

Warehouse Management System

**E.G.N.N. Edirisinghe
2023**



Warehouse Management System

**A dissertation submitted for the Degree of
Master of Information Technology**

**E.G.N.N Edirisinghe
University of Colombo School of Computing
2023**



1.1 Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: E.G.N.N Edirisighe
Registration Number: 2019/MIT/020
Index Number: 19550202

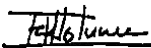


Signature: _____

Date: 27/04/2023

This is to certify that this thesis is based on the work of Mrs. E.G.N.N Edirisinghe under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:
Supervisor Name: Dr H.K.T.C Haloluwa



Signature: _____

Date: 27/04/2023

Preface

It is a great pleasure to accept this challenging project in this current situation. The motivation to carry out this project was the current loopholes in the WMS systems in current context. I hope the solutions given by this project and the prototype WMS system is successful.

I would take pleasure in heartily thanking the Supervisor, Dr Thilina Haloluwa, Senior lecturer of the University of Colombo UCSC, for excellence guidance and support during the process of project. Finally, I want to thank my Family and friends for being there for me.

Abstract

Warehouses play crucial roles in the transfer of resources, the storage and processing of commodities, the de-grouping of vehicle loads, the creation of stocks, the maintenance of unit collections, and the assembly of shipments, making them crucial nodes in a supply chain network.

The primary goal of the supply chain is to deliver goods to the final consumer as constantly and affordably as possible. This project aims to increase warehouse Operations at every Level. Mainly Storing, Stock Management and Ordering. Arranging the warehouse layout with 2D visualization is one of the major achievements of this project. That is, the Warehouse Administrator does not require to visit manually to find the stock locations. The stock arrangement can be viewed through the software in real-time. This is important to keep products in a safe environment, where, the environment will not affect the product. Accurate maintaining inventory record is another challenging task to accomplish in WMS. To overcome this notification method is introduced. Admin is notified with the list of stocks when the stock is reached its threshold value. The introduction of a unique barcode for each product is another feature introduced in the implemented WMS system. With this, the administrator can overcome the problems when adding products to the system. Login Performance has been used for admin, Employees and Stock managers. Relevant Responsibilities have been delegated according to the user roles. This system is easy to handle. Designed with consideration of UI and UX aspects. The Administrator can monitor the entire process through the computer with an internet connection. This will help the administrator to improve the warehouse processes and provide efficient services to the users. The Administrator has the privilege to generate reports. The Evaluation has been done with opinions and comments from Expertise in the relevant field. WMS System has thoroughly examined and evaluated in phrases of functionality, protection and usability.

Table of Contents

| | |
|--|-----|
| Declaration | i |
| Preface..... | ii |
| Abstract | iii |
| Table of Contents | iv |
| List of Figures | vii |
| List of Tables..... | ix |
| List of Acronyms..... | x |
| Chapter 1 - Introduction | 1 |
| 1.1. Objectives..... | 2 |
| 1.2. Scope | 3 |
| 1.3. Motivation | 3 |
| 1.4. Goal | 4 |
| Chapter 2 – Background..... | 5 |
| 2.1. Literature Review | 5 |
| The Importance of WMS..... | 6 |
| 2.2. Drawbacks of Current System..... | 6 |
| 2.3. Justification of Proposed System | 6 |
| 2.4. The Review of Similar Systems | 7 |
| 2.4.1. Blue Yonder | 7 |
| 2.4.2. WISE | 8 |
| 2.5. Comparison of the similar systems | 9 |
| 2.6. Summary | 9 |
| Chapter 3 – Design..... | 10 |
| 3.1. Analysis..... | 10 |
| 3.1.1. Functional Requirements..... | 10 |

| | | |
|--------|--|----|
| 3.1.2. | Non Functional Requirements..... | 12 |
| | Scalability..... | 12 |
| 3.1.3. | Security..... | 13 |
| 3.2. | Technique Used in Requirement gathering..... | 13 |
| 3.2.1. | Site Visit..... | 13 |
| 3.2.2. | Observe Phone Conversation..... | 13 |
| 3.3. | Design Overview..... | 13 |
| 3.4. | Design..... | 14 |
| 3.4.1. | Limitation of the System Usage..... | 14 |
| 3.4.2. | ER Modal..... | 15 |
| 3.4.3. | Use Case Diagram..... | 16 |
| 3.4.4. | Activity Diagram..... | 17 |
| 3.4.5. | Class Diagram..... | 25 |
| 3.4.6. | Database Design..... | 26 |
| 3.5. | Summary..... | 29 |
| | Chapter 4 – Implementation..... | 30 |
| 4.1. | Introduction..... | 30 |
| 4.2. | Development Process Model..... | 30 |
| 4.3. | User Interface..... | 31 |
| 4.4. | Development..... | 32 |
| 4.5. | Technology used..... | 36 |
| 4.6. | Specifications of the hardware and software..... | 36 |
| 4.6.1. | Hardware Specifications..... | 36 |
| 4.7. | Screenshots of the user interface..... | 37 |
| 4.8. | Summary..... | 40 |
| 5.1. | Test Plan and Results..... | 42 |
| 5.2. | Performance Testing..... | 44 |
| 5.3. | Security Testing..... | 45 |
| 5.4. | Cross Browser Testing..... | 45 |

| | | |
|--------|---------------------------------|----|
| 5.5. | Usability and Evaluation | 45 |
| 5.5.1. | Criteria to select Users | 45 |
| 5.5.2. | Methodology Used | 45 |
| 5.5.3. | Area Focused..... | 45 |
| 5.6. | System Evaluation..... | 46 |
| 6.1 | Conclusion | 47 |
| 6.2 | Achievement of Objectives | 47 |
| 6.3 | Future Work | 48 |
| 6.4 | Knowledge gained | 48 |
| | References | 49 |
| | Appendix A | 49 |
| | Appendix B | 52 |

List of Figures

- Figure 2.1. Proposed WMS System Functional requirement7
- Figure 2.2 Intelligent Inventory Management in Blue Yonder7
- Figure 2.3. Resource orchestration in Blue Yonder8
- Figure 2.4. Overall Functionalities of WISE.....9
- Figure 3.1 Proposed WMS System Functional requirement12
- Figure 3. 2 ER Modal for Proposed WMS System15
- Figure 3. 3 High Level Use Case Diagram of WMS16
- Figure 3.4. The proposed Activity diagram for the WMS system17
- Figure 3.5. The proposed Activity diagram for Login17
- Figure 3.6. Activity diagram for Change Password18
- Figure 3.7. Activity diagram for Edit User Profile..... 19
- Figure 3.8. Activity diagram for Add product..... 20
- Figure 3.9. Activity diagram for Add stock21
- Figure 3.10 Activity diagram for Add orders22
- Figure 3.11 Activity diagram for Add brand..... 23
- Figure 3.12 Activity diagram for generate report24
- Figure 3.13 Class Diagram of WMS24
- Figure 4.1 Phases of the V-Model30
- Figure 4.2 Common Components FE/131
- Figure 4.3 Common Components FE/2.....32
- Figure 4.4 Common Components FE/3.....32
- Figure 4.5 Interactions between Modules.....33
- Figure 4.6 Login Page37

| | |
|---|----|
| Figure 4.7 dashboard Admin Panel | 38 |
| Figure 4.8 Stock Page..... | 38 |
| Figure 4.9 Brand Page | 39 |
| Figure 4.10 Report Page | 39 |
| Figure 4.11 Order Page..... | 40 |
| Figure 5.1 Validation for Errors Login..... | 43 |
| Figure 5.2 Validation for incorrect Credentials Login Module..... | 43 |
| Figure 5.3 Sample Test Cases for data Table Module..... | 43 |
| Figure 5.4 User Evaluation Analysis | 46 |
| Figure B.1 Add Stock Form | 52 |
| Figure B.2 Stock Real Time Detail Card..... | 53 |
| Figure B.3 Real Time Brand Detail Cards | 54 |
| Figure B.4 Product Reorder Reordering..... | 57 |

List of Tables

Table 2.1 Comparison of similar system with proposed WMS.....9

Table 3.1 Structure of Table Product.....26

Table 3.2 Structure of Table Brand26

Table 3.3 Structure of Table Stock.....26

Table 3.4 Structure of Table Employee.....27

Table 3.5 Structure of Table User27

Table 3.6 Structure of Table Department28

Table 3.7 Structure of Table password Reset28

Table 3.8 Structure of Table Orders28

Table 5.1 Sample Test Cases for Login Module42

Table 5.2 User Evaluation Results44

Table 5.3 Sample Test Cases for Data Table Module46

Table B.1 Test Cases for Stock Module54

Table B.2 Test Cases for Brand Module55

Table B.3 Test Cases for Product Module.....56

Table B.4 Test Cases for Product Reorder Layout.....57

Table B.5 Test Cases for order Module.....58

List of Acronyms

WMS : Warehouse Management System

intelligent_inventory_management

Omni channel ecosystem

SDLC Model : Software Development Life Cycle

BE: Back End

FE- Front End

URL –Uniform Resource Locator

JS - JavaScript

Chapter 1 - Introduction

In Industries, products and raw materials were stored in a warehouse before delivering them to the Customer for production and Consumption purpose. The primary goal of the supply chain is to deliver products to the end user as constantly and affordably as possible. Warehouses are crucial nodes in a supply chain network because they carry out crucial tasks that enable material mobility, the storage and processing of goods, the de-grouping of vehicle loads, the creation of stocks, the maintenance of unit collections, and the assembly of shipments (Ripapel 2005).

The function of a warehouse is crucial since it serves as a location to store goods in a secure manner and ensure that their exposure to the environment will not harm them. Warehousing has been acknowledged as one of the key strategies for gaining an edge in the market. (Miralam .M 2017). Additionally, lot of experts have discovered that WMS aids businesses in order and shipment accuracy as well as visibility and monitoring of the products inside the warehouse. WMS guarantees to finish and manage the task within the allotted timeframe with the necessary accuracy. Therefore, WMS enables organizations to make the most of their employees' knowledge and experience in order to grow and improve their operations.

When considering the current industries, the companies are experiencing significantly high labor costs, an Inventory cost where it is largely based on time and manpower, when considering, to handle the inventory. More over when closely observing, the handling products in warehouses in current WMS systems have following drawbacks. Most of the time, lack of addressing operating Expenses, lack of addressing Cost of Carrying Inventory, problems in Inventory accuracy. This Project mainly focuses to following Major Problems.

- a. Lack of Efficiency in to Profitability in Current Warehouse Management Systems.
- b. Increase, the warehouse Operations in every Level (Storing, Sorting, Order Picking)

Warehouse management system (WMS) is a software solution. Intended for business owners to support daily operations including product storage, picking and packaging assistance, inventory replenishment, and so on. This makes operating the warehousing business simple and effective. This project will automate manual warehouse processes and integrate a software solution to improve warehouse operations on a daily basis.

"Bradshaw International's Good Cook" is one of the best instances of an ideal WMS system when looking at real-world examples of effective WMS systems. Leading kitchen product marketer Good Cook created Good Cook to include quality, value, and service for each product. Their distribution system underwent alteration as a result. They developed EDI software, a billing software, and WMS integration in addition to order management software. The team developed labor schedules and an appropriate workflow after profiling SKU movements, identifying pick, and storage media to model replenishment rates. They are combined with WMS. By using this approach, Bradshaw was able to meet demand.

In this project, following are the main objectives planned to be achieved at the end of the project. They could be summarized as follows.

1.1. Objectives

a. To Reduce the Number of Inventory out of Stocks.

Maintaining buffer value for each stock make easy to operate the warehouse efficiently. Implementing buffer value according to the available stock is one of a main objective of this project.

b. To Organize the Warehouse Layout Visually (2D Visualization) through the Software.

Real time organizing the warehouse product is one of the main objective of this project. During this initial warehouse product layout should be tracked and arrange on the dashboard and thereafter the Admin can alter the product storage places through system.

c. To Increase, Profitability per Order.

Reducing time taken to manual search of WMS Entries will increase the profitability. To achieve this objective implementation of sort search functionality is planned.

d. To reduce the phone Communication Cost.

To decrease the communication through telephone for warehouse operation will decrease the efficiency of the work. This could be eliminating by implementing real time update of records.

e. To Generate MIS Reports

This objective is vital for implementation for management team for real time decision making. Objective of this project is to generate reports on real time with respect to specific date range.

1.2. Scope

Keeping the products in safe condition in the store is essential and this is implemented by the proposed WMS system that is called as arranged the warehouse layout visually (with 2D visualization). There, the store administrator can see in 2D, and choose the exact space to keep the particular product or raw material without affecting the other product's spaces through the system. This feature, which eliminates the need to manually search each product's space in the store, also promotes effective space usage.

Generating MIS reports is another added feature in this system. Where this allow generating multiple reports related to the WMS store. For instance, by accurate maintaining inventory recordings, warehouse transactions will help to generate accurate and quality MIS reports.

To achieve this feature of this system, a method of tracking of inventory will be introduced. The store admin is notified with the list of stocks when the stock is low. Adding products to the system is another difficult process when maintaining a WMS system.

As a solution for this each product is introduced with a unique barcode. Where the products are being added to the system using that unique barcode with the help of barcode scanner. This will increase the efficiency of adding products to the WMS system.

1.3. Motivation

The issue of Lacking a complete open source Warehouse Management system, which is compatible for both MAC and Linux platforms. Current warehousing has caused major impacts on warehouse management in current context. This project identifies the main loopholes in current warehouse management systems by researching the relevant case studies. Then the results suggest the lack of proper knowledge of WMS processes in their current systems, Lack of knowledge for integration of WMS processes in to their industries are major issues present in current context. Not only that, absence of an open source WMS system for a reasonable price is another issue.

Since these issues could be addressed efficiently by current Information Technology, which will eliminate the above issues. This research focuses to establish modules to fulfil the above stated requirements using current technology. The implemented system could be used in an efficient and productive mean.

1.4. Goal

Primary aim of this project is to build WMS system where it is compatible with Mac OS and Linux OS, which include primary considerations for an ideal WMS System.

The proposed system is an online service, where the Architecture comprises with MVC with the layered pattern. Other important Implementation Technologies and Considerations could be summarized as follows.

- Architectural patterns: Layered Pattern
- Programming Language: PHP- Laravel
- Back End Database: MYSQL
- The Proposed WMS System focuses on following modules.
 - Login Module
 - Product Module
 - Order Module
 - Stock Module
 - Report Module

Chapter 2 – Background

2.1. Literature Review

The role of a warehouse is essential since it provides a spot for the secure storage of items and guarantees that their exposure to the environment will not harm them. One of the essential tactics for establishing a competitive edge in the market is warehousing, which has recognized from earlier. WMS also helps companies with order and shipment accuracy as well as visibility and monitoring of the products inside the warehouse, according to many experts. WMS promises to complete and manage the task with the required accuracy and within the allocated timeframe. WMS enables businesses to maximize the knowledge and experience of their staff members in order to expand and enhance their operations.

Basic warehouse activities could be listed as follows.

Warehouse flows involve a variety of tasks. These tasks include receiving, transferring and storing, order selection and picking, accumulating and sorting, cross-docking, and shipping..

- Receiving operation - This activity entails bringing commodities and products into the warehouse from the transportation network; the inbound carrier is slated to deliver the item at a particular time to increase the effectiveness of unloading and the labor productivity in the warehouse.
- Put-away operation - This procedure involves locating the things and moving the products and goods to the proper area after identifying the product, usually by scanning the product's barcode.
- Order-picking process: This calls for warehouse staff to choose the things the customer has requested from a storage location or region.
- Accumulation - If the orders were picked in batches, accumulation, or sorting them into client orders, is an essential task. In such circumstances, after the picking procedure is complete, the picked items must be organized by client order. Orders must frequently be packed and piled on the appropriate unit load after choosing (e.g. a pallet).
- Cross-docking is the process of moving the received goods directly to the shipping docks.

