Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT REPORT ON

STOCK MANAGEMENT SYSTEM

Submitted to

Department of Computer Application

Bajra International College

In partial fulfilment of the requirements for the Bachelors in Computer Application

Submitted by

Name: Milan Karki

Roll no: 4

Tu Register No: 6-2-712-52-2019

04-07-2081

Tribhuvan University

Faculty of Humanities and Social Sciences

Bajra International College

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by MILAN KARKI entitled "STOCK MANAGEMENT SYSTEM" in partial fulfilment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

Signature of the Supervisor

Mr. Er. Birendra Manandhar

Lecturer, BCA

Bajra International College

Tribhuvan University

Faculty of Humanities and Social Sciences

Bajra International college

LETTER OF APPROVAL

This is to certify that this project prepared by MILAN KARKI entitled "STOCK MANAGEMENT SYSTEM" in partial fulfilment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

………………………………… Mr. Er. Birendra Manandhar Supervisor BAJRA INT College	………………………………… Mr. Deepak Kumar Thakur Chief Academic Officer BAJRA INT College
………………………………….. Mr. Ananda KC External Examiner Tribhuvan University	………………………………….. Mr. Dr. Shyam Kumar Adhikary Internal Examiner BAJRA INT College

ACKNOWLEDGEMENT

This project has been prepared for the partial fulfillment of the requirement for BCA Eight Semester PROJECT III course designed by TU.

I would like to express our deepest appreciation to everyone who provided us the possibility to complete this report. I would like to extend my special gratitude to my project coordinator **Mr. Deepak Thakuri** whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report. Furthermore, I would like to appreciate the guidance given by my project supervisor **Mr. Birendra Manandhar** who have provided proper guidance and invested his full effort in guiding me in achieving the goal. I would also like to acknowledge with much appreciation the crucial role of the teachers who gave the permission to use all required equipment and the necessary materials to complete the **PROJECT III**. A special thanks goes to my mates who helped me to assemble the parts and aided in the successful completion of this project with effective team work. I have to appreciate the guidance given by other supervisor as well as the panels especially in my project presentation that has improved my presentation skills thanks to their comment and advices.

Thank You!!!

Name: Milan Karki

Tu Register No: 6-2-712-52-2019

ABSTRACT

Many businesses still use manual systems where staff record stock by hand, leading to problems like forgetting to update inventory, which causes delays in ordering items. Keeping physical ledger books is also inconvenient, as they must be available at all times to make updates. If the ledger is not on hand, stock information can't be updated. This proposed system solves these issues by using a web-based platform. It allows businesses to create and manage Purchase Orders, Sales Orders, Return Lists, and Invoices with ease. The system also keeps track of Remaining Stock in real time, helping users know exactly what's available. A heatmap feature gives a visual overview of stock movement, and analytics tools provide insights into sales and performance. Additionally, the bulk email feature allows easy communication with customers or suppliers. By using this technology, the system makes stock management faster, more accurate, and more efficient.

Keywords: *Stock Management System*; *Purchase Order; Sales Order; Return List; Remaining Stock; Invoice; Heatmap, Analytics, Login.*

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

BO: Back Order

CSS: Cascading Style Sheet

HTML: Hyper Text Markup Language

MySQL: My Structured Query Language

PHP: Hypertext Preprocessor

PO: Purchase Order

SMS: Stock Management System

CHAPTER 1: INTRODUCTION

1.1 Introduction

Stock Management is the practice of ordering, storing, tracking, and controlling inventory Stock Management applies to every item a business uses to produce its products or services from raw materials to finished goods. In other words, stock management covers every aspect of a business's inventory. Stock Management System helps in the efficient monitoring of constant flow of units into and out of an existing inventory. This process usually involves controlling the transfer in of units in order to prevent the inventory from becoming too high, or to low so that the operation of the company into difficulties [3].

Stock Management System is very important organizations especially where there are a lot of orders are being placed every day and there are lot of materials and the maintenance is really important which the system will do and also will record the time taken to process an order and this system is really important as it can help the organizations to be alerted when the level of inventory is very low and focuses on the three aspects of inventory management and prevent from failures in the future [7]. Stock Management System also demands a solid understanding of how long it will take for those materials to transfer out of the inventory to be established. By Knowing these two important lead key aspects makes it possible to know when to place an order and how many units must be ordered to keep production running smoothly [3].

