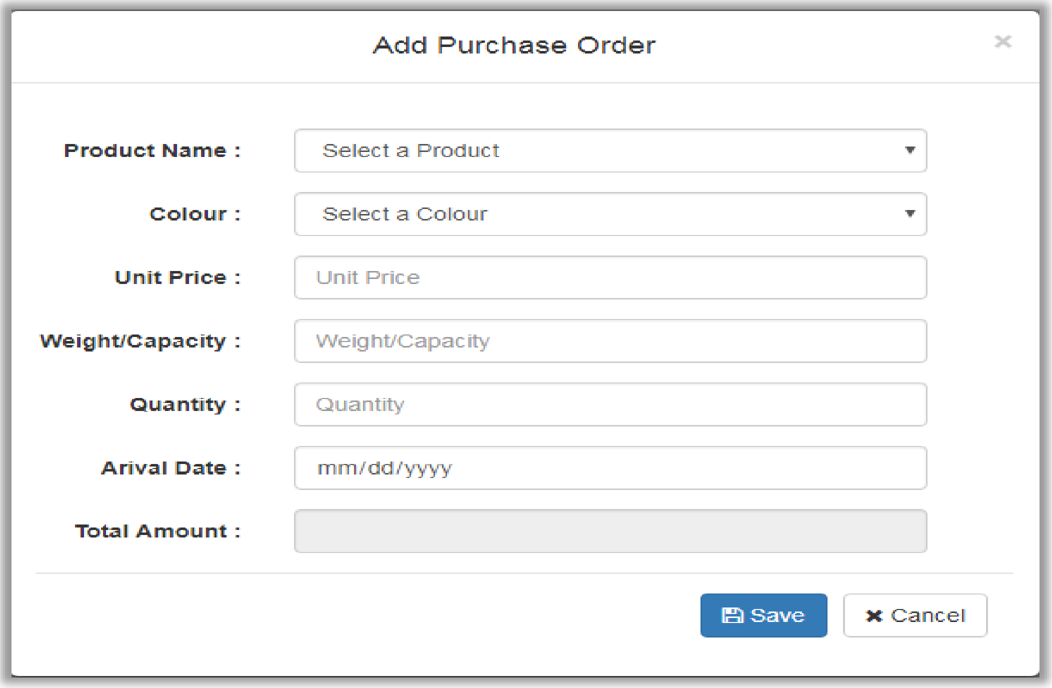
Points : Points

*Figure C.7: Customer Registration Form*

## ➢ Purchase Order Form

All the purchase orders are place using this form. User can insert, update, delete and view individually. Overall records. If user place the purchase order it want to update states field to ‘complete’ then its quantity is update the stock. If user trying to save with empty field system checks and display error message.

Figure C.8 show how Purchase order form looks like it.



Add Purchase Order

Product Name : Select a Product

Colour : Select a Colour

Unit Price : Unit Price

Weight/Capacity : Weight/Capacity

Quantity : Quantity

Arival Date : mm/dd/yyyy

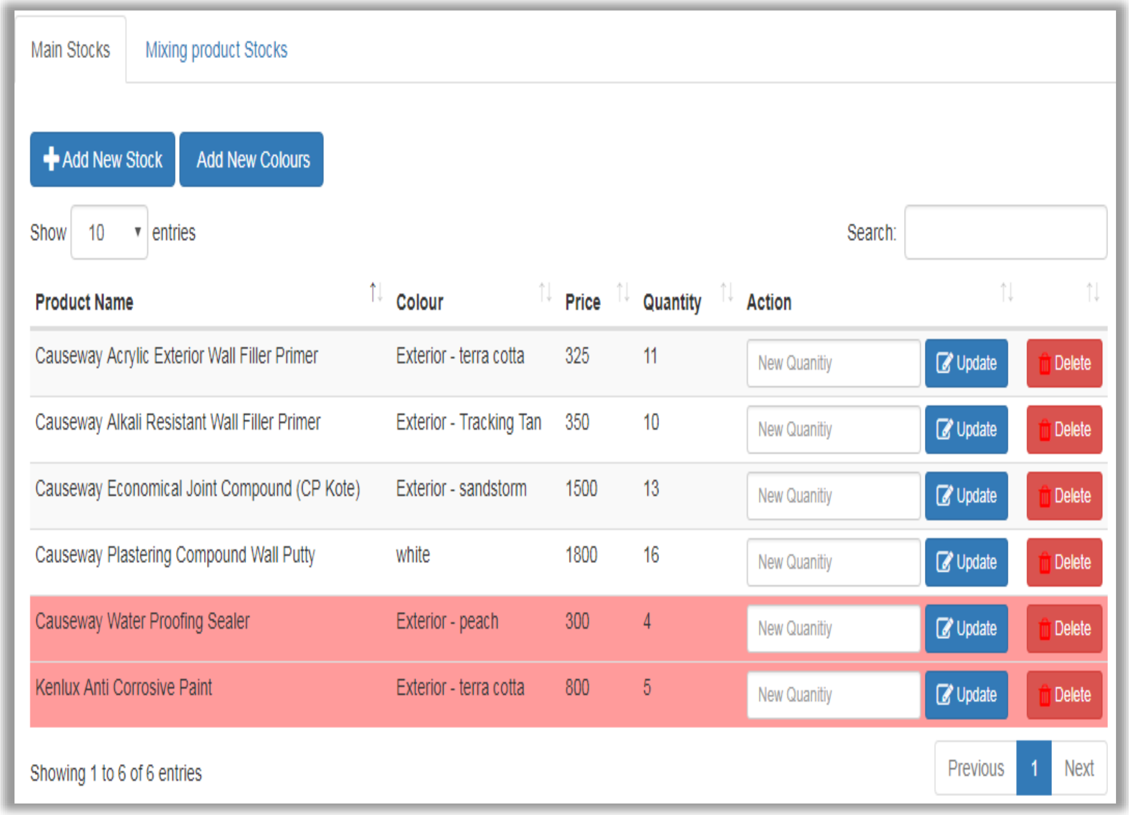
Total Amount :

*Figure C.8: Purchase Order Form*

## ➢ Stock Update Form

All the stocks are update using this form. Low level stock quantities showing highlighted rows. User can insert new stock, update stock level, search stocks. User can’t update stock quantity without value.

Figure C.9 show how the stock update form.