In order to smooth running of business automating of warehousing activities play a major role. This makes easy and efficient run of business. This project will incorporate a software solution and automate manual warehouse activities to enhance everyday warehouse operations.

The Importance of WMS

Lack of a comprehensive open source solution that works on both MAC and Linux systems is the problem. Current warehousing has had a significant impact on warehouse management. Through a review of pertinent case studies, this study pinpoints the primary weaknesses in the existing warehouse management systems. The findings then point to a fundamental problem in the existing situation: a lack of proper knowledge of WMS processes in their current systems and a lack of understanding of how to integrate WMS procedures into their industries. Another problem is the lack of an open source WMS system that are affordable.

2.2. Drawbacks of Current System

Currently Most of the industries in Sri Lanka are not able to maintain a smooth-running WMS system. According to researchers, the main issue for this is, the lack of Knowledge to automate WMS Processes, Following loopholes found in the existing WMS System. Inefficient product storage. Because of that, the ineffective use of space is still a common pain point in warehouses, which is not addressed fully by the current warehouse systems. The best way to deal with this burden would be to create the most efficient warehouse architecture possible using modeling.

2.3. Justification of Proposed System

In the proposed WMS, this issue addresses using 2D visualization of warehouse layout feature. Another loophole in common warehouse software is bad inventory management. When inventory-checking activities are proceed with using old software, errors occur easily. Real-time inventory management features in general warehouse management software could help to solve this problem. In this approach, inventory data first collected via a handheld or stationary device, like a barcode scanner.

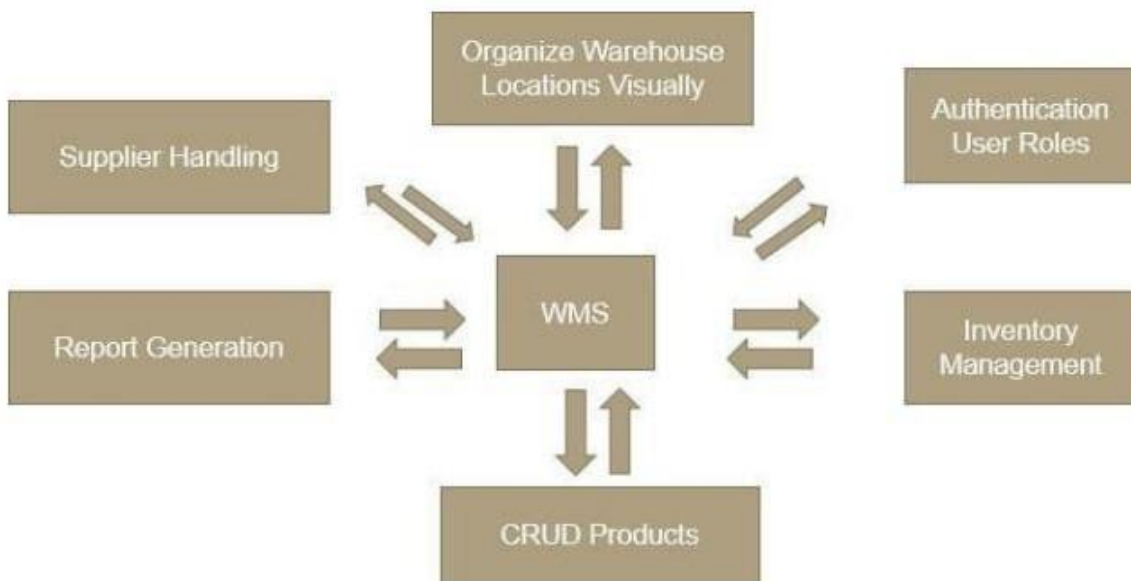


Figure 2.1. Functional Requirements of proposed WMS

2.4. The Review of Similar Systems

2.4.1. Blue Yonder

Blue Yonder is one of powerful warehouse management solutions. It keeps the users on track and allows us to execute the correct inventory at the right time. The interface is simple and speedier manner. In this system, Complex and time-consuming operations can be carry out intelligently in less time. The most important aspect of this solution is that it allows a large corporation to survive in a continuously changing, densely networked

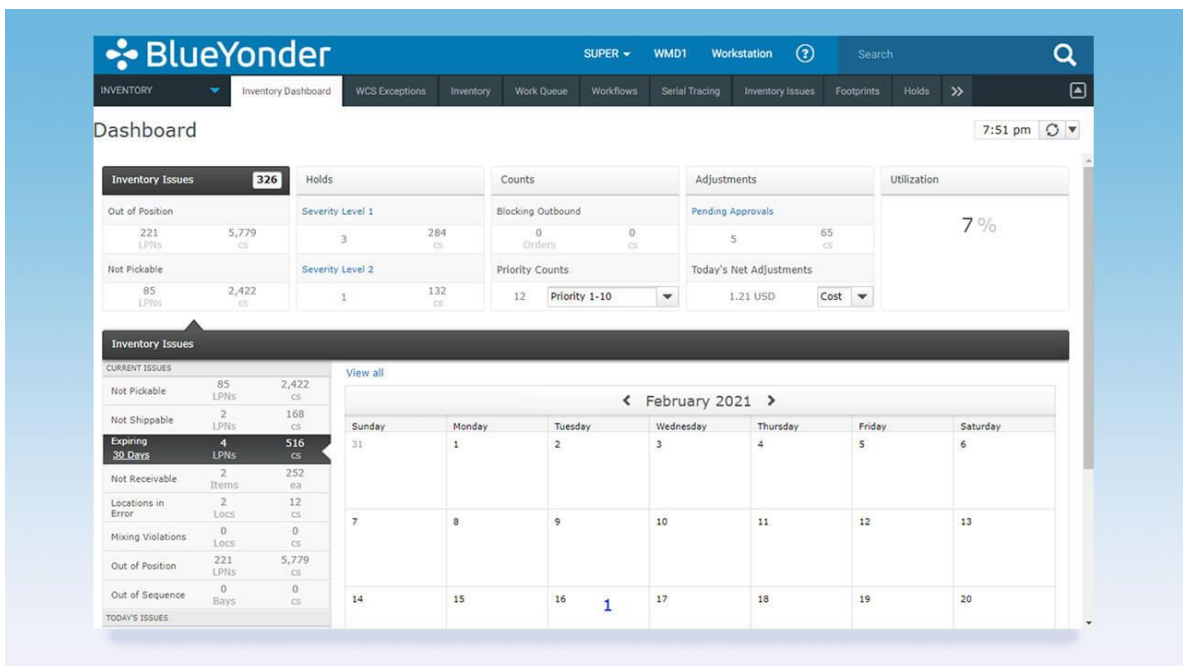


Figure 2.2. Intelligent Inventory Management in Blue Yonder

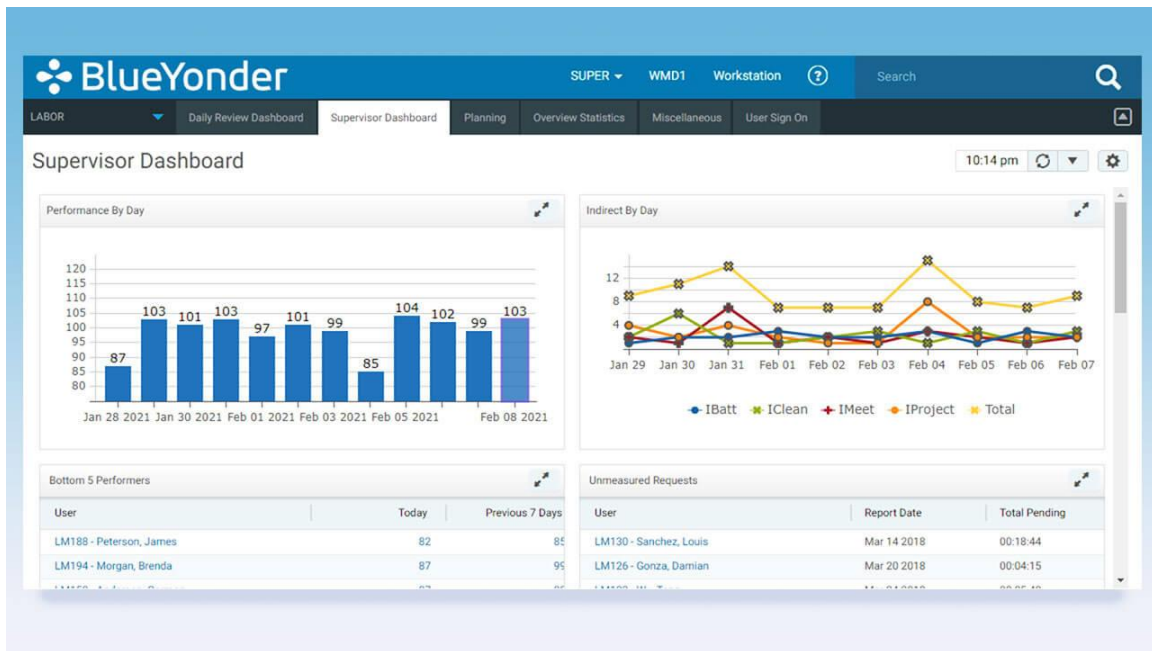


Figure 2.3. Resource orchestration in Blue Yonder

2.4.2. WISE

A rules-based warehouse management system (WMS) developed by Royal 4 Systems, called WISE, may handle a variety of industries, including 3PL, the automobile, food and beverage, and industrial machinery.

Users may manage key warehouse activities like tasks, operations, budgets, and productivity tracking with the use of WISE. Receiving, directed putaway, picking and slotting, cross-docking, enterprise resource planning (ERP) connection, and other features are among its features.

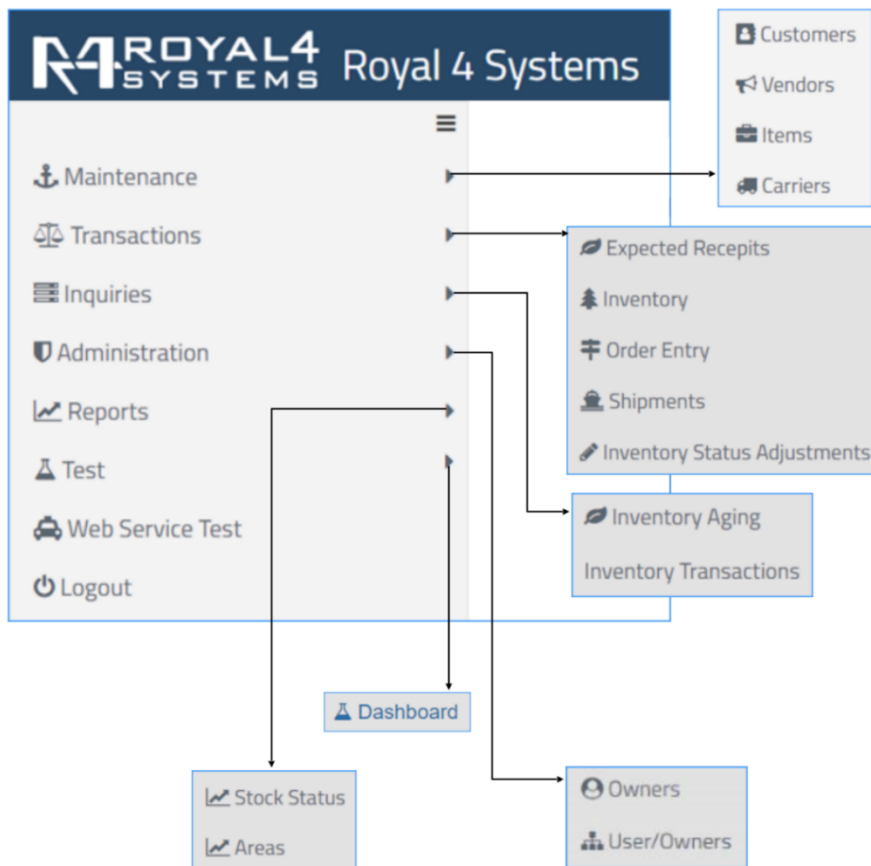


Figure 2.4. Overall Functionalities of WISE

2.5. Comparison of the similar systems

| | Blue Yonder | WISE | Proposed System |
|---|---------------|---------------|-----------------|
| 2D Visualization of Store Products | Not Available | Not Available | Available |
| Rearrange the Products shelves through system | Not Available | Not Available | Available |
| Maintain a Buffer Stock | Not Available | Available | Available |
| Availability in Open Source | Not Available | Not Available | Available |
| Security | Available | Available | Available |
| Authentication | Available | Available | Available |

Table 2.1. Comparison of similar system with proposed WMS

2.6. Summary

This Chapter summarizes the Similar Web Application for the industry request. This includes information on the Warehouse Management system

Chapter 3 – Design

3.1. Analysis

In the Analysis Stage Published Research, data used to understand the real world Operations of WMS. Several issues discovered and documented during analysis stage. According to the importance and complexity of the problem, they were classified and prioritized. They can be summarize as follows.

1. Operating expenses associated with the WMS
2. Lack of Addressing the Carrying cost of inventory
3. Gain efficiency and Profitability through operating a WMS
4. Problems in inventory accuracy

In order to address those areas it is crucial to know the overall perspectives of WMS in general. Generally, WMS includes the activities starting from receiving operation to the shipping process.

The project carried out in accordance with the principles of software Engineering. In here, the V model chosen as SDLC Model where execution of Processes happens in a sequential manner in V shape. The advantage of applying the V- model is the simplicity it poses and project processes could be a process gradually. Moreover, the basic software Engineering principles are followed, such as, perform continues validation of the product, usage of modern programming practices, maintain the product quality throughout the project development.

3.1.1. Functional Requirements

Functional requirements of the proposed system illustrated as follows.

Signup and User Authentication

In order to access, this system a user needs to sign up to the system. There are three user types in web application. Super admin, Warehouse admin, Stock Manager. Then users authorized when login to the system and the user roles will be different according to the user type. Users assigned to a department and User is an Employee in the proposed system. According to User roles, permissions are delegate in the given system.

Product Registration and adding a unique barcode

This functionality is restricted to Warehouse admin. Whenever a product registered to the system, introduced with a product unique barcode. Product details added in this point such as Product Id, name, Description, Expiry Date, Manufacture date, etc. Auto count also will be capture to the system where it will update the inventory for the specific product. Moreover, the admin will have features to create, edit or delete, update product records.

Proposed WMS is an IT web-based system, which could be used to coordinate warehouse activities and increase warehouse efficiency.

Inventory Management

The warehouse products are being add and tracked in this Module. Real time updating of stock item is ensured in the system. Modifying the stock items , deleting of stocks functionalities are restricted according to user roles. In the proposed system, stocks are being updated, once an order is added to the system. . Search functionality for any attribute, pagination sort functionalities are facilitated in this module.

Orders Management

Adding orders is the main functionality performed in this system. This is also restricted according to the user roles. Notifying the invalid stocks is triggered when adding an order. This module allows for tracking of payment information, client information, outstanding balance, and order status. This module facilitates search capabilities for any attribute and pagination sort functionality.

Brands Management

This module is used to add new brands to the system, change existing brand information, and keep track of a brand's availability. This module facilitates search capabilities for any attribute and pagination sort functionality.

Report Generation Module

Generation of MIS report is one of main functionalities in a WMS. In the proposed prototype, Reports of Orders, Report of Stocks real time count with availability, report of Brands availability, Report of Products details could be generated with a selected date frame.

Above-mentioned business functions could be outlined as follows.