Nowadays, many companies use the system to avoid overstock, miscount and outages. It is a system for organizing a better inventory data than that was used before which is generally stored in manual form books or in spreadsheets. This application has admin component to manage the inventory and maintenance of the inventory system. This System has general organization profile, stock details, purchase details and the remaining stock that are presented in the organization [5]. This System also provides the remaining balance of the stock as well as the details of the balance of transaction. Each new stock is created and entitled with the named and the entry date of that stock and it can also be updating any time when required as per the transaction or the sales is returned in case. Here the login page is created to protect the management of the stock of organization in order to prevent it from the threads and misuse of the inventory.

1.2 Problem Statement

Many businesses still manage their stock by hand, which can lead to mistakes like forgetting to update inventory or missing important information. Relying on physical books to track stock is also inconvenient because they need to be with staff all the time, and if they're not available, updates can't be made. This causes delays in ordering new stock and creates confusion about what's available.

Additionally, the manual system makes it hard to track when stock is running low. Without reminders, staff may forget to reorder items in time, leading to stockouts and disruptions in operations. The current system is slow, prone to errors, and not efficient for managing inventory or reordering stock.

1.3 Objective

1. To design and develop a user-friendly system that effectively handles and manages information about items or products, while also providing calculations for the information system.

2. To develop a system that deals with the day-to-day needs of organization like managing purchase, sales, return, and available stocks.

3. Keep each and every calculation and help to generate reports of transactions in Excel, PDF format with addition of Email and Printing Features as well.

4. Additionally, the System will Show Visual representation like Charts and Heat Map with addition of Advance Algorithms like Rule Based, ABC Inventory and Safety Stock Check Algorithms.

1.4 Scope and Limitations

1.4.1 Scope

A stock management system is designed to help businesses efficiently track and manage their inventory levels. The scope of this system includes:

1. All authorized users with proper access credentials can use this system to manage Stock.
2. The system allows users to easily input, track, and update stock levels.
3. The system offers comprehensive reports and analytics, enabling users to make well-informed decisions regarding purchasing and Stock management.
4. The system accurately calculates and maintains records of transactions, facilitating the generation of reports in Excel and PDF formats. This feature ensures that users have access to detailed transactional information.

1.4.2 Limitations

While a stock management system can provide many benefits to businesses, there are also some limitations to consider. These include:

1. The accuracy of inventory levels depends on the accuracy of data input by users.
2. The system may not be able to integrate with all other business software, depending on compatibility.
3. The system's functionality may vary depending on the specific industry or type of business.
4. The system has limited amount of category.

1.5 Development Methodology

For the development of the Stock Management System (SMS), we followed the **Waterfall** model. This approach is straightforward, where each phase of the project is completed before moving to the next. It allowed us to carefully plan and develop the system in clear steps, ensuring that everything was done in order.

The process was broken down as follows:

i. Requirement Gathering and Analysis

- We started by researching different stock management systems and gathered input from users to understand what the system needed to do.

- After collecting this information, we created a list of all the system requirements, like how it should manage stock levels, purchase orders, and backorders.

ii. System Design

- We designed diagrams to show how data would move within the system Object and Class Diagram, Dynamic Modelling: State and Sequential Diagram, Process Modelling: Activity Diagram design of the system with Component Diagram, and Deployment Diagram, Refinement of Classes and Object, the implementations of Algorithm and planned out the database structure.
- The database, using MySQL, was set up to manage information about products, suppliers, and stock levels.
- We also designed a simple and easy-to-use interface for users, using HTML and CSS.

iii. Implementation

- The frontend (user interface) was built with HTML, CSS, and JavaScript to allow users to easily manage stock, purchase orders, and backorders.
- On the backend, PHP and MySQL were used to handle all the main functions of the system, like updating stock and generating purchase orders automatically when stock levels are low.

iv. Integration and Testing

- After building the individual parts of the system, we integrated everything to make sure it worked together smoothly.
- We ran tests on each part (unit testing) and the system as a whole (system testing) to ensure everything functioned properly, even under heavy use.