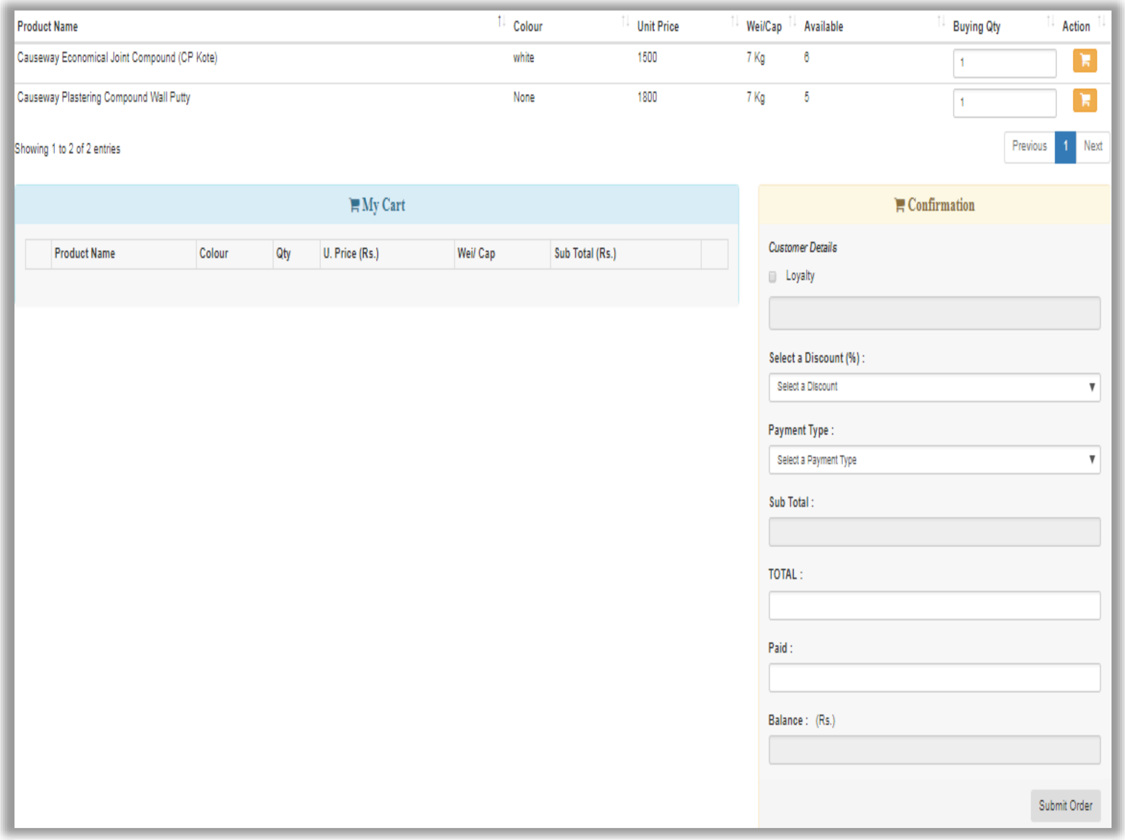
| Product Name | Colour | Price | Quantity | Action | |
|---|---|---|---|---|---|
| Causeway Acrylic Exterior Wall Filler Primer | Exterior - terra cotta | 325 | 11 | New Quantity Update | Delete |
| Causeway Alkali Resistant Wall Filler Primer | Exterior - Tracking Tan | 350 | 10 | New Quantity Update | Delete |
| Causeway Economical Joint Compound (CP Kote) | Exterior - sandstorm | 1500 | 13 | New Quantity Update | Delete |
| Causeway Plastering Compound Wall Putty | white | 1800 | 16 | New Quantity Update | Delete |
| Causeway Water Proofing Sealer | Exterior - peach | 300 | 4 | New Quantity Update | Delete |
| Kenlux Anti Corrosive Paint | Exterior - terra cotta | 800 | 5 | New Quantity Update | Delete |

*Figure C.9: Stock Update Form*

## ➢ Order Processing Form

All the customer orders place in this form. User can insert, update quantity’s. user can’t add ordered table without selecting product.

Figure C.10 show how the order processing form



| Product Name | Colour | Unit Price | Wei/Cap | Available | Buying Qty | Action |
|---|---|---|---|---|---|---|
| Causeway Economical Joint Compound (CP Kote) | white | 1500 | 7 Kg | 6 | 1 | |
| Causeway Plastering Compound Wall Putty | None | 1800 | 7 Kg | 5 | 1 | |

Showing 1 to 2 of 2 entries

My Cart

| Product Name | Colour | Qty | U. Price (Rs.) | Wei/ Cap | Sub Total (Rs.) |
|---|---|---|---|---|---|

Confirmation

Customer Details

□ Loyalty

Select a Discount (%) : Select a Discount

Payment Type : Select a Payment Type

Sub Total :

TOTAL :

Paid :

Balance : (Rs.)

*Figure C.10: Order Processing Form*

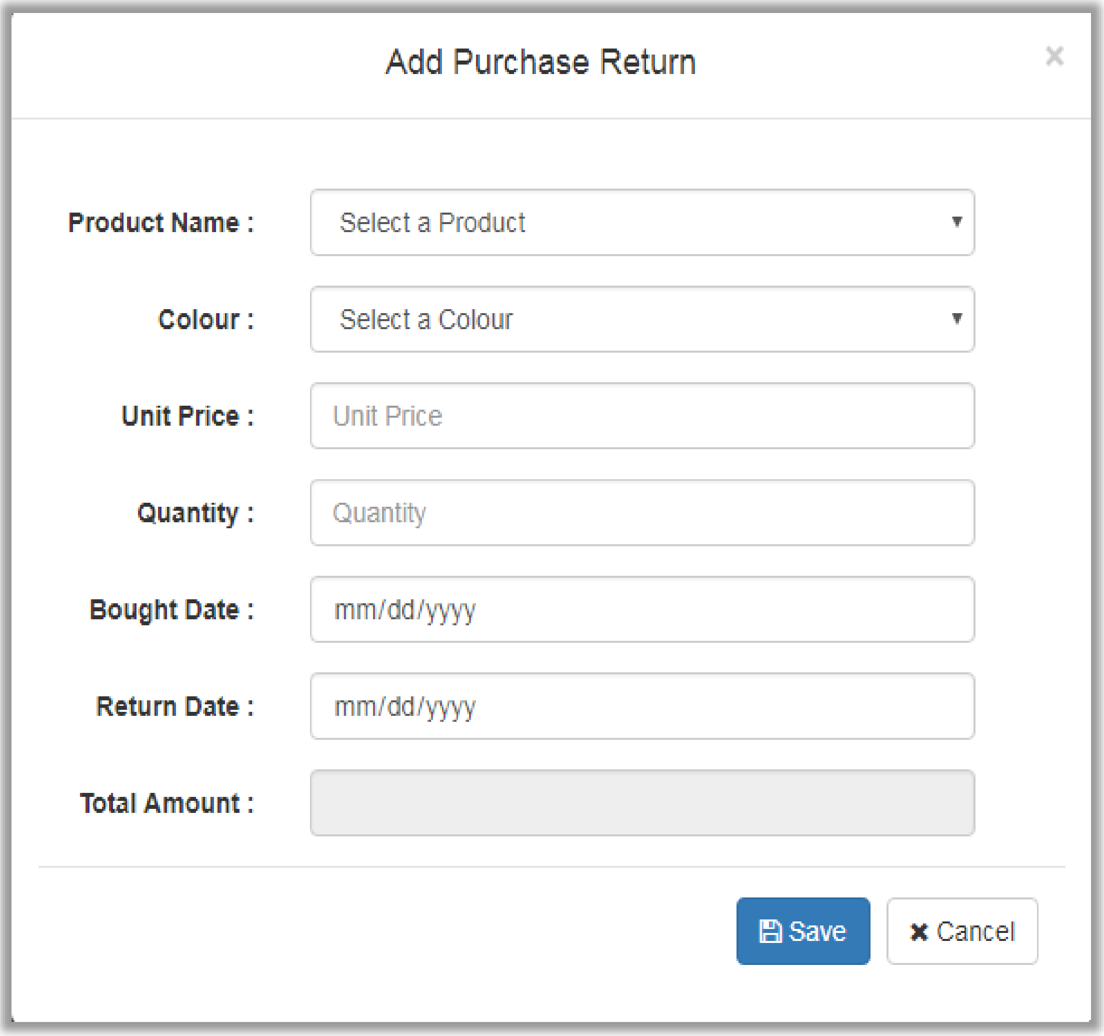
## ➢ Return Management

All the product returns are divided into two parts. Its showing below.