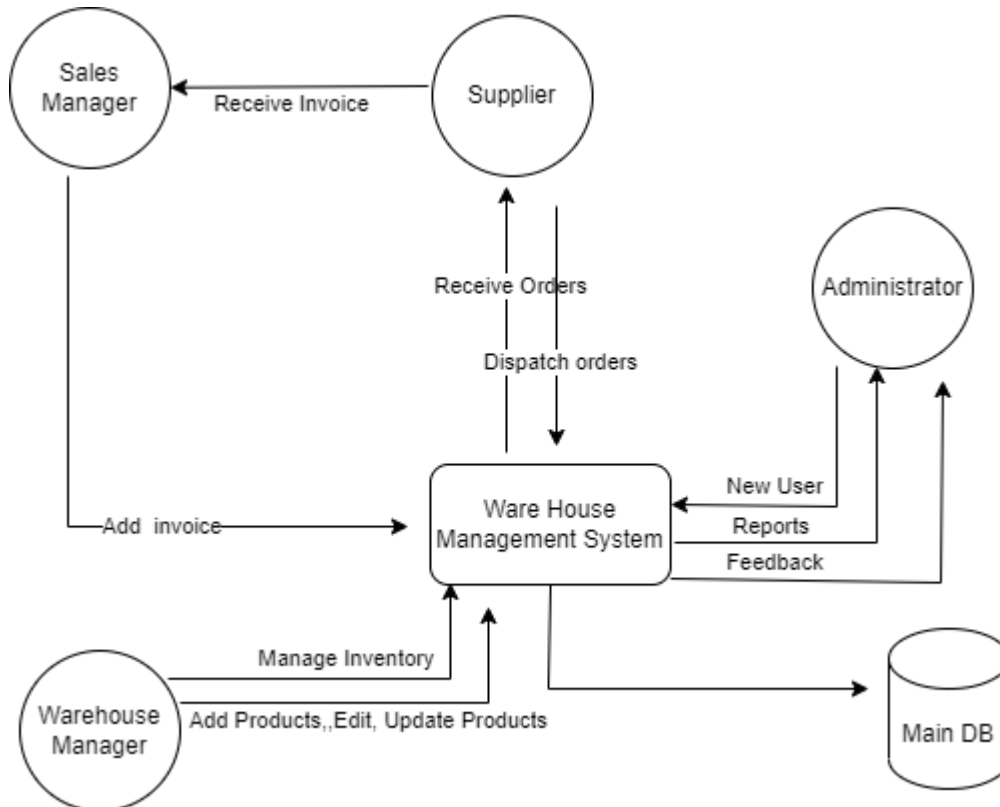


Figure 3.1. Proposed WMS System Functional requirement

3.1.2. Non Functional Requirements

Scalability

Test for the highest workload the system could hold. Maintain the highest threshold value is must in a successful system. The Consequences should be followed to achieve this feature.

Usability

The proposed system should be user friendly. Tests should be carried out to check the usability level of the system.

Reliability and Availability

The system should test for critical failure time in normal usage. The system should not wait for a long time to respond under this condition.

3.1.3. Security

- The System comprised of secure Database.
- Regular users are permit to view information only, and not permitted to alter or modify anything other than their personal information.
- There are access levels for users.
- User authentication has done properly.
- The user's information and any other information cannot be hacked.
- Users without access to the database must have their own space, and only the administrator permits to make changes to the database.

3.2. Technique Used in Requirement gathering

3.2.1. Site Visit

During this information gathered by visiting the site. Discussion with the Support team to understand the operating methods of the Current WMS system. During this discussion, importance of a new system was discuss. Not only that, what they expect from new system was discuss. Moreover, current drawbacks of the existing system was revealed.

3.2.2. Observe Phone Conversation

Track the phone conversation related to current WMS system with IT department. Currently issues were configured with the help of telephone conversation through IT department. Many ambiguous and doubts are in those conversation related to current WMS

3.3. Design

Software in client/server architecture is split into two layers. Server and client. Where the client is capable of processing and storing data. This architecture allows for a significant reduction in both server computation and network traffic. Additionally, the design can allocate

jobs to the client and server to obtain appropriate results, lowering system overhead, and fully utilizing the benefits of both ends of the hardware environment.

Browser, Web server, and data server are the three components that make up the Browser/Server (B/S) architectural software. The Web server is the main component of the structure. It is in charge of receiving remote HTTP query requests, interpreting the query's conditions, sending the results into HTML and various page description languages, and sending the translated results back to the browser that issued the query request. Similar requests are made by browsers to Web servers for the addition, deletion, and modification of data record description languages.

3.4. Design

3.4.1. Limitation of the System Usage

- The administrator and user must remember the login ID and password.
- The user must have an Internet connection.
- Initial Warehouse layout should give to arrange the store visually.