1.5 Report Organization

Chapter 1 Deals with the introduction of the system with its objectives and limitations along with the reason why the system is made.

Chapter 2 Summarizes the work that has been carried out in the field of Stock management and also describes the features about some existing applications related to the Stock Management Systems.

Chapter 3 Focuses on the different requirement of the system, which describes about the functional, non-functional, feasibility analysis, Object Modelling: Object and Class Diagram, Dynamic Modelling: State and Sequential Diagram, Process Modelling: Activity Diagram design of the system with Component Diagram, and Deployment Diagram, Refinement of Classes and Object, and the implementations of Algorithm with its details.

Chapter 4 Emphasizes tools used in system development, implementing details and result of test performed.

Chapter 5 Highlights brief summary of lesson learnt, outcome and conclusion of the whole project and explain what have been done and what further improvements could be done.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

In today's world the use and access to the internet is so high so most of the people are busy on their own work so we have developed this module so that user can create and access to their account through the use of internet and general concept and terminologies are mentioned below.

1. User Interface: The system should have a user-friendly interface that allows users to easily navigate through the system and perform various actions such as adding or editing stock items, generating reports, and managing inventory levels.

2. Dashboard: The system should have a dashboard that provides a quick overview of stock levels, sales trends, and other important metrics.

3. Stock Tracking: The system should be able to track stock levels for each item, including current stock levels, incoming stock, and outgoing stock.

4. Purchase Order Management: The system should allow users to create and manage purchase orders for stock items, including tracking the status of each order.

5. Sales Order Management: The system should allow users to create and manage sales orders for stock items, including tracking the status of each order.

6. Reporting: The system should provide a range of reports, including stock levels, sales trends, and inventory turnover, to help users make informed decisions.

7. Integration: The system should be able to integrate with other systems such as accounting software and e-commerce platforms, to streamline business processes.

8. Security: The system should have robust security measures in place to protect sensitive data such as stock levels, purchase orders, and sales orders. This may include user authentication, data encryption, and role-based access control.

9. Scalability: The system should be scalable to meet the needs of growing businesses, with the ability to handle larger volumes of stock.

2.2 Literature Review

A study of an existing system for stock management involves analyzing the current system that a business is using to manage its inventory. The purpose of this study is to identify any weaknesses in the current system and suggest improvements that can be made to enhance the efficiency and accuracy of stock management. The study may involve analyzing various aspects of the current system, such as the user interface, the stock tracking mechanism, the purchase order and sales order management, and the reporting capabilities.

During the study, it is essential to gather feedback from the users of the current system to identify any pain points they are experiencing. This can be achieved through surveys, interviews, and observations of the users in action. The feedback gathered can provide valuable insights into the current system's strengths and weaknesses and can be used to inform the development of a new system or the improvement of the existing one.

The study may also involve analyzing the data generated by the current system to identify patterns and trends in stock levels, sales, and inventory turnover. This data can be used to optimize the stock management process by identifying areas where improvements can be made, such as reducing excess stock or improving order fulfillment times.

Ultimately, the goal of studying an existing system for stock management is to identify opportunities for improvement that can help the business run more efficiently and effectively. By gathering feedback from users, analyzing data, and identifying pain points, businesses can develop a better understanding of their stock management needs and create a new system or improve the existing one that meets those needs.

After conducting research on existing stock management systems, we have identified a system called TradeGecko that provides a comprehensive solution for managing inventory, sales, and purchasing. TradeGecko offers a cloud-based platform that allows users to manage their stock levels in real-time, track orders and shipments, and generate reports on inventory performance.

The TradeGecko system includes features such as inventory tracking, order management, purchase orders, sales orders, and reporting. It also integrates with popular e-commerce platforms, accounting software, and shipping providers, making it a versatile solution for businesses of all sizes [1].

One of the key benefits of TradeGecko is its user-friendly interface, which makes it easy for users to navigate the system and perform various actions such as adding or editing stock items, generating reports, and managing inventory levels which allows users to easily add and update stock items, making stocktaking and stock tracking more accurate and efficient [1].