### • Purchase Return Form

All the purchased product return details are stored using this form. User can insert purchase returns with stock updating, delete, search and view individually. If user trying to save with empty field system checks and display error message.

Figure C.11 showing how the purchase return form



Add Purchase Return

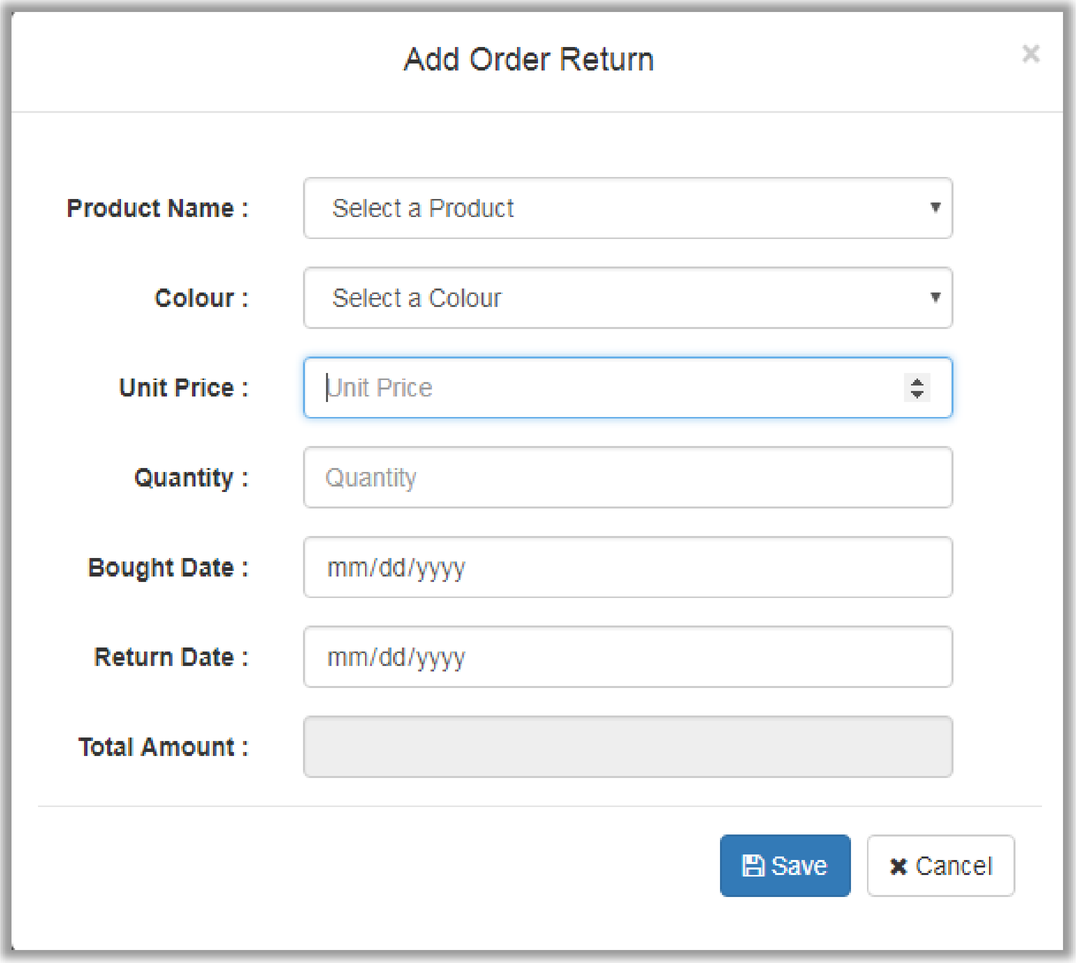
| | |
|---|---|
| Product Name : | Select a Product ▾ |
| Colour : | Select a Colour ▾ |
| Unit Price : | Unit Price |
| Quantity : | Quantity |
| Bought Date : | mm/dd/yyyy |
| Return Date : | mm/dd/yyyy |
| Total Amount : | |

*Figure C.11: Purchase Return Form*

## • Order Return Form

All the ordered product return details are stored using this form. User can insert order returns with stock updating, delete, search and view individually. If user trying to save with empty field system checks and display error message.