3.4.2. ER Modal

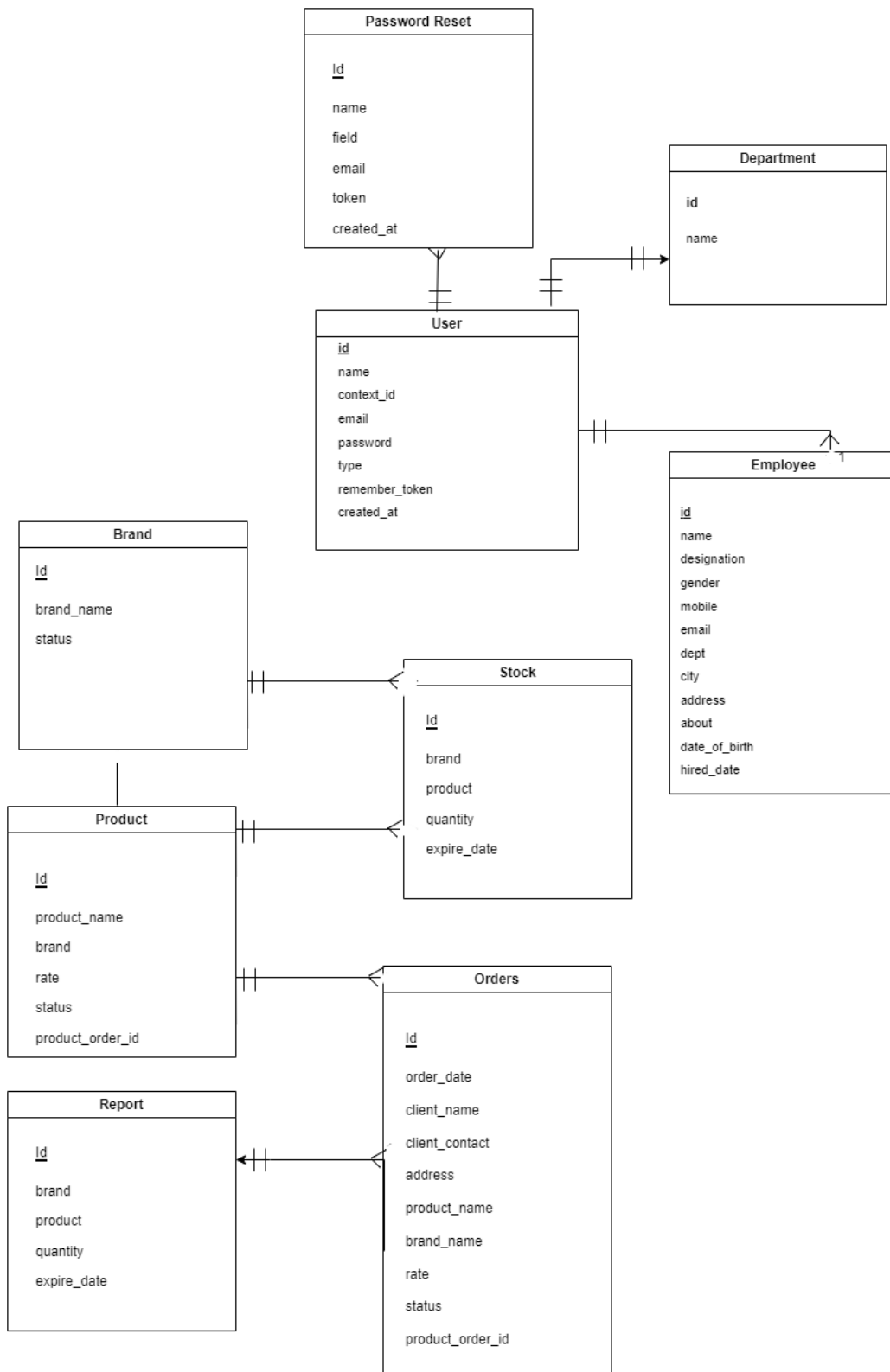


Figure 3.2. ER Modal for Proposed WMS System

3.4.3. Use Case Diagram



Figure 3.3 High Level Use Case Diagram of WMS

3.4.4. Activity Diagram

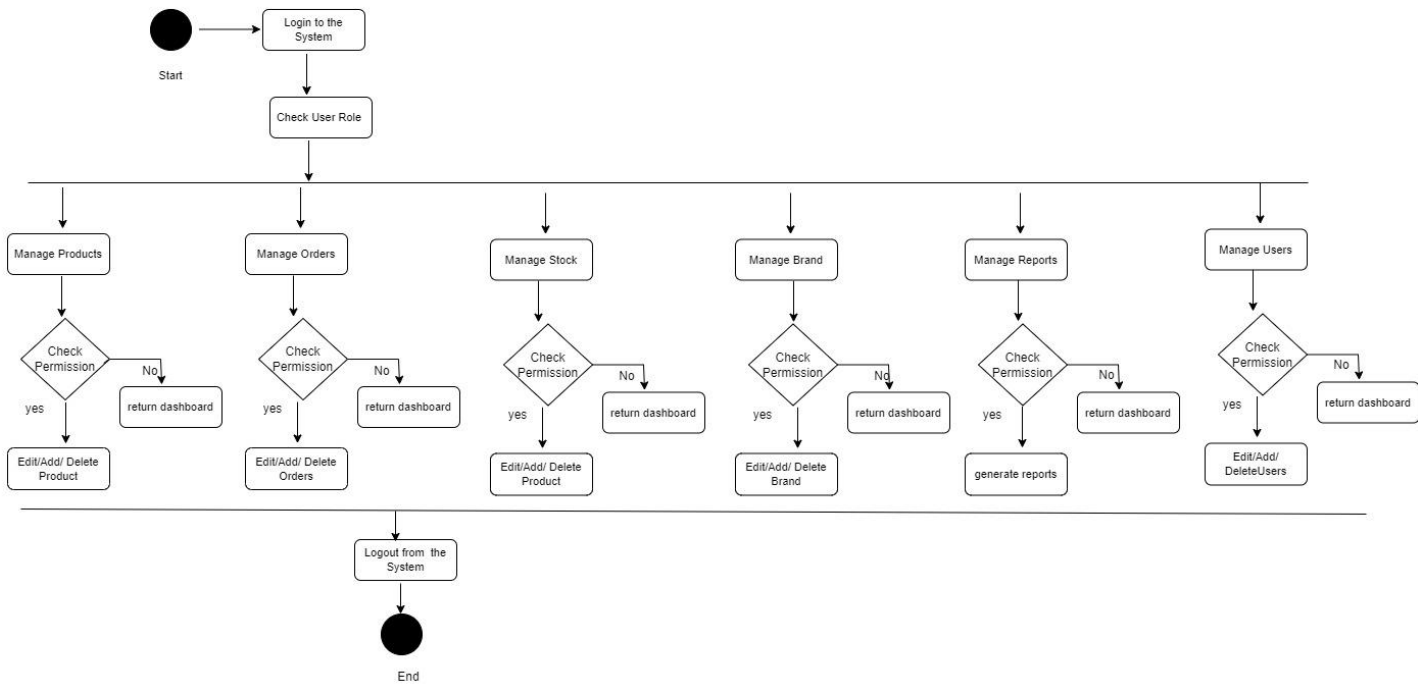


Figure 3.4. The proposed Activity diagram for the WMS system

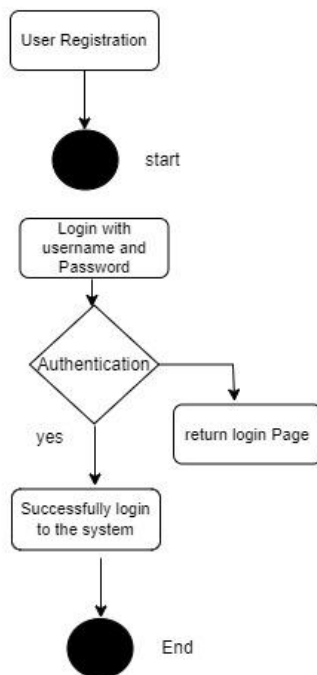


Figure 3.5. The proposed Activity diagram for Login

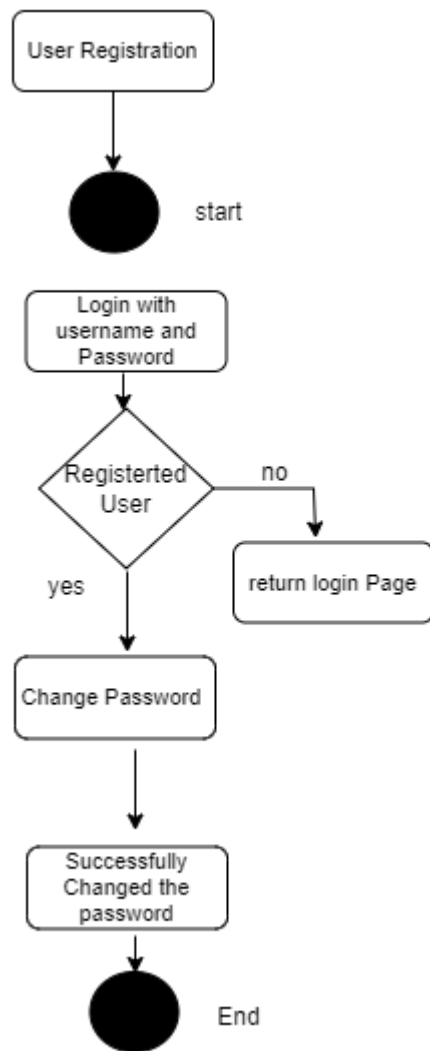


Figure 3.6. Activity diagram for Change Password

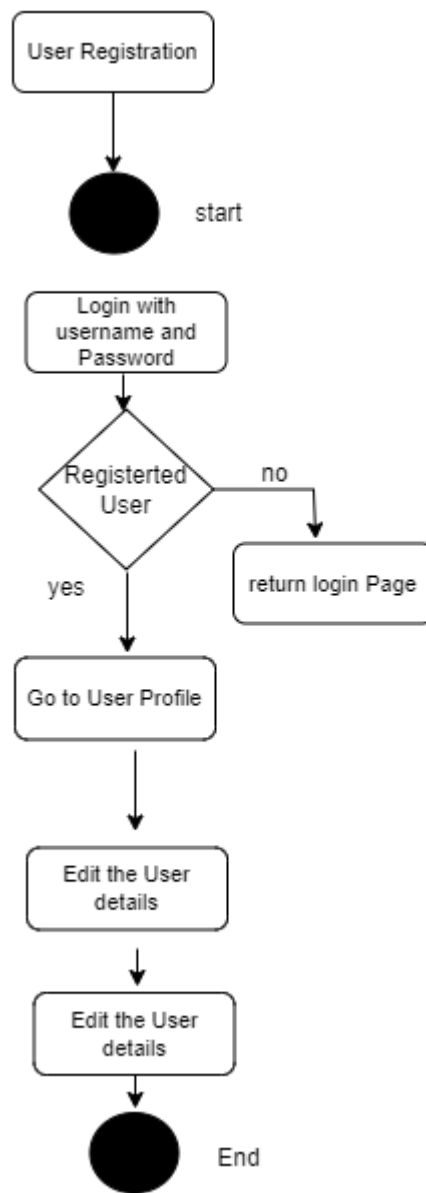


Figure 3.7. Activity diagram for Edit User Profile

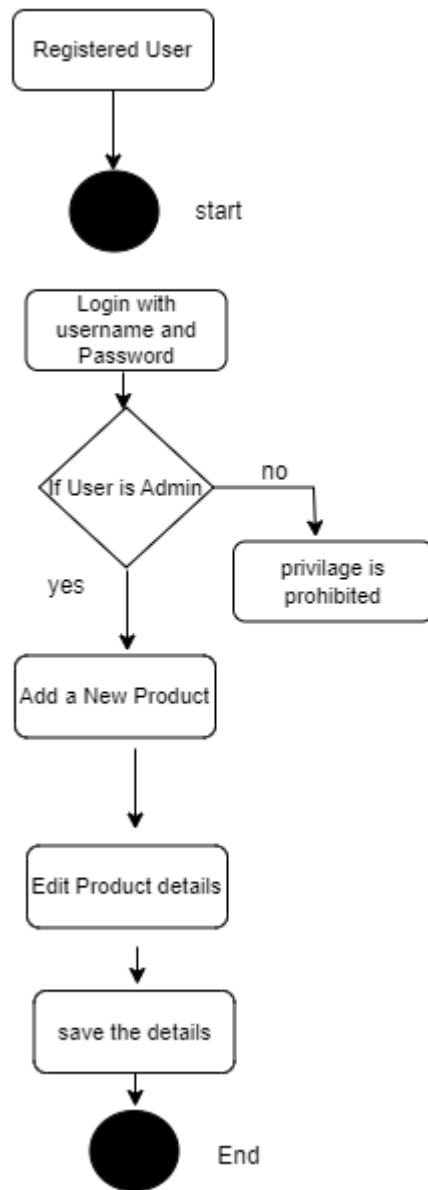


Figure 3.8. Activity diagram for Add product

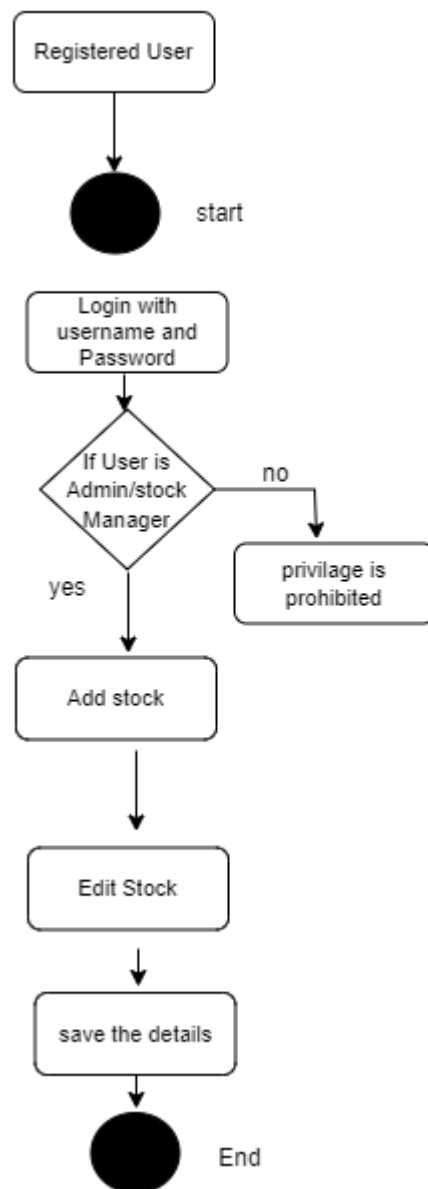


Figure 3.9. Activity diagram for Add stock

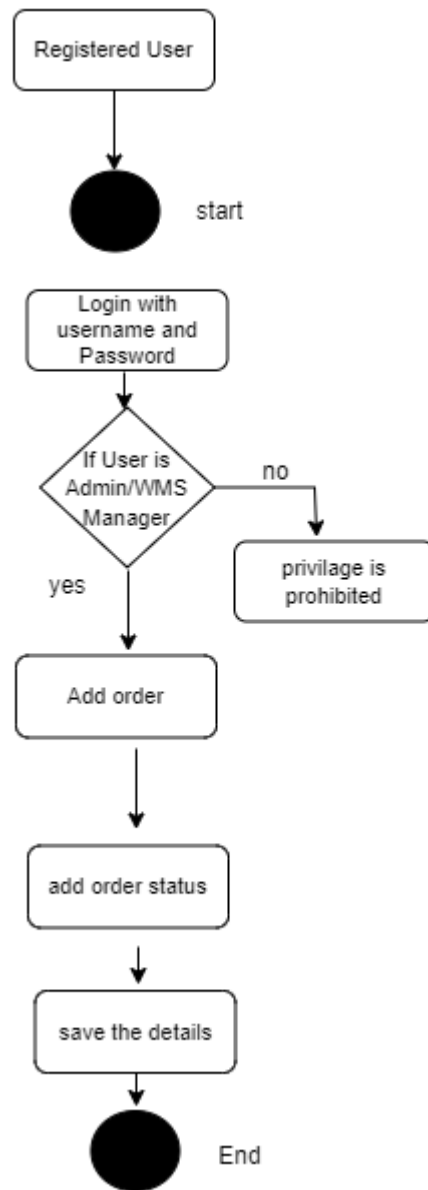


Figure 3.10 Activity diagram for Add orders

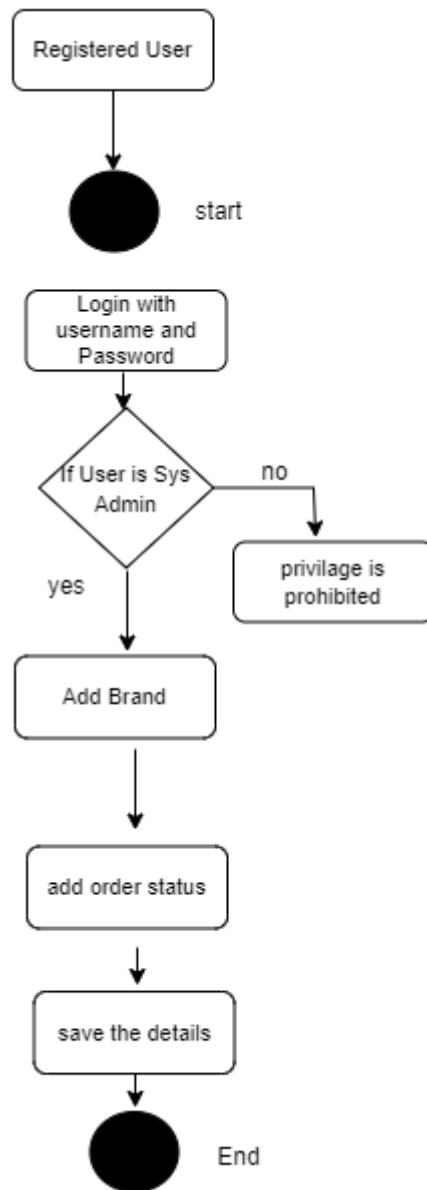


Figure 3.11 Activity diagram for Add brand

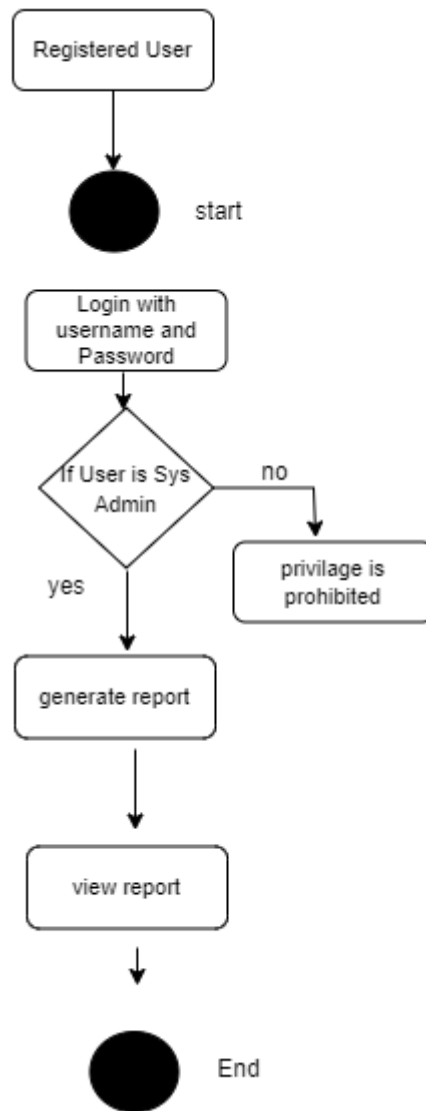


Figure 3.12 Activity diagram for generate report

3.4.5. Class Diagram

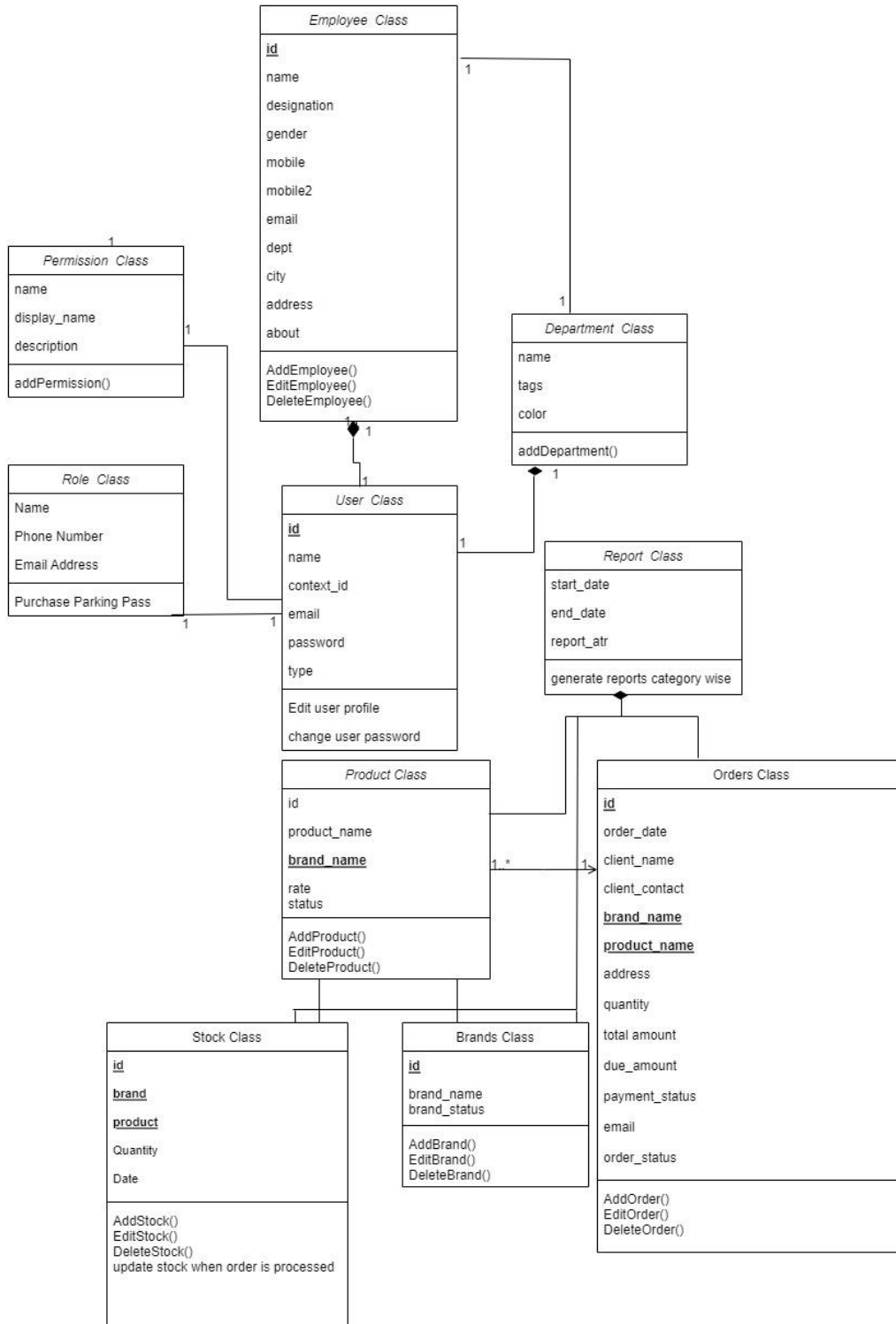


Figure 3.13 Class diagram of WMS

3.4.6. Database Design

MySQL is used to establish logical model for the database. The following tables with the description is given below. This include data type, length, null, default, Primary and Foreign Keys.

| Field | Type | Null | Key | Default | Extras |
|------------------|--------------|------|-----|---------|----------------|
| Id | Int(150) | No | PRI | None | Auto Increment |
| product_name | Varchar(256) | No | | None | |
| Brand | Varchar(256) | No | | None | |
| rate | Varchar(256) | No | | None | |
| status | Varchar(80) | No | | None | |
| product_order_id | Int(5) | No | | None | |

Table 3.1 Structure of Table Product

| Field | Type | Null | Key | Default | Extras |
|------------|--------------|------|-----|---------|----------------|
| Id | Int(150) | No | PRI | None | Auto Increment |
| brand_name | Varchar(256) | No | | None | |
| status | Varchar(80) | No | | None | |

Table 3.2 Structure of Table Brand

| Field | Type | Null | Key | Default | Extras |
|-------------|--------------|------|-----|---------|----------------|
| Id | Int(10) | No | PRI | None | Auto Increment |
| brand | Int(10) | No | FK | None | |
| product | Int(10) | No | FK | None | |
| quantity | Varchar(256) | No | | None | |
| expire_date | date | No | | None | |

Table 3.3 Structure of Table Stock

| Field | Type | Null | Key | Default | Extras |
|---------------|--------------|------|-----|---------|----------------|
| id | Int(10) | No | PK | None | Auto Increment |
| name | Varchar(250) | No | | None | |
| designation | Varchar(50) | No | | None | |
| gender | Varchar(255) | No | | None | |
| mobile | Varchar(20) | No | | None | |
| email | Varchar(250) | No | | None | |
| dept | Int(10) | | FK | | |
| city | Varchar(255) | No | | None | |
| address | Varchar(255) | No | | None | |
| about | Varchar(255) | No | | None | |
| date_of_birth | Date | No | | None | |
| hired_date | Date | No | | None | |

Table 3.4 Structure of Table Employee

| Field | Type | Null | Key | Default | Extras |
|----------------|---------------|------|-----|---------|----------------|
| id | Int (10) | No | PK | None | Auto Increment |
| name | Varchar (255) | No | | None | |
| context_id | Varchar (255) | No | | None | |
| email | Varchar (250) | No | | None | |
| password | Varchar (100) | No | | None | |
| type | Varchar (255) | No | | None | |
| remember_token | Varchar (100) | No | | None | |
| created_at | time_stamp | No | | None | |

Table 3.5 Structure of Table User

| Field | Type | Null | Key | Default | Extras |
|-------|--------------|------|-----|---------|----------------|
| id | Int(10) | No | | None | Auto Increment |
| name | Varchar(250) | No | | None | |

Table 3.6 Structure of Table departments

| Field | Type | Null | Key | Default | Extras |
|------------|--------------|------|-----|---------|----------------|
| id | Int(10) | No | | None | Auto Increment |
| name | Varchar(250) | No | | None | |
| Field | Type | Null | Key | Default | Extras |
| email | Varchar(250) | No | | None | |
| token | Varchar(250) | No | | None | |
| created_at | timestamp | No | | None | |

Table 3.7 Structure of Table password Reset

| Field | Type | Null | Key | Default | Extras |
|------------------|--------------|------|-----|---------|----------------|
| Id | Int(150) | No | PRI | None | Auto Increment |
| order_date | Varchar(256) | No | | None | |
| client_name | Varchar(256) | No | | None | |
| client_contact | Varchar(256) | No | | None | |
| address | Varchar(256) | No | | None | |
| product_name | Varchar(256) | No | | FK | |
| brand_name | Varchar(256) | No | | FK | |
| ra | Varchar(256) | No | | None | |
| status | Varchar(80) | No | | None | |
| product_order_id | Int(5) | No | | None | |

Table 3.8 Structure of Table Orders

3.5. Summary

The significance of the alternative system has been outlined in this chapter. The specifications for functional, non-functional, and security have also been included. It has also been mentioned that the data has been gathered to correctly build the system. Additionally, the UML diagram design, the database structure is described.