TradeGecko also offers a range of reporting features, including stock levels, sales trends, and inventory turnover, which provide valuable insights into business performance. The system is scalable and can handle large volumes of stock items, purchase orders, and sales orders, making it suitable for businesses that are growing rapidly [4].

SAP ERP Inventory Management: SAP ERP is a comprehensive enterprise resource planning system that includes inventory management functionality. This system allows businesses to track inventory levels, monitor product movement, and optimize replenishment processes. SAP ERP also includes features for managing warehouse operations, such as picking, packing, and shipping [7].

QuickBooks Online Inventory Management: QuickBooks Online is a cloud-based accounting and inventory management system designed for small and medium-sized businesses. This system allows businesses to track inventory levels, create purchase orders, and manage sales orders. QuickBooks Online also includes features for managing customers, vendors, and financial transactions [6].

Zoho Inventory: Zoho Inventory is a cloud-based inventory management system that allows businesses to track inventory levels across multiple channels, including e-commerce platforms and third-party marketplaces. This system also includes features for managing purchase orders, sales orders, and shipping operations. Zoho Inventory integrates with other Zoho applications, such as Zoho CRM and Zoho Books [6].

Fishbowl Inventory: Fishbowl Inventory is an inventory management system designed for small and mid-sized businesses. This system includes features for managing inventory levels, tracking product movement, and generating reports. Fishbowl Inventory also includes features for managing manufacturing operations, such as bill of materials and work orders [6].

Literature review is the formal methods that can be used to review the critical points of current knowledge including findings as well as theoretical and methodological particular topic for supporting issues. Products are considered as the business resources for the organization. This includes managing the product with appropriate way to review any time as per the requirement.

Therefore it is important to have a computer based which has the ability to generate reports, maintain the balance of the stock, details about the purchase and sales in the organization. Before developing this application we came up with several Inventory Management System existing in the market, which helps to give the knowledge for the development of this project. These application software are only used by the large organization but so we came up with the application which can be used by the small company for the management of their stock in the production houses. After analysing the other inventory management system we decided to include some of common and key features that should be included in every inventory management system. So we decided to include those things that help the small organization to adapt with this application [7].

Stock management systems have become increasingly important in the modern business landscape, especially with the rise of e-commerce and online retail. Such systems help businesses keep track of their inventory levels, monitor product movement, and ensure that they always have enough stock on hand to meet customer demand. In this literature review, we will examine some of the key research studies and articles that have been published on stock management systems [3].

One study that explored the importance of stock management systems was conducted by Saravanan. The study examined the impact of a stock management system on the operational efficiency of a retail store. The results of the study showed that the use of a stock management system significantly improved the store's inventory accuracy, reduced stockouts, and increased the store's overall profitability [8].

Another study by Wang et al explored the use of a real-time stock management system in the manufacturing industry. The study found that the system helped reduce excess inventory levels, improve product quality, and increase production efficiency [9].

In a similar study, Parvez and Kabir examined the impact of a stock management system on supply chain performance. The results of the study showed that the use of a stock management system helped reduce inventory holding costs, minimize stockouts, and improve order fulfillment rates [9].

A more recent study by Jeevitha et al explored the use of a stock management system in the healthcare industry. The study found that the system helped reduce waste and spoilage, improve inventory accuracy, and increase the efficiency of medical supply management [7].

In addition to these research studies, there are many articles and reports available that provide insights into the best practices for stock management systems. For example, the Harvard Business Review published an article by Lee and Billington that provided a framework for improving supply chain management, including stock management. The article emphasized the importance of data analytics, collaboration between suppliers and customers, and the use of technology to improve inventory visibility [6].

Overall, the literature suggests that stock management systems can provide significant benefits to businesses across a wide range of industries. By improving inventory accuracy, reducing stockouts, and increasing operational efficiency, these systems can help businesses meet customer demand while minimizing costs and maximizing profitability.

In addition to the studies and articles mentioned above, there are also some key trends and developments in the field of stock management systems that are worth considering.