Figure C.12 showing how the order return form



Add Order Return

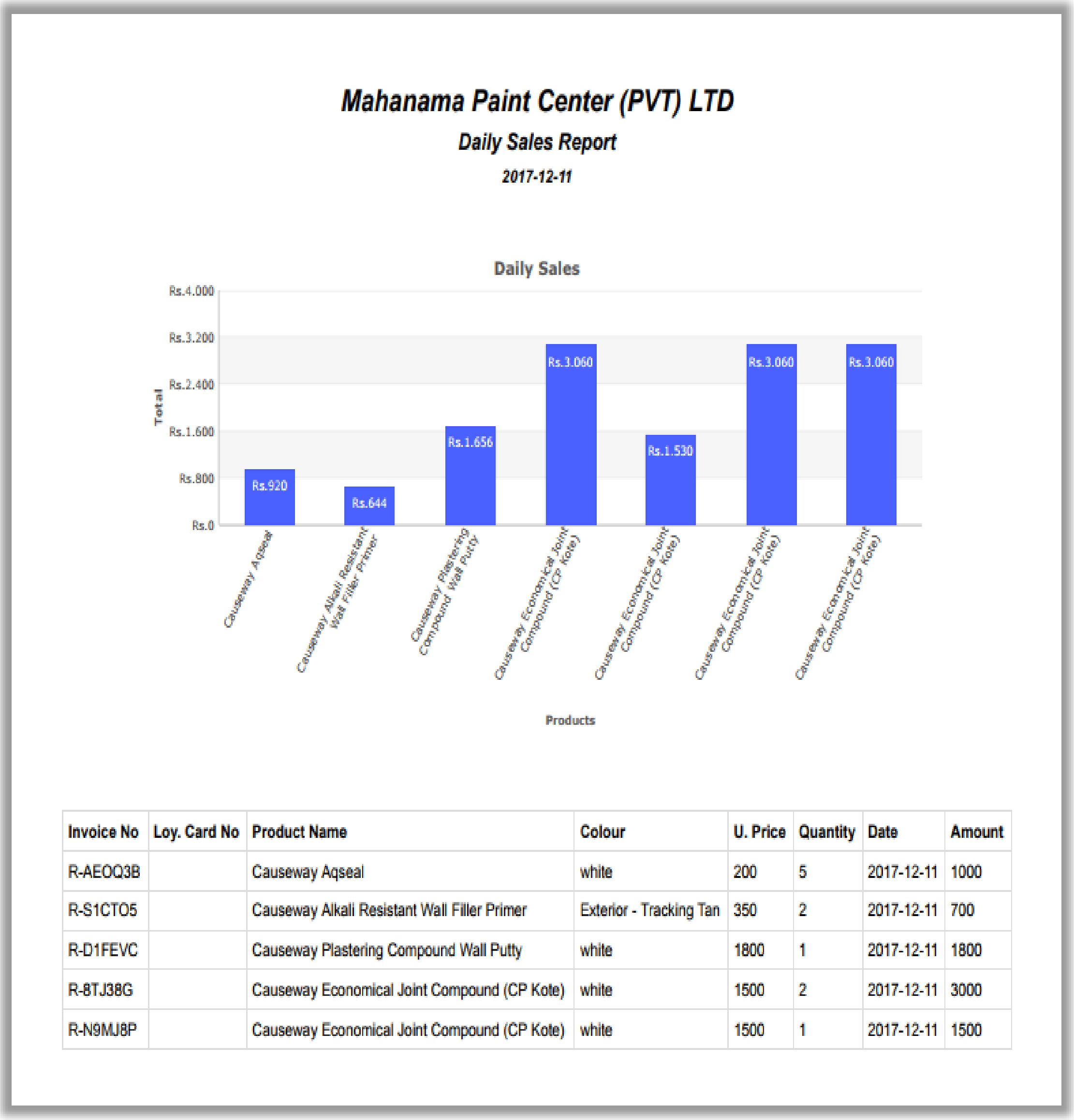
| | |
|---|---|
| Product Name : | Select a Product |
| Colour : | Select a Colour |
| Unit Price : | Unit Price |
| Quantity : | Quantity |
| Bought Date : | mm/dd/yyyy |
| Return Date : | mm/dd/yyyy |
| Total Amount : | |

*Figure C.12: Order Return Form*

# Appendix D – Management Report

## Sales Reports

Sales of ta particular time period (daily, monthly) can be identified by the management with this report. Daily report and monthly report of the shop are displayed under figure D.1 and figure D.2 respectively.



Mahanama Paint Center (PVT) LTD

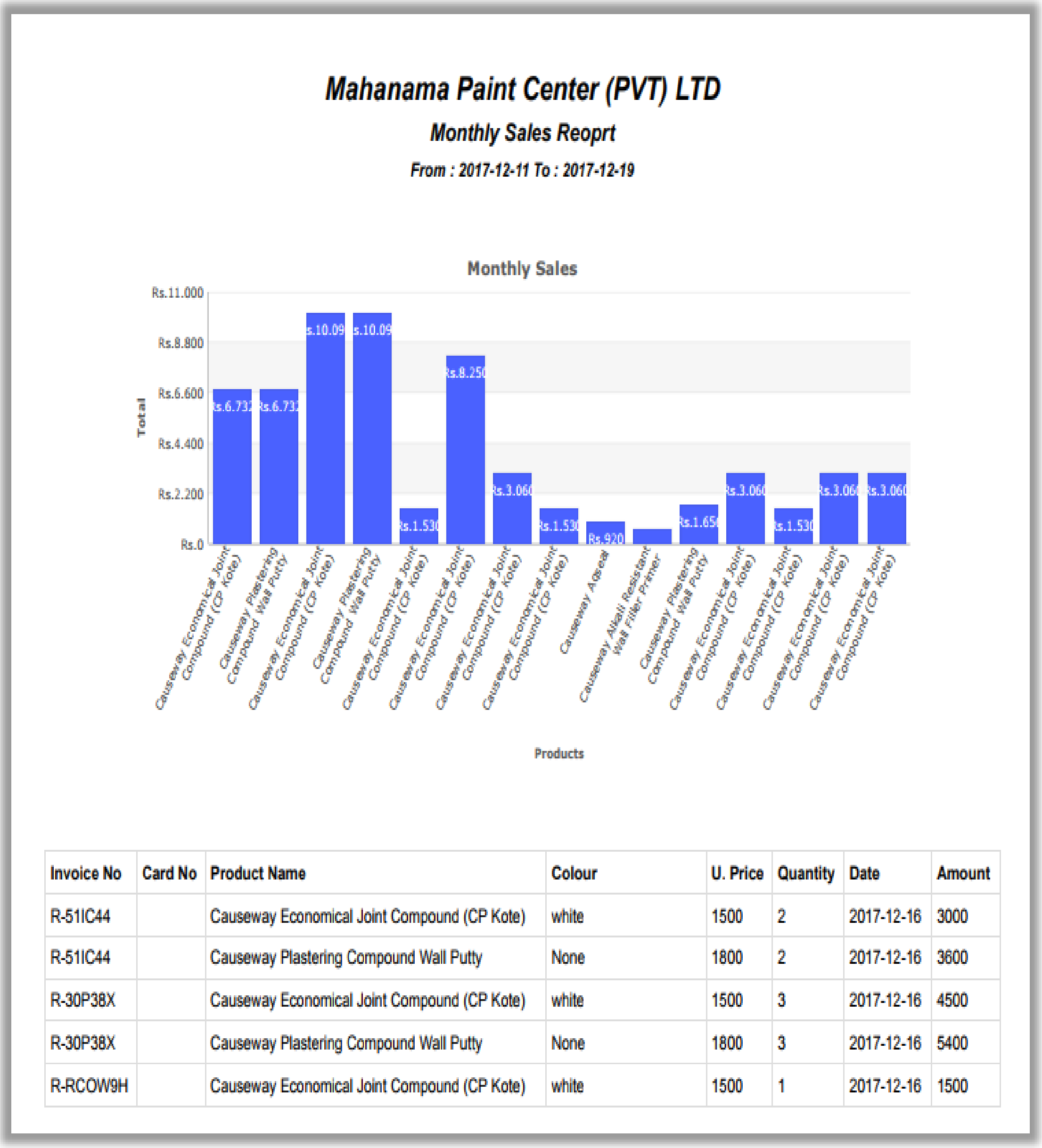
Daily Sales Report

2017-12-11

Daily Sales
Total
Rs.4.000
Rs.3.200
Rs.2.400
Rs.1.600
Rs.800
Rs.0
Rs.920
Rs.644
Rs.1.656
Rs.3.060
Rs.1.530
Rs.3.060
Rs.3.060
Causeway Aqseal
Causeway Alkali Resistant Wall Filler Primer
Causeway Plastering Compound Wall Putty
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Products

| Invoice No | Loy. Card No | Product Name | Colour | U. Price | Quantity | Date | Amount |
|---|---|---|---|---|---|---|---|
| R-AEOQ3B | | Causeway Aqseal | white | 200 | 5 | 2017-12-11 | 1000 |
| R-S1CTO5 | | Causeway Alkali Resistant Wall Filler Primer | Exterior - Tracking Tan | 350 | 2 | 2017-12-11 | 700 |
| R-D1FEVC | | Causeway Plastering Compound Wall Putty | white | 1800 | 1 | 2017-12-11 | 1800 |
| R-8TJ38G | | Causeway Economical Joint Compound (CP Kote) | white | 1500 | 2 | 2017-12-11 | 3000 |
| R-N9MJ8P | | Causeway Economical Joint Compound (CP Kote) | white | 1500 | 1 | 2017-12-11 | 1500 |