Chapter 4 – Implementation

4.1. Introduction

WMS is broken up by three modules. The system management module, the data management module, and the warehouse management module are the three core modules. Authentication and Authorization of system functions are the primary responsibilities of the system administration module. The primary goals of the data management module are the storage and retrieval of data. The following are the key features of the warehouse management module. Allocating space in the warehouse, keeping track of the inventory there, and allocating products are all laborious processes.

4.2. Development Process Model

Since the requirements of the system is identified early, clearly and the requirements do not Subject to change V-model is chosen as Process, execution occurs sequentially and in a V-shape according to the SDLC model. Based on the correlation between a testing phase and each related development stage, this is the case. This is a very structured approach, and the start of the subsequent phase only occurs after the conclusion of the preceding phase. The several stages of the SDLC's V-Model are shown in the following figure.

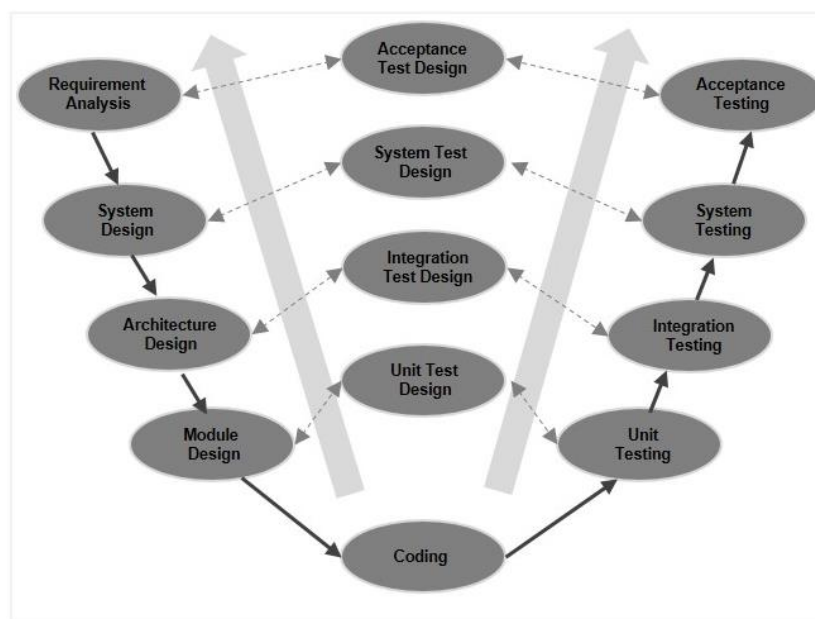


Figure 4.1. Phases of the V-Model

The benefit of using the V-model method is that it is very simple to comprehend and put into practice. The model's simplicity also makes management simpler. Since the requirements are predefined and identified clearly this model helps to proceed the project easily gradually.

4.3. User Interface

The user interfaces have implemented using HTML5, CSS3, jQuery, JavaScript, Ajax, Bootstrap. Majority of form fields, tables styles were inherited from Bootstrap default colors and CSS styling.

Form Validation are used to reduce human errors. Error handling is done commonly for similar form data.

Front End Common Modules Used in styling

1. Data Table Components Styling
 - i. Search
 - ii. Sort
 - iii. Action Buttons (Edit, Delete)
2. Pagination
3. Form Controls
4. Buttons in forms
5. Detailed cards

The screenshot shows a data table with the following structure and content:

| Id | Product Name | Brand | Rate | Status | product Order Id | Actions |
|----|--------------|----------|------|---------------|------------------|----------------|
| 1 | Cement | Relience | 3000 | Available | 1 | Action Buttons |
| 2 | Steal | Tata | 1200 | Available | 2 | |
| 3 | Tyre | Tata | 3000 | Not Available | 3 | |
| 5 | metal bars | Relience | 3000 | Available | 4 | |

Annotations in the image include: 'pagination' pointing to a dropdown menu showing '10'; 'sort' pointing to a double-headed arrow icon in the 'Id' header; 'search' pointing to a search input field; and 'Action Buttons' pointing to the edit, delete, and refresh icons in the first row's actions column.

Figure 4.2 Common Components FE/1

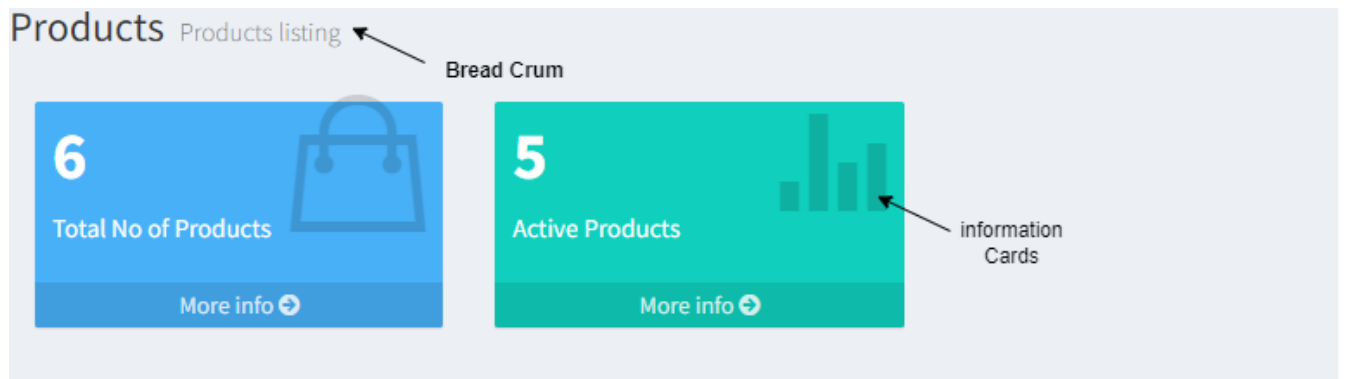


Figure 4.3 Common Components FE/2

Add Product
×

Product Name* :

Brand :

Rate* :

Status* :

product Order Id :

Figure 4.4 Common Form Components FE

4.4. Development

During Implementation Breakdown of common modules for both Front End and Back End are carried during the Project. Usage of Common modules is enhanced throughout the development of the project. Especially in this scenario, Backend Controllers, Sort, search Functionalities, models, Ajax Functions and Front End Styling with the common component is followed. Following are the common components used throughout the project.

Back End Common Modules Used

1. Data table
2. The Form Handling
3. Routing, Controller Handling
4. Sort Functionality
5. Search Functionality
6. Pagination functionality

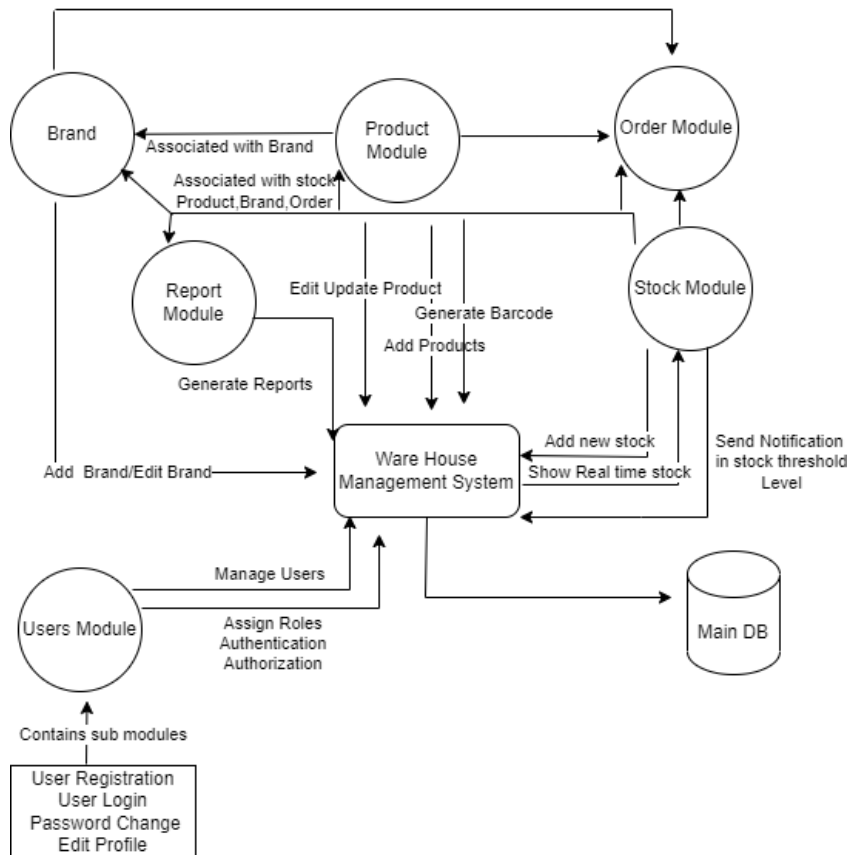


Figure 4.5 Interactions between Modules

Product module is associated with the following modules. Namely Brand Module, Stock Module, Order Module and Report Module.

With Brand Module, Product Module has many to many Relationships.

With Orders Module, Product Module has many to many Relationships.

With Stock Module, also have many to many Relationship with products.

When Order is processed, the product is shown if the stock is available. Furthermore, when the stock is dropped below the stock threshold value a notification is sent to Admin a Stock Manager.

Generation Barcodes for Products

For this use PHP Library. The Library is included in the composer.json file.

```
picqer/php-barcode-generator": "^2.2"
```

```
<form action="add-purchase.php" method="post">
    <input name="productId" onmouseover="this.focus();" type="text">
</form>
```

Generating barcodes display for printing

```
$db = new C_DataBase(PHPGRID_DB_HOSTNAME, PHPGRID_DB_USERNAME,
PHPGRID_DB_PASSWORD, PHPGRID_DB_NAME,
PHPGRID_DB_TYPE, PHPGRID_DB_CHARSET);

$results = $db->db_query($sql);
$data1 = array();
$count = 0;
while($row = $db->fetch_array_assoc($results)) {
    $data_row = array();
    for($i = 0; $i < $db->num_fields($results); $i++) {
        $col_name = $db->field_name($results, $i);
        $data1[$count][$col_name] = $row[$col_name];
    }
    $count++;
}
```

Reused code modules

Data Table is used to show the Inserted data into database. The Same Template is used in the entire project.

The reused Code to load data in the data table using ajax is shown below.

```
*/
public function dtajax()
{
    $values = DB::table('orders')->select($this->listing_cols)->whereNull('deleted_at');
    $out = Datatables::of($values)->make();
    $data = $out->getData();

    $fields_popup = ModuleFields::getModuleFields('Orders');

    for($i=0; $i < count($data->data); $i++) {
        for ($j=0; $j < count($this->listing_cols); $j++) {
            $col = $this->listing_cols[$j];
            if($fields_popup[$col] != null && starts_with($fields_popup[$col]->popup_vals, "@")) {
                $data->data[$i][$j] = ModuleFields::getFieldValue($fields_popup[$col], $data->data[$i]
                [$j]);
            }
            if($col == $this->view_col) {
                $data->data[$i][$j] = '<a href="'.url(config('laraadmin.adminRoute') . '/orders/'.
                $data->data[$i][0]).'">'.$data->data[$i][$j]. '</a>';
            }
            // else if($col == "author") {
            //     $data->data[$i][$j];
            // }
        }
    }
}
```

Edit, Delete Actions in Data Table

```
if($this->show_action) {
    $output = '';
    if(Module::hasAccess("Orders", "edit")) {
        $output .= '<a href="'.url(config('laraadmin.adminRoute') . '/orders/'.
        $data->data[$i][0].'/edit').'" class="btn btn-warning btn-xs"
        style="display:inline;padding:2px 5px 3px 5px;"><i class="fa fa-edit"></i></a>';
    }

    if(Module::hasAccess("Orders", "delete")) {
        $output .= Form::open(['route' => [config('laraadmin.adminRoute') . '.orders.destroy'
        , $data->data[$i][0]], 'method' => 'delete', 'style'=>'display:inline']);
        $output .= ' <button class="btn btn-danger btn-xs" type="submit"><i class="fa
        fa-times"></i></button>';
        $output .= Form::close();
    }
    $data->data[$i][] = (string)$output;
}
```

4.5. Technology used

As the first step, Laravel framework should be setup. Secondly, The DB connection was given with MYSQL. To set up the basic development Environment following Tools and Technologies were encountered.

1. Composer Version 2.3.10
2. PHP version 7.3.16
3. Laravel Version 5.3
4. picqer/php-barcode-generator": "^2.2
5. LaraAdmin : 1.0.40
6. Bootstrap
7. SASS
8. JavaScript
9. jQuery
10. Ajax
11. Font Awesome Cheatsheet

Referred Technical documents

1. Laravel 5.3 Official Documentation is used as the reference for main Technical documentation.

a. <https://laravel.com/docs/5.3> ,

2. LaraAdmin Official documentation is used as the reference for framework.

a. <https://laraadmin.com/docs/1.0>

4.6. Specifications of the hardware and software

4.6.1. Hardware Specifications

- CPU-i3 generation or above
- RAM-4GB or above
- Hard Disk minimum 100GB
- Router to get Internet Connection

4.6.2. Software Requirements

- Windows 10
- Web Server – Apache tomcat Latest Version
- Database Server – MYSQL
- PHP

4.7. Screenshots of the user interface

Login Page

Registered Users are login to the system from this form

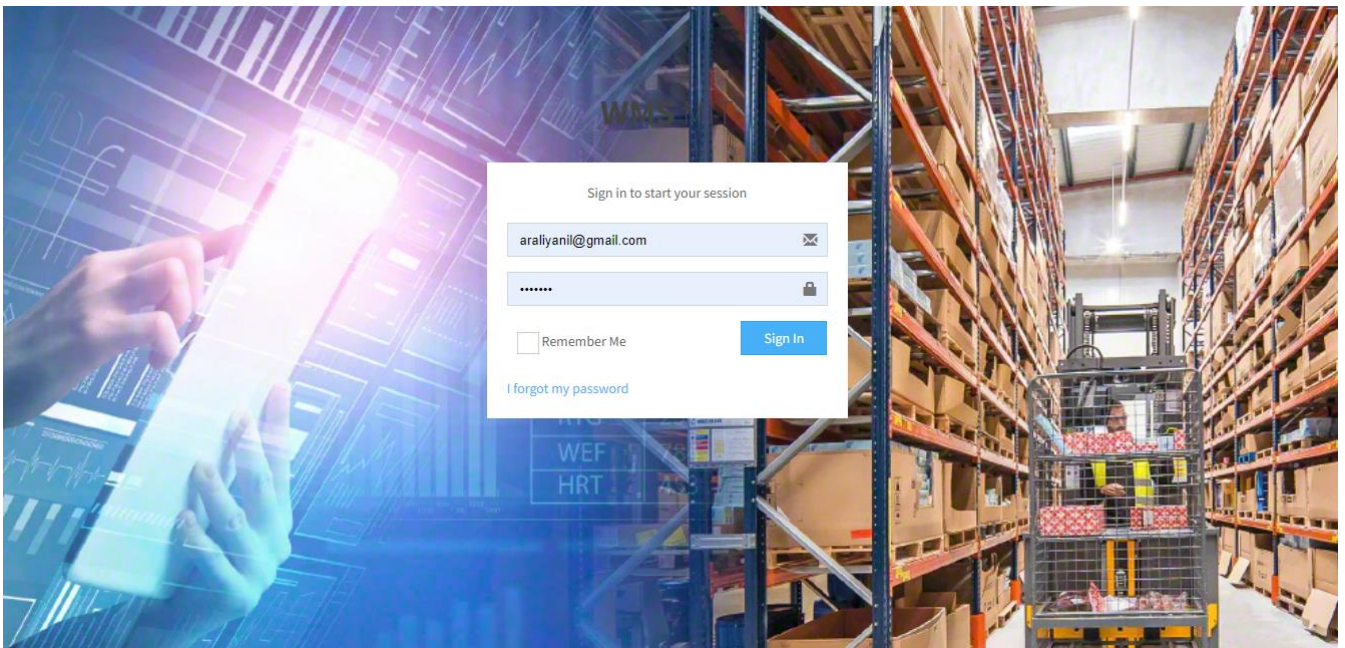


Figure 4.6 Login Page

Product Page

Once followed by a successful login user directed to this dashboard. Tabular Description of available products, Active Product count, Available product Count showed in this page.