One trend is the increasing use of artificial intelligence (AI) and machine learning (ML) in stock management systems. AI and ML can be used to analyze large amounts of data and identify patterns and trends that can help businesses make more informed decisions about inventory management. For example, AI and ML can be used to predict demand for certain products based on historical sales data, seasonal trends, and other factors. This can help businesses optimize their inventory levels and reduce waste and excess inventory [1].

Another trend is the move towards cloud-based stock management systems. Cloud-based systems offer several benefits over traditional on-premise systems, including lower upfront costs, greater scalability, and easier access to real-time data. With a cloud-based system, businesses can access their inventory data from anywhere and at any time, which can be particularly useful for companies with multiple locations or remote workers. There is also a growing interest in sustainable stock management systems that prioritize environmental and social responsibility. This includes reducing waste and excess inventory, using environmentally-friendly packaging materials, and sourcing products from ethical and sustainable suppliers [1].

There is a recognition that stock management systems must be integrated with other systems and processes within a business in order to be truly effective. This includes integrating with accounting systems, order management systems, and logistics systems, among others. By

integrating these systems, businesses can achieve greater visibility and control over their entire supply chain, from raw materials to finished products.

In addition to the trends and developments in stock management systems, there are also some challenges that businesses may face when implementing these systems. One challenge is the need to balance inventory levels with customer demand. Businesses need to ensure that they have enough inventory on hand to meet customer demand, while also avoiding excess inventory that can lead to waste and higher holding costs.

Another challenge is the complexity of managing inventory across multiple channels, such as brick-and-mortar stores, e-commerce platforms, and third-party marketplaces. Businesses need to ensure that their inventory levels are consistent across all channels, while also taking into account differences in customer demand and fulfillment capabilities [9].

To overcome these challenges, businesses can implement best practices such as demand forecasting, safety stock planning, and continuous inventory tracking. They can also invest in inventory management software that offers real-time tracking and analytics capabilities. By taking a strategic approach to inventory management and leveraging the latest technologies and best practices, businesses can optimize their inventory levels and maximize their overall operational efficiency and profitability [8].

In summary, the field of stock management systems is constantly evolving, with new technologies, trends, and best practices emerging all the time. By staying up-to-date on these developments and implementing the most effective strategies and systems, businesses can optimize their inventory management and improve their overall operational efficiency and profitability.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

A system development methodology in software engineering is the main guidance for constructing, planning and also controlling the process of developing an information system. Common methodologies used were waterfall model, prototyping, spiral development, extreme programming and also some other various types of methodologies.

Considering the fact that this project involves design and implementation of a software system regardless that is web-based, it will be important to mention and consider some models used in software development and deployment, some general models of software development are namely waterfall model, prototyping, spiral development, extreme programming and also some other various types of methodologies.

I've chosen water fall model as I'm developing software for predefined problems only. The main aim of using this approach is we can focus on each part of the model during development and come back to it if need be. The project can easily de broken down into different parts based on this model. This is the model that will be used to develop the Stock Management System. However, feedback loops will be allowed during the whole software development process. I find this model suitable for me to follow.

It requires that software development follows the following stages:
- Requirements are to be proposed.
- System design should be made according to the requirements.
- Implementation of the features according to the design.
- Integration and testing of the system.
- Deployment of the system.
- Maintenance of the system.