*Figure D.1: Daily Report*



## Mahanama Paint Center (PVT) LTD

**Monthly Sales Reoprt**

**From : 2017-12-11 To : 2017-12-19**

Monthly Sales
Total
Rs.11.000
Rs.8.800
Rs.6.600
Rs.4.400
Rs.2.200
Rs.0
Rs.6.732
Rs.6.732
s.10.09
s.10.09
Rs.1.530
Rs.8.250
Rs.3.060
Rs.1.530
Rs.920
Rs.1.650
Rs.3.060
Rs.1.530
Rs.3.060
Rs.3.060
Causeway Economical Joint Compound (CP Kote)
Causeway Plastering Compound Wall Putty
Causeway Economical Joint Compound (CP Kote)
Causeway Plastering Compound Wall Putty
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Aqseal
Causeway Alkali Resistant Wall Filler Primer
Causeway Plastering Compound Wall Putty
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Causeway Economical Joint Compound (CP Kote)
Products

| Invoice No | Card No | Product Name | Colour | U. Price | Quantity | Date | Amount |
|---|---|---|---|---|---|---|---|
| R-51IC44 | | Causeway Economical Joint Compound (CP Kote) | white | 1500 | 2 | 2017-12-16 | 3000 |
| R-51IC44 | | Causeway Plastering Compound Wall Putty | None | 1800 | 2 | 2017-12-16 | 3600 |
| R-30P38X | | Causeway Economical Joint Compound (CP Kote) | white | 1500 | 3 | 2017-12-16 | 4500 |
| R-30P38X | | Causeway Plastering Compound Wall Putty | None | 1800 | 3 | 2017-12-16 | 5400 |
| R-RCOW9H | | Causeway Economical Joint Compound (CP Kote) | white | 1500 | 1 | 2017-12-16 | 1500 |

*Figure D.2: Monthly Sales Report*

# Appendix E – Test Results

Major test cases at the evaluation stage along with test results are contained in this chapter.

## Test Results for User Authentication

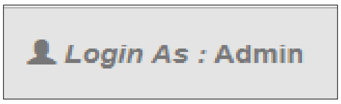
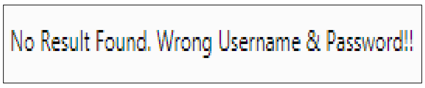
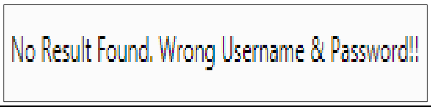
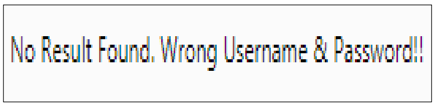
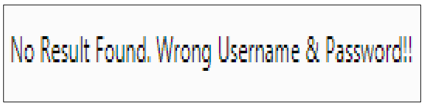
| Test No | Steps to test | Expected result | Status |
|---|---|---|---|
| 1 | Enter correct username<br>Enter correct password<br>Enable account status | Successfully Login into the system.<br><br>Login As : Admin | ✓ |
| 2 | Enter correct username<br>Enter incorrect password<br>Enable account status | Display error message “No Record Found. Wrong Username or Password”<br><br>No Result Found. Wrong Username & Password!! | ✓ |
| 3 | Enter incorrect username<br>Enter correct password<br>Enable account status | Display error message “No Record Found. Wrong Username or Password”<br><br>No Result Found. Wrong Username & Password!! | ✓ |
| 4 | Enter both username & password incorrect<br>Enable account status | Display error message “No Record Found. Wrong Username or Password”<br><br>No Result Found. Wrong Username & Password!! | ✓ |
| 5 | Enter correct username<br>Enter correct password<br>Enable account status disable | Display error message “No Record Found. Wrong Username or Password”<br><br>No Result Found. Wrong Username & Password!! | ✓ |

*Figure E.1: User Authentication Test Result*

## Test Results for Product Management

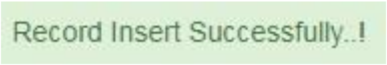
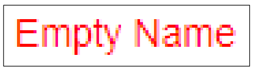
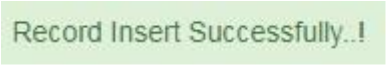
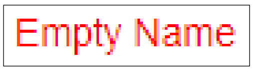
| Test No | Steps to test | Expected result | status |
|---|---|---|---|
| 1 | Add information with relevant details | Display message “Record Insert Successfully”<br><br>Record Insert Successfully..! | ✓ |
| 2 | Try to add information by keeping all fields in empty | Display message “Empty First Field Name”<br><br>Empty Name | ✓ |
| 3 | Enter all fields without select supplier | Display message “Select a Supplier”<br> <br>Please Select a Supplier | ✓ |
| 4 | Add Product for already existing Product Name | Display message “Already added”<br> <br>Already Added | ✓ |
| 5 | Choose Product from the table, click view details and update details | Display message “Record Update Successfully”<br> <br>Record Update Successfully..! | ✓ |
| 6 | Choose Product from the table, click delete icon | Popup the confirm dialog box “Do you want to Delete product?”<br> <br>Do You want to Delete Product ? : | ✓ |

*Figure E.2: Manage Product Test Results*

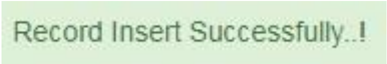
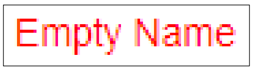
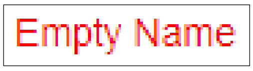
## Test Results for Supplier Management

| Test No | Steps to test | Expected result | Status |
|---|---|---|---|
| 1 | Add information with relevant details | Display message “Record Insert Successfully”<br><br>Record Insert Successfully..! | ✓ |
| 2 | Try to add information by keeping all fields in empty | Display message “Empty First Field Name”<br><br>Empty Name | ✓ |

| | | | |
|---|---|---|---|
| 3 | Add Supplier for already existing Supplier Name | Display message “This supplier Already added”<br> <br>This supplier Already Added | ✓ |
| 4 | Choose Supplier from the table, click view details and update details | Display message “Record Update Successfully”<br> <br>Record Update Successfully..! | ✓ |
| 5 | Choose Supplier from the table, click delete icon | Popup the confirm dialog box “Do you want to Delete Supplier?”<br> <br>Do You want to Delete Supplier ? : | ✓ |