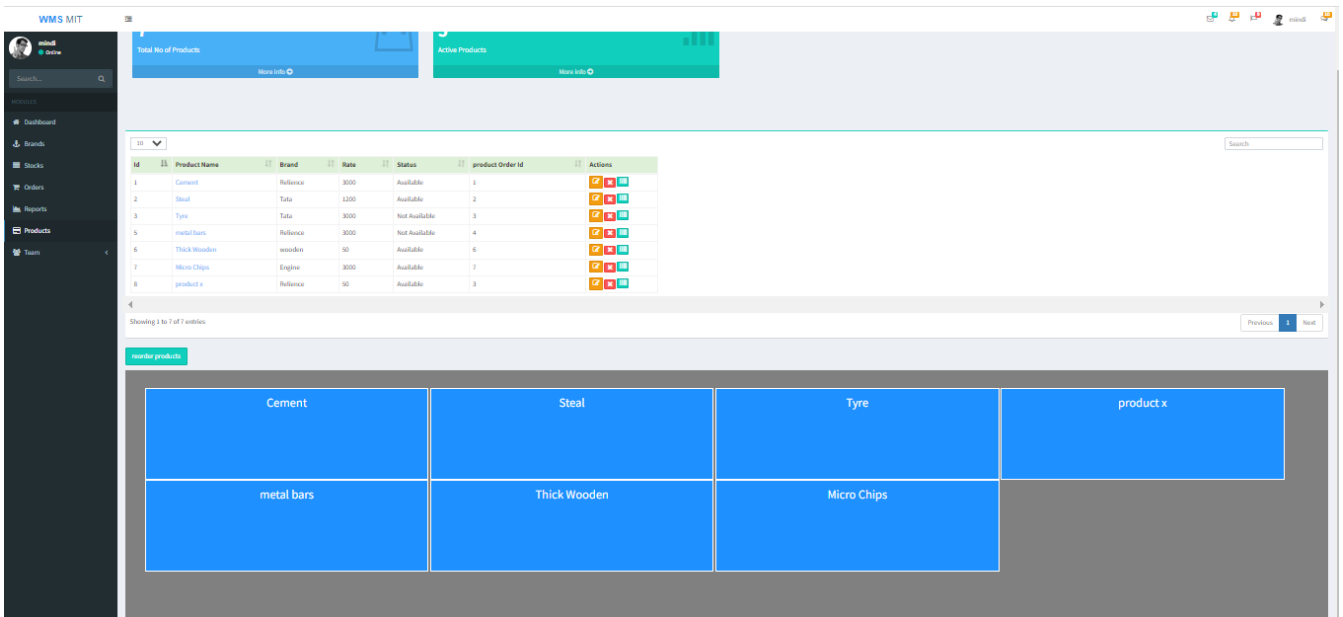


Figure 4.7 dashboard Admin Panel

Stock Page

Addition of Stock carried in this Page real time update of stock functionality is available when ordering. Moreover adding new stock functionality also present in this module

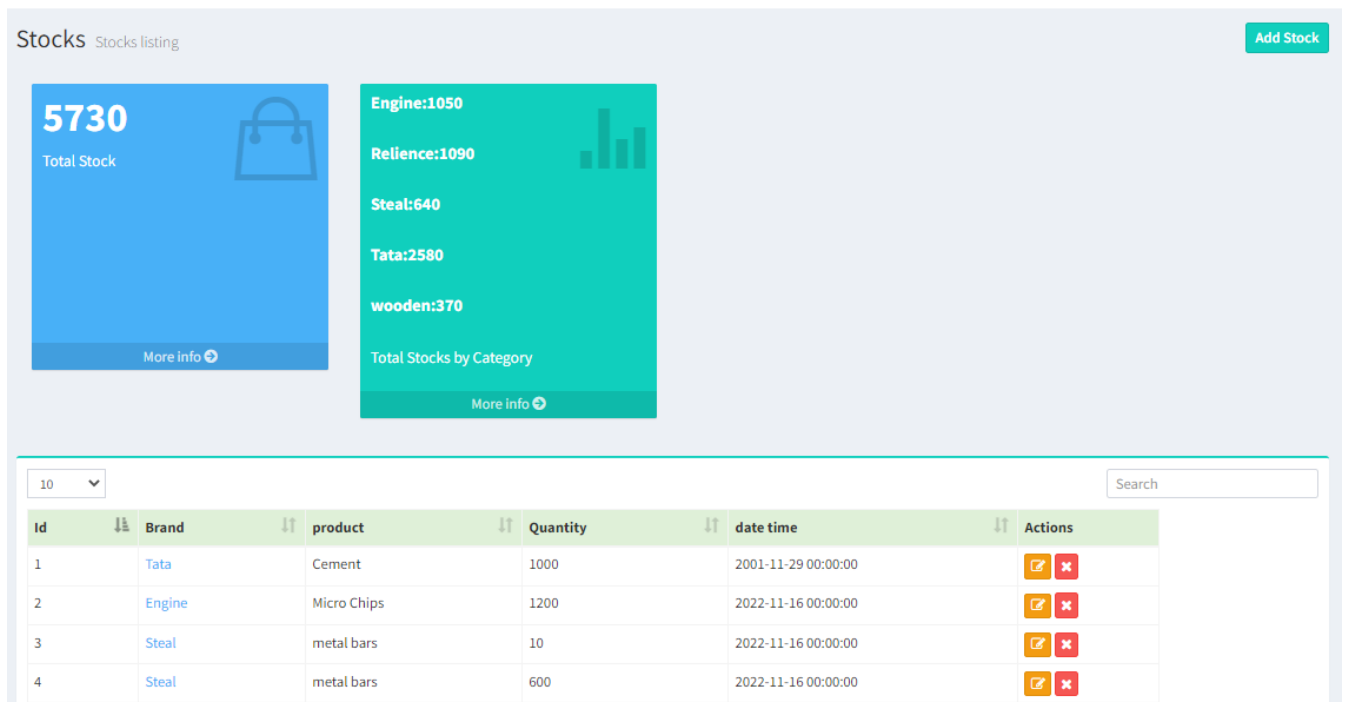


Figure 4.8 Stock Page

Brand Page

Available Brands with their count, Addition of new Brand Functionalities can done in this page

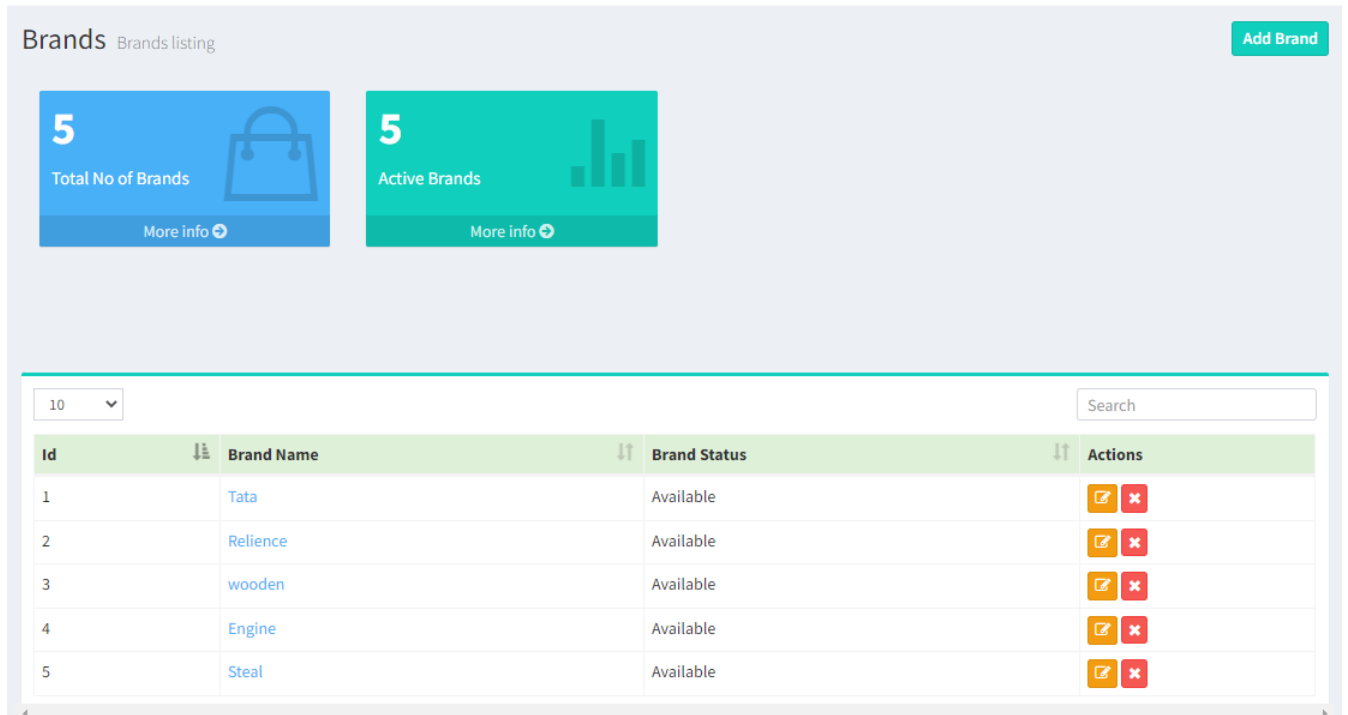


Figure 4.9 Brand Page

Report Page

Generation of Different Types of Reports with respect to Date can generate in this Report Page.

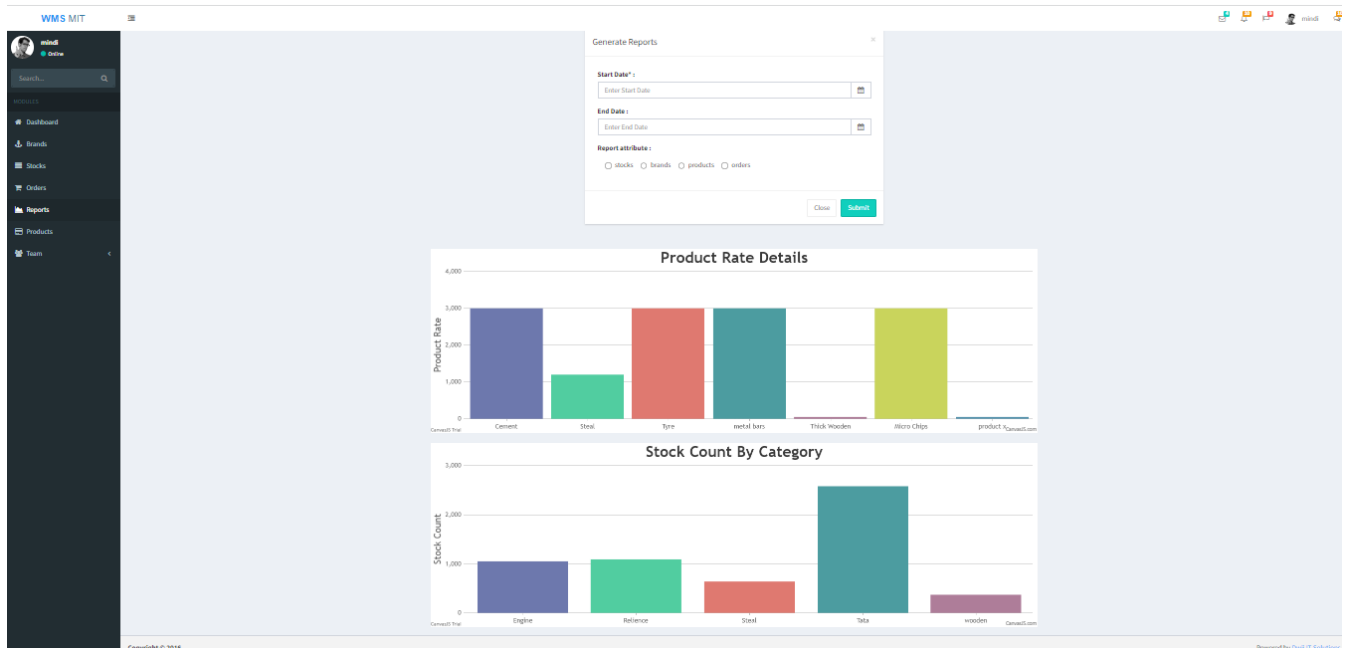
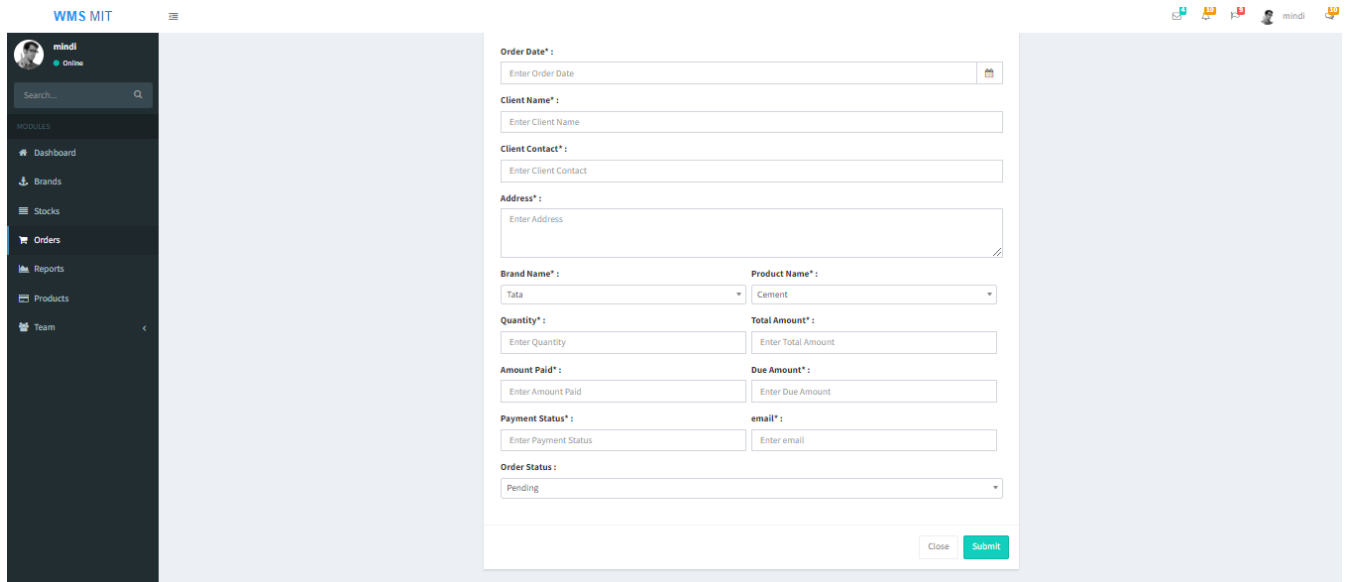


Figure 4.10 Report Page

Order Page

Placing Order carried in this Page



The screenshot displays the 'Order Page' in the WMS MIT system. On the left is a dark sidebar with navigation options: Dashboard, Brands, Stocks, Orders (highlighted), Reports, Products, and Team. The main content area contains a form with the following fields:

- Order Date* :** Enter Order Date
- Client Name* :** Enter Client Name
- Client Contact* :** Enter Client Contact
- Address* :** Enter Address
- Brand Name* :** Tata (dropdown)
- Product Name* :** Cement (dropdown)
- Quantity* :** Enter Quantity
- Total Amount* :** Enter Total Amount
- Amount Paid* :** Enter Amount Paid
- Due Amount* :** Enter Due Amount
- Payment Status* :** Enter Payment Status
- email* :** Enter email
- Order Status :** Pending (dropdown)

At the bottom right of the form are 'Close' and 'Submit' buttons.