Waterfall methodology is used while building this website. This project has specific documentation, fixed time, fixed requirements, and well-understanding technology so in order to build this system waterfall methodology can be properly utilized.
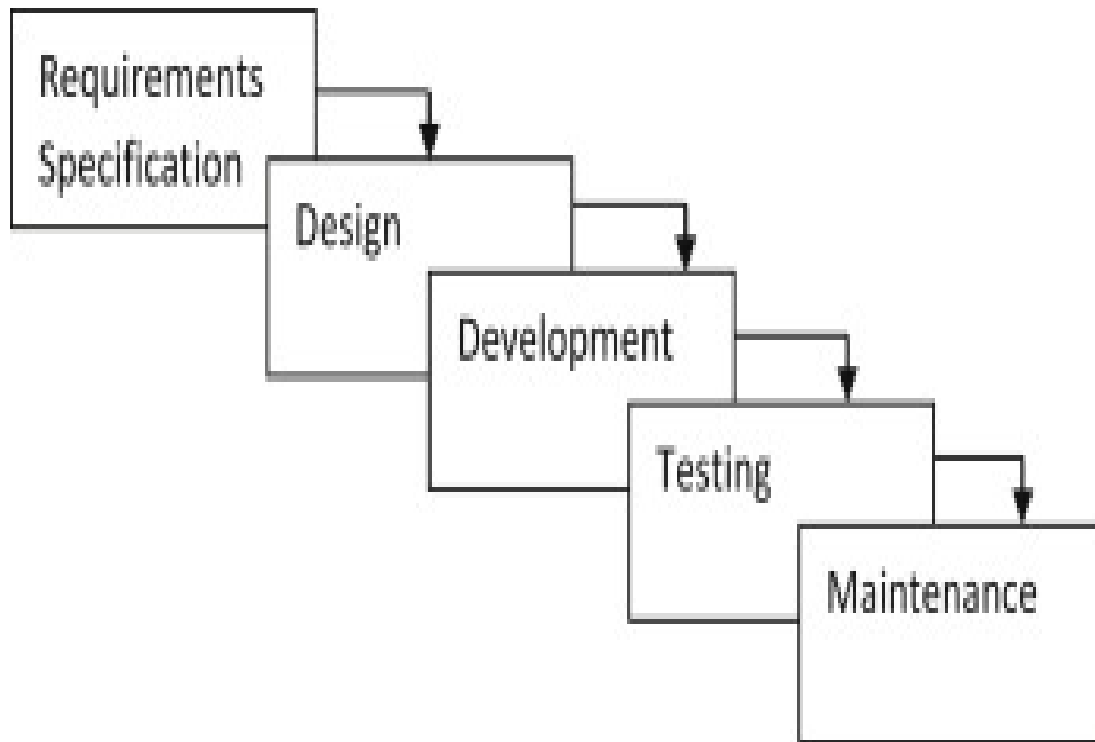
Fig 3.1: Waterfall model

1. Requirements Analysis: In the first phase of the waterfall model, the requirements for the stock management system are gathered from stakeholders such as business owners, managers, and users. This involves defining the features, functionalities, and scope of the system.

2. Design: In the second phase, the requirements gathered in the first phase are used to design the system. This includes creating a detailed system design document that outlines the system architecture, database schema, user interface design, and system workflows.

3. Deployment: In the fifth phase, the stock management system is deployed to the production environment. This involves installing the system on the production servers, configuring the system for the production environment, and performing any necessary data migration..

4. Testing: In the fourth phase, the stock management system is thoroughly tested to ensure that it meets the requirements outlined in the first phase. This includes unit testing, integration testing, system testing, and user acceptance testing.

5. Maintenance: In the final phase, the stock management system is maintained and updated as necessary to ensure that it continues to meet the needs of the business. This includes bug fixes, system updates, and new feature development

3.1.1 Requirements Analysis

Requirement analysis is done so that the project gets the necessary features and will be easy for analysing system. Requirement analysis is an important phase in developing a stock management system, as it helps to ensure that the system meets the needs and expectations of the business It is a key instrument used to determine the needs and expectations of a new product. In this project requirements are categorized into two parts i.e. functional requirements and non-functional requirements.

For any system, there are functional and non-functional requirements to be considered while determining the requirements of the System.

I. Functional requirement: In this project the functional requirements are categorized into two different models i.e., Staff Module and Admin Module. Under Staff Model user can easily use the features like login, viewing products managing products, receiving products. Whereas Admin Module consists of using the system as an administrative which consists of features like managing products, managing users, managing suppler and viewing orders and printing orders and exporting data into excel or PDF. The Functional requirements in the project are mentioned below.

User Module:

- User shall login the system.
- User shall Manage Purchase, Sales, Back Order and Return Orders.
- User shall Print Purchase, Sales, Return Orders and Exporting data into excel or PDF.
- User shall View Analytics.
- User shall view available Stocks.