***Figure E.3: Manage Supplier Test Results***

## Test Results for Customer Management

| Test No | Steps to test | Expected result | status |
|---|---|---|---|
| 1 | Add information with relevant details | Display message “Record Insert Successfully”<br><br>Record Insert Successfully..! | ✓ |
| 2 | Try to add information by keeping all fields in empty | Display message “Empty First Field Name”<br><br>Empty Name | ✓ |
| 3 | Enter all fields without First Field | Display message “Empty First Name”<br><br>Empty Name | ✓ |
| 4 | Add Loyalty Customer for already existing Loyalty Customer NIC | Display message “This customer Already added”<br> <br>This Customer Already Added | ✓ |
| 5 | Choose Loyalty Customer from the table, click view details and update details | Display message “Record Update Successfully”<br> <br>Record Update Successfully..! | ✓ |

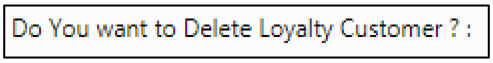
| | | | |
|---|---|---|---|
| 6 | Choose Loyalty Customer from the table, click delete icon | Popup the confirm dialog box "Do you want to Delete Loyalty Customer?"<br><br>Do You want to Delete Loyalty Customer ? : | ✓ |

*Figure E.4: Manage Customer Test Results*

## Test Results for Purchase Order Management

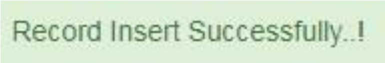
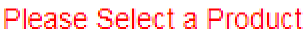
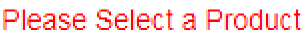
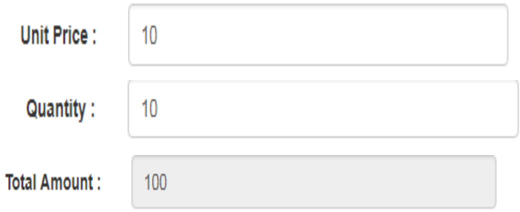
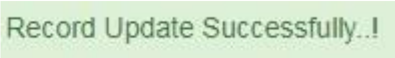
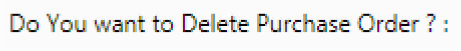
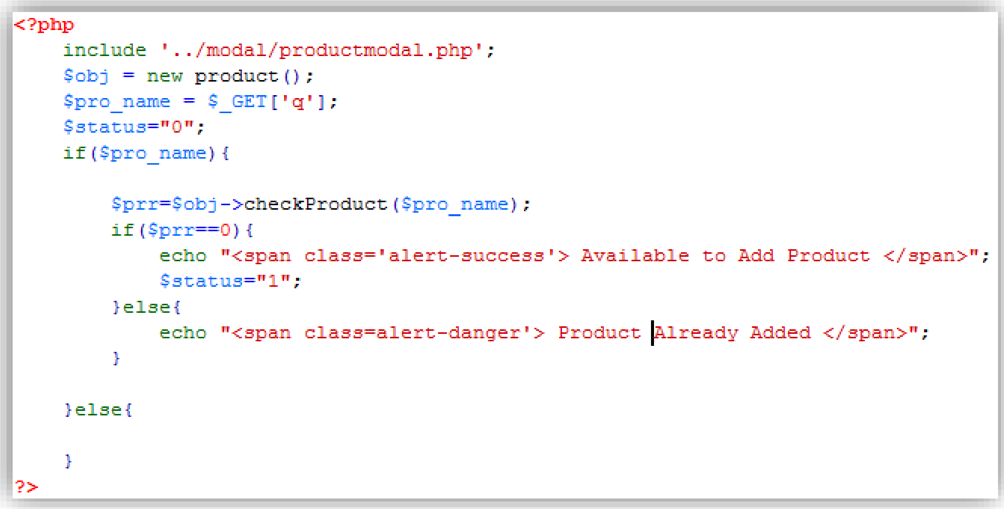
| Test No | Steps to test | Expected result | status |
|---|---|---|---|
| 1 | Add information with relevant details | Display message "Record Insert Successfully"<br><br>Record Insert Successfully..! | ✓ |
| 2 | Try to add information by keeping all fields in empty | Display message "Empty First Field Name"<br><br>Please Select a Product | ✓ |
| 3 | Enter all fields without select Product | Display message "Select a Product"<br><br>Please Select a Product | ✓ |
| 4 | Add unit price & quantity | Display Total last text field<br><br>Unit Price : 10<br>Quantity : 10<br>Total Amount : 100 | ✓ |
| 5 | Choose Purchase Order from the table, click view details and update details | Display message "Record Update Successfully"<br><br>Record Update Successfully..! | ✓ |
| 6 | Choose Purchase Order from the table, click delete icon | Popup the confirm dialog box "Do you want to Delete Purchase Order?"<br><br>Do You want to Delete Purchase Order ? : | ✓ |

*Figure E.5: Purchase Order Test Results*

# Appendix F – Code Listing

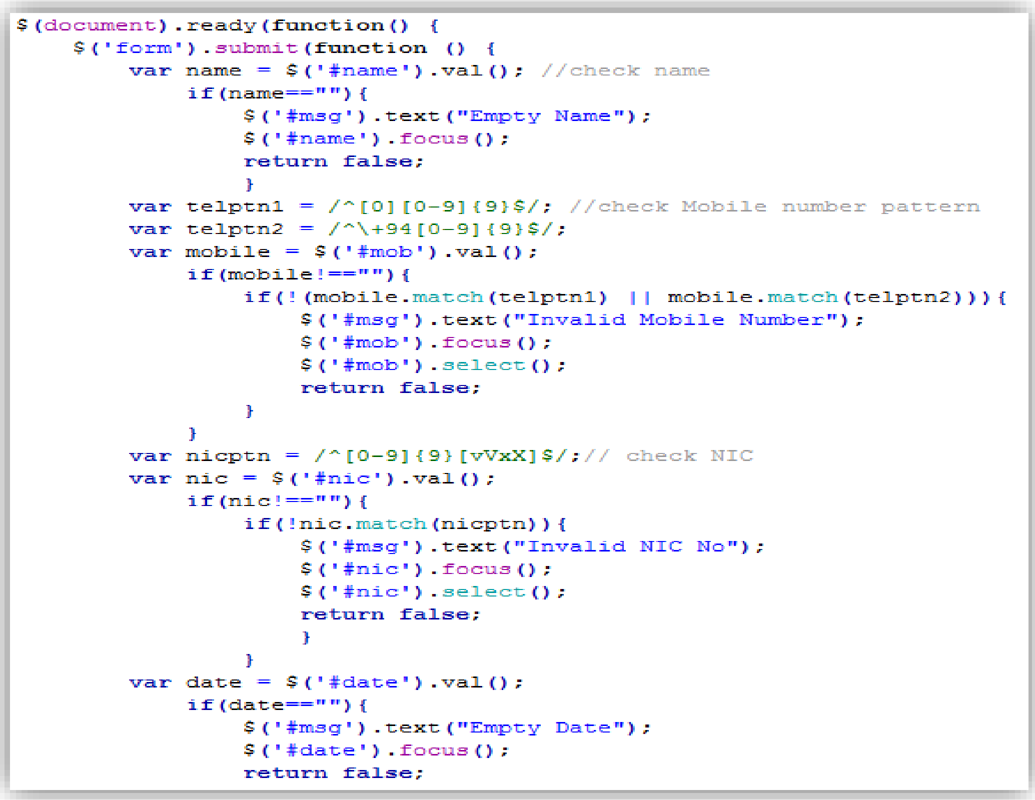
Major code fragments for anyone who interested in refreshing functionality of the system are contained in this document. Please refer the CD-ROM for see the complete codes for all code fragments.