Figure 4.11 Order Page

4.8. Summary

This chapter summarizes the techniques used to development. Technologies used and justification of those to design the system. Moreover, the hardware and software requirements for deployment discussed in detail.

Chapter 5 – Testing and Evaluation

The implemented work is undergo through continues unit testing by the developer before subject to integration testing, regression testing and system testing .Once a shippable single functionality is built with in the module, the unit testing is carried out to verify the requirement is achieved.

During this, various testing strategies followed to capture the work done along with the requirements defined in the design phase. One of the strategy used was Breakdown of entire

tasks in to Front tasks and Back end tasks. Through this test scope coverage is planned.

- Developer Testing (including both Front End and Back End)
- Front End Testing
- Back End Functionality Testing

Developer Testing (Unit Testing)

The development carried in a modular wise. Once the basic client shippable functionalities of a module is developed, the developer carry out unit testing. This is a full functionality testing. In this, verification of the functionalities, with the initial defined requirements is carried out. Not only that during this process, check page speed, resolving console errors also resolved and configured. Furthermore, Broken URLs also configured.

Front End Testing

Since the styling is done in modular wise, modules were identified for testing. For example, data Table styling is test only in one template page since other pages uses same template with same styling. Similarly, UI testing is done in a component-testing strategy.

Form validation testing also fall under Front End Testing. There validation for each form fields were tested. Along with that, appropriation of validation error messages were checked. Broken Image URLs, Broken URLs also tested under Front End Testing.

Back End Testing

This is mainly focus to functionalities of defined Back end tasks in each module. Best practice of back end Testing is to list down the backend tasks performed by the developer and convert them to test cases. In this project, Following Back End Functionalities were identified in modular wise. Then Testing techniques, such as state transition, Boundary value

Analysis, Decision Table based testing, Error Guessing were followed. Following sample test case could be shown which is used in this project.

5.1. Test Plan and Results

Sample Test Cases Related to Login, Register of Users. Other Test Cases related to login is annexed in appendix Table B.1

| TC ID | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results | Testing techniques to be used |
|---|--|--|--|-----------------------------------|----------------|--------------------------------|
| Create an Account | Admin should be able to create an account successfully | Admin should be in Create Account page | <ol style="list-style-type: none"> Admin Enter valid Email Address Click “Create Account Button” The User is directed to form with Personal Information Address details Click “Register” Button | Directed to “Users Listing ” Page | | State Transition |
| User should be able to login with valid credentials | User should be logged into the application | User is in the login page of the application | <ol style="list-style-type: none"> User enter a valid username User enters a valid password User click on the login button | | | Decision Table based testing |
| Register a Account | Add a First name , last name phone no and submit to register | User should be in create Account Page | Go to the Registration form Add a numbers and letters to the First Name and Last name | Validation Errors | DO | Equivalence Class Partitioning |

Table 5.1 Sample Test Cases for Login Module

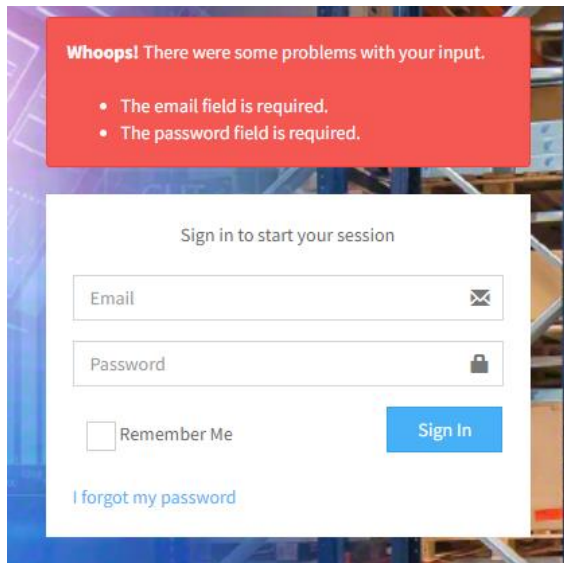


Figure 5.1 Validation for Errors Login

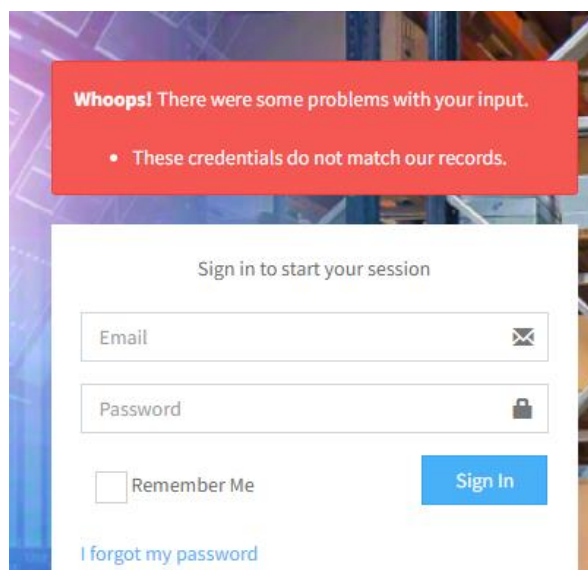


Figure 5.2 Validation for incorrect Credentials Login Module

Test Cases for Testing Data Table Functionality

| Id | Order Date | Client Name | Client Contact | Address | Brand Name | Product Name | Quantity | Total Amount | Amount Paid | Due Amount | Payment Status | email |
|----|------------|-------------|----------------|----------------|------------|--------------|----------|--------------|-------------|------------|----------------|------------------|
| 1 | 2022-11-17 | Mindi | 94706088380 | Dehiwala 10350 | Tata | Cement | 400 | 10000 | 3000 | 70000 | pending | Sw@gmail.com |
| 2 | 2022-11-17 | Mindi | 94706088380 | Kandy 20400 | Tata | Cement | 400 | 10000 | 3000 | 70000 | pending | Sw@gmail.com |
| 3 | 2022-11-17 | Mindi | 94706088380 | Kandy 20400 | Tata | Cement | 400 | 10000 | 3000 | 70000 | pending | Sw@gmail.com |
| 4 | 2022-11-17 | Mindi | 94706088381 | sdsds | Tata | Cement | 600 | 10000 | 3000 | 70000 | pending | Sw@gmail.com |
| 5 | 2022-11-17 | Mindi | 94706088381 | sewew | Tata | Cement | 10 | 5000 | 2000 | 3000 | pending | Sw@gmail.com |
| 6 | 2022-11-17 | Mindi | 94706088381 | egg | wooden | Thick Wooden | 30 | 5000 | 2000 | 3000 | pending | araliyanil@gmail |

Figure 5.3 Sample Test Cases for data Table Module

| TC ID | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results | |
|-------------------------|--|--|--|---|----------------|----------------|
| Testing search field | Search Results should be given according to the search term in the data tables | User should be in index page | <ol style="list-style-type: none"> 1. Go to “search form” 2. Type for a word 3. Hit Search button | Should view the search term matched results | | Error Guessing |
| Check for Sort Function | Sort function should work to ascending and descending order | User should be in data table containing page | Go to Orders in side menu | Should view the record according to the give sort order | do | Error Guessing |
| Check Pagnation | Page next ,Prev button should work as Ajax | User should be in a data table containing page | Click next and prev buttons in pagination | Should view the records with in pagination | do | Error Guessing |

Table 5.2 Sample Test Cases for Data Table Module

Test Cases for other Modules annexed in the Appendix B

5.2. Performance Testing

The performance testing has done with the use of google developer tool Page speed insights. This tool summarize the time taken to website first load, loading time for navigation , speed of repeat load etc. The suggestions were followed to optimize the first load , optimize the navigation speed and also for repeat load

Following actions were taken to increase the Page speed

- Removing of unnecessary. Scripts from header and footer
- Modification of written SQL Commands
- Modify JavaScript functions

5.3. Security Testing

Login implemented with password protected. Password policy was encouraged once a login created.

Authorization enhanced where delegation is different according to user roles. Defined roles are tested using dummy users and tested the login function. In the database, passwords protected with MD5.

5.4. Cross Browser Testing

Since mostly used open source web browser is google chrome it is used mainly for this task. The web based system was built in HTML5 and because of that Google chrome provides more support for HTML5 tags and Elements. Hence, Secondly the system was tested in Mozilla web browser and Microsoft Edge.

5.5. Usability and Evaluation

5.5.1. Criteria to select Users

Ten users have selected for the evaluation of usability. The cluster is composed of people with IT sound knowledge, People with average IT knowledge, and Employees who work with WMS systems.

5.5.2. Methodology Used

Questionnaire given to evaluate usability. These questionnaire is simple and closed type. Feedback collected through google form.

5.5.3. Area Focused

UI and UX Experience focused when creating questions for the Evaluation. When utilizing the system, the user should take into account the following: Login Form, Dashboard, Forms (product, brand, stock, order) Look and Feel, Navigation, Consistency, Look and Feel, and Satisfaction with the Entire System.

Results obtained summarized as follows.

| Question # | Feature | Rating | | |
|------------|-----------------------------------|--------|---------|------|
| | | Bad | Average | Good |
| 1 | Look and Feel | 0 | 3 | 7 |
| 2 | Consistency of the UI | 0 | 0 | 10 |
| 3 | UX of Form Submission | 0 | 9 | 1 |
| 4 | UX of Search Functionality | 0 | 10 | 0 |
| 5 | UX of placing an Order | 0 | 7 | 3 |
| 6 | UX of Adding a stock | 0 | 0 | 10 |
| 7 | Navigation | 0 | 6 | 4 |
| 8 | Total Satisfaction for the system | | 4 | 6 |

Table 5.3 Sample Test Cases for Data Table Module

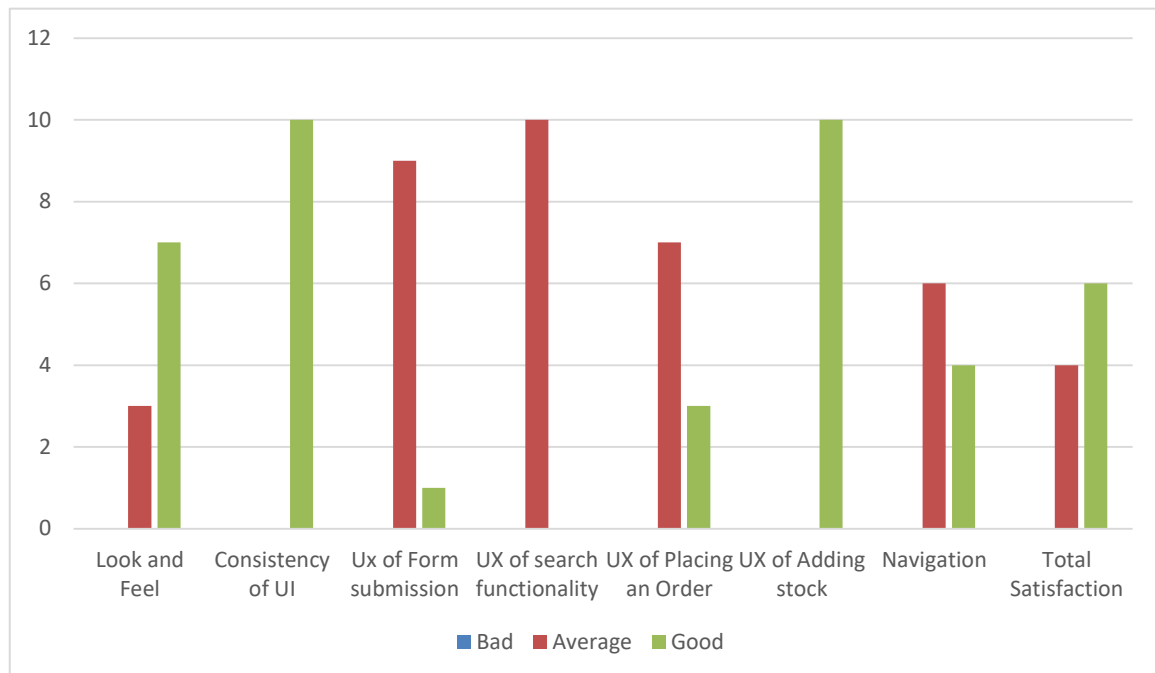


Figure 5.4 User Evaluation Analysis

5.6. System Evaluation

To demonstrate that the system possesses the necessary properties to operate in the specified context, a system evaluation was undertaken. These qualitative characteristics have been used as standards for assessing systems. Both the system's design and operations have been reviewed.

Chapter 6 – Conclusion

The Warehouse Management system intended to build to mitigate current loopholes in WMS system. Specially to increase the efficiency of WMS processes. Developing visually arranging the WMS store is one of the success factor in this project. This will eliminate extra time need to search places for products in the store.

To manage orders effectively and avoid double handling errors, orders, stocks, products and customer data must be synchronize in one system. This achieved in the current system by managing stocks, orders, products centrally. Sending Alerts or notifications when the stock is low

Automating from classic spreadsheets to Efficient Stock management systems delivers the needed clarity centralizing all the data in a single location through an inventory management system.

6.1. Achievement of Objectives

Objective 1

Organize the Warehouse Layout Visually (2D Visualization) through the Software.

This was achieved successfully in the implemented system. The administrator could view the layout in 2D in real time and arrange the space accordingly.

Objective 2

Increase, Profitability per Order.

The system provides Search functionality, sort functionality and generation of reports and graphs. This will eliminate the time required to search available stocks, orders etc which will ultimately save time and increase the profitability.

Objective 3

Reduce the phone Communication Cost.

Since the records need can be view real-time with the help of search sort, reports , graphs functions the

telephone conversation have drastically reduced.

Objective 4

Generate MIS Reports

Real time report generation is implemented in the system. Where those could generate with respect to specific date.

6.2. Future Work

Potential Future work may focus in to Labor Management inside the warehouse management System. Supervising the human side of warehouse operations entails the use of a labor management system integrated into WMS. This allows you to track productivity, identify underperforming shifts or workers.

6.3. Knowledge gained

This project has provided vital knowledge. From the development phase to the feasibility studies. The project offered the chance to put to use and test the majority of the concepts and methods discovered during the MIT degree program.