Admin Module:

- Admin shall login the system.
- Admin shall Manage Purchase, Sales, Back Order and Return Orders.
- Admin shall see the registered users and create the user also delete the users.
- Admin shall view available Stocks.
- Admin shall Print Purchase, Sales, Return Orders and Exporting data into excel or PDF
- Admin shall manage supplier and Items.
- Admin shall View Analytics

3.1.1 Use Case Diagram

A use case diagram is used in this project which will help to understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors a graphic depiction of interactions among different elements in a system. A use case diagram for a stock management system would include actors, use cases, and the system boundary. The actors in this system would be the Admin and the User. The use cases would include, Login, Create Edit View Purchase Order, Approve Purchase Order, View Back Order, and Create Edit View Sales order, Export the Report in PDF, Excel Format, and use Print also. Remaining Stock with the proper validation in the data. View of Analytics which will cover the chart like bar chart, Radar chart and line chart of various Properties and the action. Admin can only be able to manage the Item list, Supplier List, and User list and will also be able to manage the user and create the user from Admin panel.

The system boundary would be represented by a rectangle box, and they Include and Extend relationships would indicate necessary and optional features, respectively.

By using a use case diagram, the stock management system can be visualized and designed to meet the needs of both the Admin and User actors, ensuring that the system is efficient and user-friendly.