### Adding Products Name Validation by Ajax



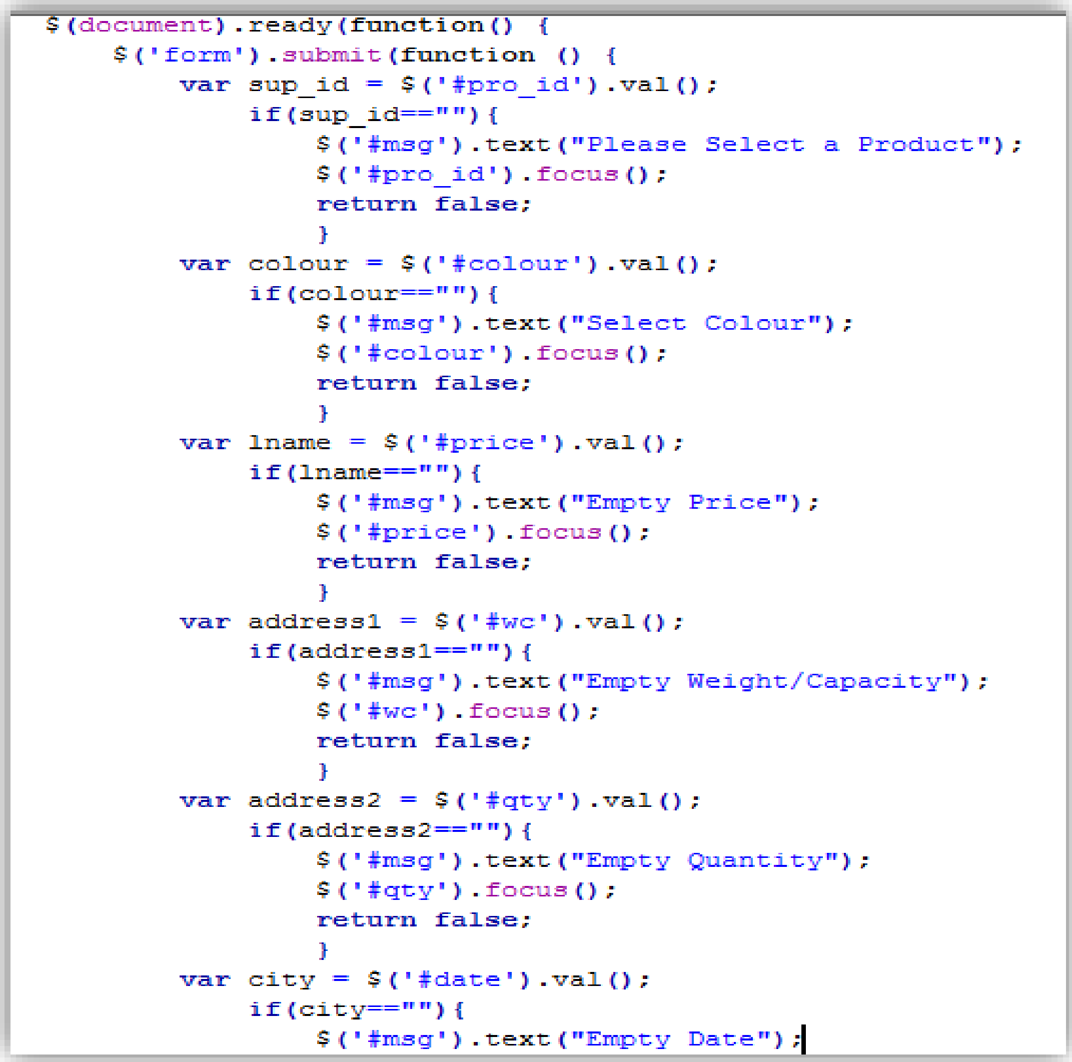
```
<?php
    include '../modal/productmodal.php';
    $obj = new product();
    $pro_name = $_GET['q'];
    $status="0";
    if($pro_name){

        $prr=$obj->checkProduct($pro_name);
        if($prr==0){
            echo "<span class='alert-success'> Available to Add Product </span>";
            $status="1";
        }else{
            echo "<span class=alert-danger'> Product Already Added </span>";
        }

    }else{

    }
?>
```

### Adding Customer Form Validation



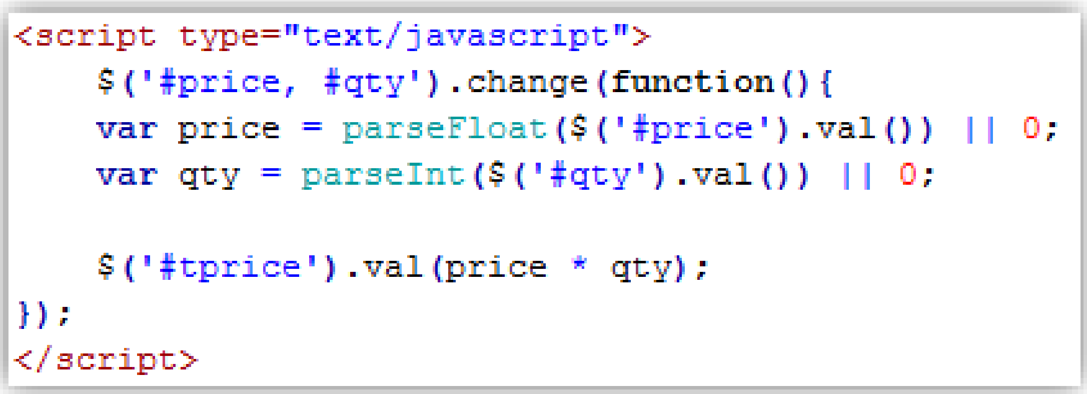
```
$(document).ready(function()  {
    $('form').submit(function  ()  {
        var name = $('#name').val(); //check name
            if(name==""){
                $('#msg').text("Empty Name");
                $('#name').focus();
                return false;
                }
        var telptn1 = /^[0][0-9]{9}$/; //check Mobile number pattern
        var telptn2 = /^\+94[0-9]{9}$/;
        var mobile = $('#mob').val();
            if(mobile!==""){
                if(!(mobile.match(telptn1) || mobile.match(telptn2))){
                    $('#msg').text("Invalid Mobile Number");
                    $('#mob').focus();
                    $('#mob').select();
                    return false;
                }
            }
        var nicptn = /^[0-9]{9}[vVxX]$/;// check NIC
        var nic = $('#nic').val();
            if(nic!==""){
                if(!nic.match(nicptn)){
                    $('#msg').text("Invalid NIC No");
                    $('#nic').focus();
                    $('#nic').select();
                    return false;
                    }
                }
        var date = $('#date').val();
            if(date==""){
                $('#msg').text("Empty Date");
                $('#date').focus();
                return false;
```