Additionally, it inspired them to master cutting-edge technologies that are actually fascinating in order to boost system performance. A special effort has made to practice MVC design.

It was an opportunity to work with numerous people from many fields while also developing technical and intellectual capabilities. The most crucial elements of time management and planning discovered throughout this endeavor. The work completed on time.

Appendix A - References

Andiyappillai, N and Dr. Prakash, T.2019.*Implementing Warehouse Management Systems (WMS) in Logistics*

Assis, R and Sagawa, J.K.2018.*Assessment of the implementation of a Warehouse Management System in a multinational company of industrial gears and drives, Gestão & Produção*

Dekker, R., de Koster, R., Roodbergen, K.J. and Van Kalleveen, H., Improving order-picking response time at Ankor's warehouse. *Interfaces*, 2004, 34(4), 303-313.

Elsayed, E.A., Algorithms for optimal material handling in automatic warehousing systems. *International Journal of Production Research*, 1981, 19(5), 525-535.

Elsayed, E.A. and Lee, M.K., Order processing in automated storage/retrieval systems with due dates. *International Journal of Production Research*, 1996, 28(7), 567-577.

Elsayed, E.A. and Stern, R.G., Computerized algorithms for order processing in automated warehousing systems. *International Journal of Production Research*, 1983, 21(4), 579-586.

Graves, S.C., Hausman, W.H. and Schwarz, L.B., Storage-retrieval interleaving in automatic warehousing systems. *Management Science*, 1977, 23, 935-945.

Kearney, A.T, 2004. *Excellence in Logistics Brussels*: ELA.

Langevin, A., and Ripopel, D, 2005. *Logistics Systems: Design and Optimization*. New York: Springer.

Hall, R.W., Distance approximation for routing manual pickers in a warehouse. *IIE Transactions*, 1993, 25, 77-87.

Hausman, W.H., Schwarz, L.B. and Graves, S.C., Optimal storage assignment in automatic warehousing systems. *Management Science*, 1976, 22(6), 629-638. r

Min, H.2006, *The applications of warehouse management systems*, International journal of Logistics Research and Applications, 6 (2), p111 – 126.

Miralam, M. 2017. *Impact of Implementing Warehouse Management System on Auto Spare Part Industry Market in Saudi Arabia*, Review of Integrative Business and Economics Research, 6(3), [online] Available at <http://buscompress.com/journal-home.html>

Pyeman and Zakariah. 2013. *Logistics Cost Accounting and Management in Malaysia: Current State and Challenge*, International Journal of Trade,4(3), p120-121.

Ramaa, A, Subramanya, K.N. and Rangaswamy, T.N. 2012.*Impact of Warehouse Management System in a Supply Chain*, International Journal of Computer Applications, 54 (1), p14 – 20.

Appendix A Usability Evaluation Questionnaire

This survey has carried out for the project. The information it collects is maintained confidential. It has only used for the project.

Your age.....

Your profession.....

Your answer to the questions below:

1. The System Overall look and Feel
 - a. Bad
 - b. Average
 - c. Good
 2. How do you feel about the Navigation of the System
 - a. Bad
 - b. Average
 - c. Good
 3. Do you understand about the system
 - a. Bad
 - b. Average
 - c. Good
 4. Consistency of the system
 - a. Bad
 - b. Average
 - c. Good
 5. Are you satisfied with the overall system?
 - a. Bad
 - b. Average
 - c. Good
 6. How much level of Easy was to fill forms?
 - a. Bad
 - b. Average
 - c. Good
 7. What is your feedback of the system?
 - a. Bad
 - b. Average
 - c. Good
-

Appendix B -


s

Add Stock ×

Brand* :

product* :

Quantity :

date time* :
 
This field is required.


Expiry Date :
 

Figure B.1 Add Stock Form

Stocks Stocks listing

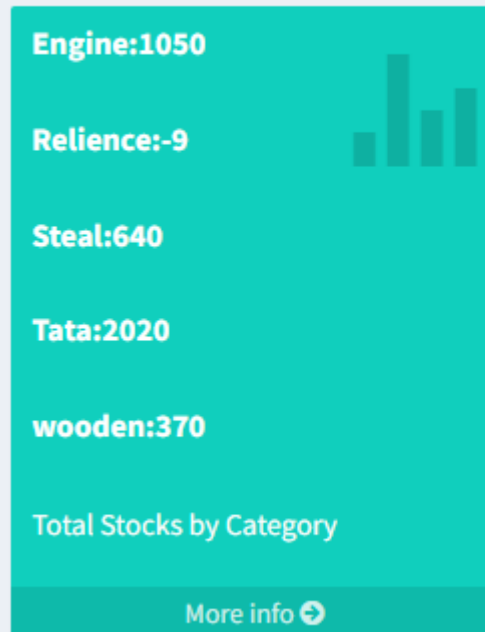
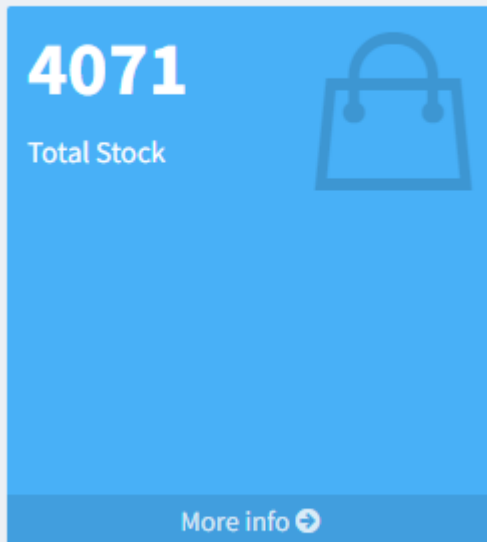


Figure B.2 Stock Real Time Detail Card

| TC ID | TC | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results |
|-------|---|--|------------------------------|--|---|----------------|
| 1 | Testing form Validation | Fill input one by one and hit "Add" Button | User should be in Stock Page | <ol style="list-style-type: none"> Go to "Add Stock form" Insert data form field one at a time Hit Add button | Should generate Validation Errors for required fields | do |
| 2 | Check for displaying values for Total Stock | Add a new stock with given category And Check the category | User should be in Stock Page | <ol style="list-style-type: none"> Go to Add stock. Add a Stock Check the count | Count should be increase for added category | do |

| | | | | | | |
|---|--|---|------------------------------|---|--|----|
| | | count Increases | | | | |
| 3 | Check for displaying values for Total Stock in RS | Add a new stock with given category And Check the total stock count Increases | User should be in Stock Page | <ol style="list-style-type: none"> 1. Go to Add stock. 2. Add a Stock 3. Check the total Amount in Rs increases | Amount should be increase for added stock | do |
| 4 | Check Edit Stock | Click on Edit Button for a selected stock | User should be in Stock Page | <ol style="list-style-type: none"> 1. Go to the Entry in the stock table. 2. Select one Entry 3. Click Edit Button 4. Make necessary changes. 5. Hit "Save" Button | The Modified data should be show in the Table | do |
| 5 | Check Delete Stock | Click on Delete Button for a selected Stock | User should be in Brand Page | <ol style="list-style-type: none"> 1. Go to the Entry in the Stock table. 2. Select one Entry 3. Click Delete Button | The Deleted data should be remove from the Table | do |

Table B.1 Test Cases for Stock Module

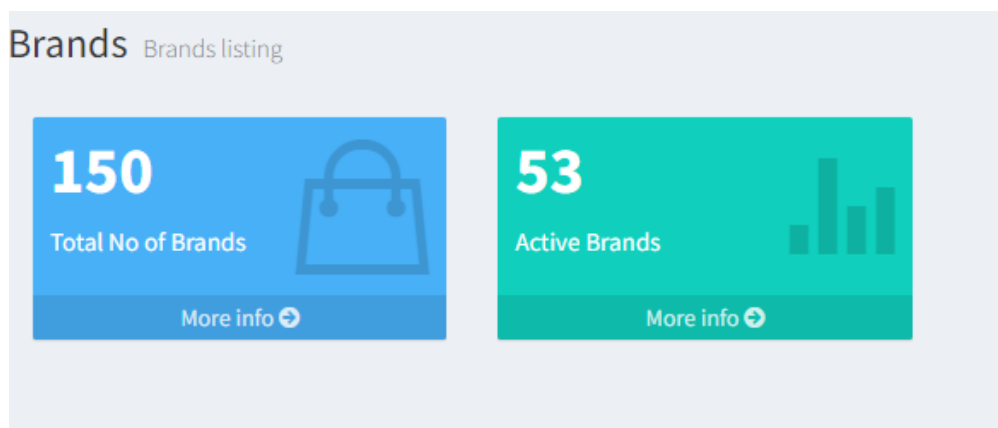


Figure B.3 Real Time Brand Detail Cards

| TC Id | TC Case | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results |
|-------|--|---|------------------------------|--|--|----------------|
| 1 | Testing form Validation | Fill input “Brand“ and click on Add” Button | User should be in Brand Page | <ol style="list-style-type: none"> 1. Go to “Add Brand form” 2. Insert data form field Brand 3. Hit Add button | Should generate Validation Errors for Brand fields | do |
| 2 | Check for displaying count values for Total No of Brands | Add a new Brand. And Check the Brand count Increases | User should be in Brand Page | <ol style="list-style-type: none"> 1. Go to Add Brand. 2. Add a Brand 3. Check the count | Count should be increase when Brand is added | do |
| 3 | Check for displaying Count for Availability of Brands | Add a new Brand with status “Available”. And Check the Brand Availability count Increases | User should be in Brand Page | <ol style="list-style-type: none"> 1. Go to Add Brand. 2. Add a Brand 3. Check the count | Count should be increase when Brand is added | do |
| 4 | Check Edit Brand | Click on Edit Button for a selected Brand | User should be in Brand Page | <ol style="list-style-type: none"> 1. Go to the Entry in the Brand table. 2. Select one Entry 3. Click Edit Button 4. Make necessary changes. 5. Hit “Save ” Button | The Modified data should be show in the Table | do |
| 5 | Check Delete Brand | Click on Delete Button for a selected Brand | User should be in Brand Page | <ol style="list-style-type: none"> 1. Go to the Entry in the Brand table. 2. Select one Entry 3. Click Deletet Button | The Deleted data should be remove from the Table | do |

Table B.2 Test Cases for Brand Module

| TC Id | TC Case | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results |
|-------|--|---|--------------------------------|--|---|----------------|
| 1 | Testing form Validation | Fill input “Product“ and click on Add” Button | User should be in Product Page | <ol style="list-style-type: none"> 1. Go to “Add Product form” 2. Insert data form field Product 3. Hit Add button | Should generate Validation Errors for required fields | do |
| 2 | Check for displaying count values for Total No of Products | Add a new Product. And Check the Product count Increases | User should be in Product Page | <ol style="list-style-type: none"> 1. Go to Add Product. 2. Add a Product 3. Check the count | Count should be increase when Product is added | do |
| 3 | Check for displaying Count for Availability of Product | Add a new Product with status “Available”. And Check the Product Availability count Increases | User should be in Product Page | <ol style="list-style-type: none"> 1. Go to Add product. 2. Add a Product 3. Check the count | Count should be increase when Product is added | do |
| 4 | Check Edit Product | Click on Edit Button for a selected Product | User should be in Product Page | <ol style="list-style-type: none"> 1. Go to the Entry in the Product table. 2. Select one Entry 3. Click Edit Button 4. Make necessary changes. 5. Hit “Save ” Button | The Modified data should be show in the Table | do |
| 5 | Check Delete Product | Click on Delete Button for a selected Product | User should be in Product Page | <ol style="list-style-type: none"> 1. Go to the Entry in the Product table. 2. Select one Entry 3. Click Delete Button | The Deleted data should be remove from the Table | do |

Table B.3 Test Cases for Product Module

save reordering

1.drag item to reorder.
2.click 'save Recording' when finished.

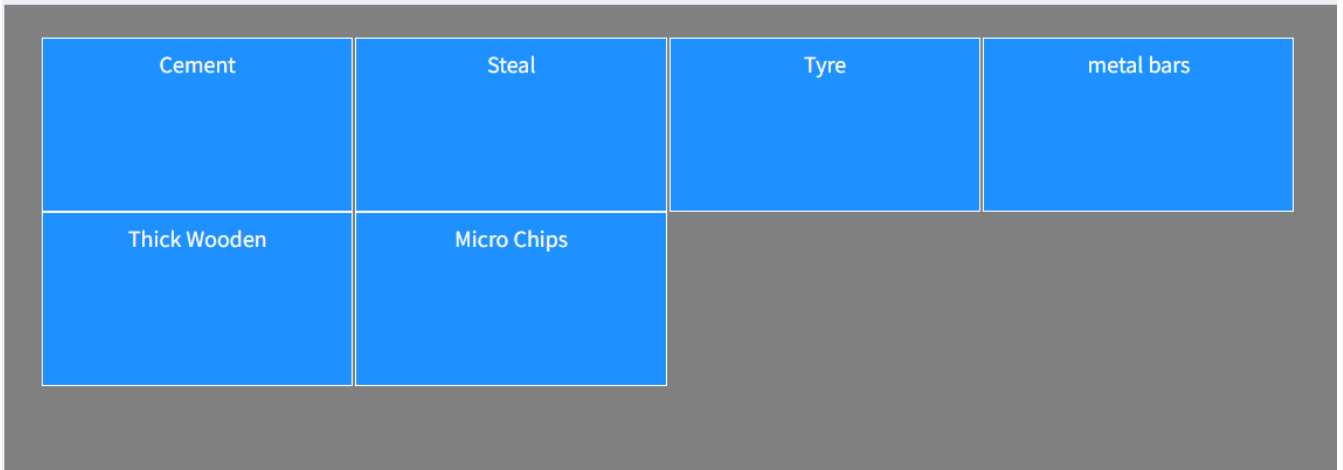


Figure B .4 Product Reorder Layout

| TC Id | TC Case | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results |
|-------|---|--|---|---|---|----------------|
| 6 | Check Reorder functionality of Products in 2D Visualization | Click on “Save Reorder” Button | User should be in Product Page and in Reorder product Panel | 1.Click “Save Reorder Button” 2.Drag and Drop Product Cards 3.Click “Save Reorder” Button 4.Refresh the page to see the changers are applied | The Modified Layout should be show in the Table | do |
| 7 | Check Newly Added Product is added to the Reorder Panel | Check the New Products is added to the Reorder frame | User should be in Product Page and in Reorder product Panel | 1. Add a Product From by clicking “Add Product form. 2. Check the added product is added in to the Layout | The New Product should be in the layout | do |

Table B.4 Test Cases for Product Reorder Layout

| TC Id | TC Case | Test Objective | Pre-Conditions | Steps | Expected Results | Actual Results |
|-------|-------------------------|---|------------------------------|--|---|----------------|
| 1 | Testing form Validation | Fill input fields and click on Add" Button | User should be in Order Page | <ol style="list-style-type: none"> 1. Go to "Add Order form" 2. Insert data in to form fields 3. Hit Add button | Should generate Validation Errors for required fields | do |
| 4 | Check Edit Order | Click on Edit Button for a selected Order | User should be in Order Page | <ol style="list-style-type: none"> 1. Go to the Entry in the Order table. 2. Select one Entry 3. Click Edit Button 4. Make necessary changes. 5. Hit "Save " Button | The Modified data should be show in the Table | do |
| 5 | Check Delete Order | Click on Delete Button for a selected Order | User should be in Order Page | <ol style="list-style-type: none"> 4. Go to the Entry in the Product table. 5. Select one Entry 6. Click Delete Button | The Deleted data should be remove from the Table | do |

Table B.5 Test Cases for order Module