### Adding Purchase Order Form Validation



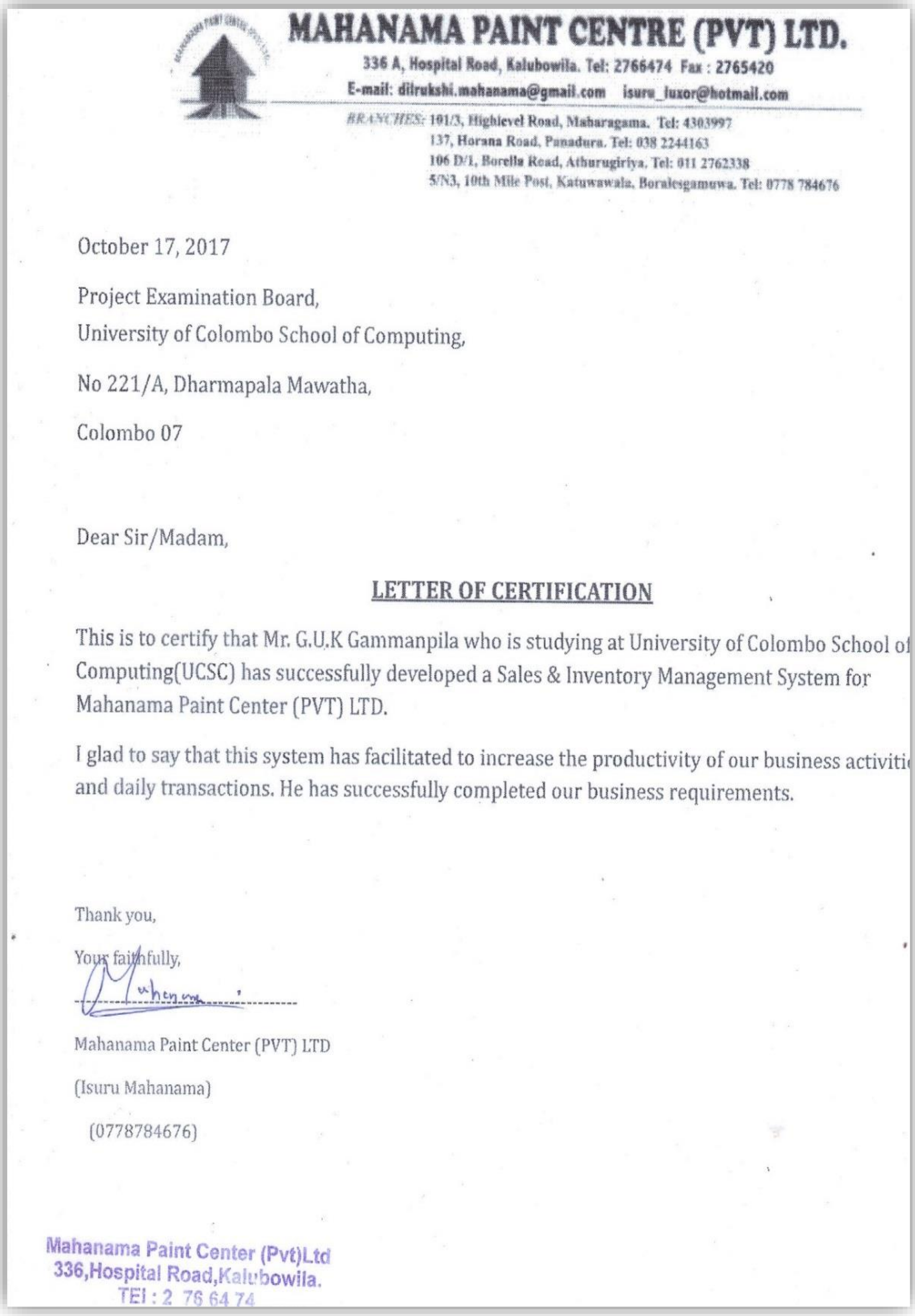
```
$(document).ready(function() {
    $('form').submit(function () {
        var sup_id = $('#pro_id').val();
            if(sup_id==""){
                $('#msg').text("Please Select a Product");
                $('#pro_id').focus();
                return false;
                }
        var colour = $('#colour').val();
            if(colour==""){
                $('#msg').text("Select Colour");
                $('#colour').focus();
                return false;
                }
        var lname = $('#price').val();
            if(lname==""){
                $('#msg').text("Empty Price");
                $('#price').focus();
                return false;
                }
        var address1 = $('#wc').val();
            if(address1==""){
                $('#msg').text("Empty Weight/Capacity");
                $('#wc').focus();
                return false;
                }
        var address2 = $('#qty').val();
            if(address2==""){
                $('#msg').text("Empty Quantity");
                $('#qty').focus();
                return false;
                }
        var city = $('#date').val();
            if(city==""){
                $('#msg').text("Empty Date");
```

### Stock Calculation



```
<script type="text/javascript">
    $('#price, #qty').change(function(){
    var price = parseFloat($('#price').val()) || 0;
    var qty = parseInt($('#qty').val()) || 0;

    $('#tprice').val(price * qty);
});
</script>
```

# Appendix G - Client Certificate



**MAHANAMA PAINT CENTRE (PVT) LTD.**

336 A, Hospital Road, Kalubowila. Tel: 2766474 Fax : 2765420
E-mail: dilrukshi.mahanama@gmail.com isuru_luxor@hotmail.com

*BRANCHES:* 101/3, Highlevel Road, Maharagama. Tel: 4303997
137, Horana Road, Panadura. Tel: 038 2244163
106 D/1, Borella Road, Athurugiriya. Tel: 011 2762338
5/N3, 10th Mile Post, Katuwawala, Boralesgamuwa. Tel: 0778 784676

October 17, 2017

Project Examination Board,
University of Colombo School of Computing,
No 221/A, Dharmapala Mawatha,
Colombo 07

Dear Sir/Madam,

**LETTER OF CERTIFICATION**

This is to certify that Mr. G.U.K Gammanpila who is studying at University of Colombo School of Computing(UCSC) has successfully developed a Sales & Inventory Management System for Mahanama Paint Center (PVT) LTD.

I glad to say that this system has facilitated to increase the productivity of our business activiti and daily transactions. He has successfully completed our business requirements.

Thank you,

Your faithfully,

Mahanama Paint Center (PVT) LTD

(Isuru Mahanama)

(0778784676)

Mahanama Paint Center (Pvt)Ltd
336,Hospital Road,Kalubowila.
TEl : 2 76 64 74

# Glossary

**AJAX –** Stand for Asynchronous JavaScript and xml. Its running client side and helps to develop interactive web applications.

**CSS –** Stand for Cascading Style Sheets. This is use to apply different styles for markup languages.

**HTML –** Stand for Hyper Text Markup Language. Using design, the web pages.

**MySQL –** Most popular database management system and it can manage big amount of data in different datatypes.

**RUP –** Stand for Rational Unified Process. It's an iterative and incremental software methodology

**UML –** Stand for Unified Modeling Language.

**XAMPP –** This is open source bundle of software package. Its including Apache, PHP, MySQL, Perl

# Index