STOCK MANAGEMENT SYSTEM

Login

Create Edit and View Purchase Order

Approve /Receive PO

View Back Order

Create Edit and View Return Lists

Create Edit and View Sales

Stock Lists

Generate Reports in Pdf ,Excel Format

Print Reports

View Of Analytics

Manage User

Manage Supplier

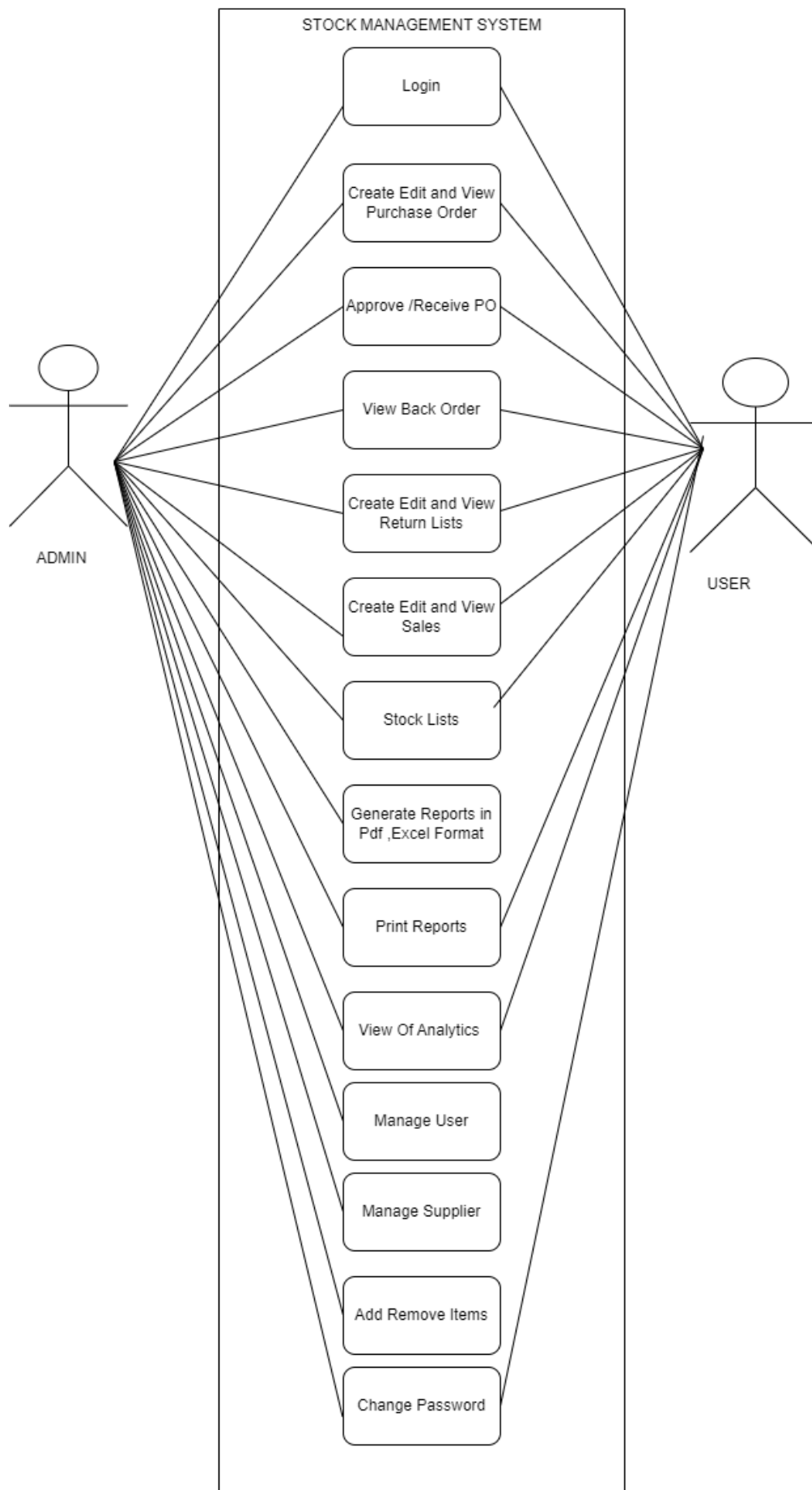Add Remove Items

Change Password

ADMIN

USER

Fig 3.1.1: Use case diagram for Stock Management System

II. Non-functional requirements: They are an important aspect of developing a stock management system as they focus on the systems overall performance and usability. Here are some key non-functional requirements that should be considered during the development of a stock management system

- Performance: The system should be designed to handle large amounts of data and transactions with minimal response time. It should be able to process multiple user requests simultaneously without affecting system performance.

- Reliability: The system should be reliable and available at all times. It should be able to recover from any failures or crashes without losing data or disrupting business operations.

- Security: The system should be secure and protect sensitive inventory and customer data. It should be designed to prevent unauthorized access, data breaches, and other security threats.

- Usability: The system should be user-friendly and easy to use, with a well-designed user interface and clear navigation. It should be able to support multiple languages and be accessible to users with disabilities.

- Scalability: The system should be able to scale to handle increased data volumes, users, and transactions as the business grows. It should be designed to handle future expansion without significant system redesign.

- Maintainability: The system should be easy to maintain and update, with minimal downtime and disruptions to business operations. It should be designed with modular architecture and well-documented code to facilitate future updates and enhancements.

3.1.2 Feasibility study

A feasibility study is an evaluation and analysis of a project or system that somebody has proposed. Following feasibilities were studied before building the system to see if the system could be built with exact requirements in required time.

I. Technical Feasibility

In order to design this system, it uses existing technologies, software and hardware so there is no technological hurdle to build this system.

- The UI of our project is very simple
- User will require internet browser and internet to use it
- For Email Services a Auth Token Needs to be Configured.

Tools and Technology Used:

The Following software is used for the development of the System.

- IDE: VS Code

Visual Studio Code is the primary development environment. It offers code highlighting, debugging, and extensions, making coding more efficient.

- Database: MySQL

MySQL is used as the database for storing information like stock levels, orders, and user data. It's a reliable and widely used solution for managing large amounts of data.

- Hosting: Apache (XAMPP)

XAMPP provides a local server environment using Apache, allowing the web application to be hosted and tested during development.

- Languages: PHP, JavaScript, HTML, CSS

PHP handles the server-side logic of the system, processing data and connecting to the database. JavaScript adds interactivity to the user interface. HTML is used to structure the web pages, while CSS styles them to look modern and user-friendly.

- Frameworks: Bootstrap

Bootstrap is used to create a responsive and mobile-friendly design. It simplifies the styling process and ensures that the application looks good on any device.

- Libraries: PHP Mailer, Highcharts, Charts.js

PHP Mailer is used for sending emails from the system, while Highcharts and Charts.js generate interactive reports and charts to help users understand sales, stock levels, and other data.

- Version Controlling: GIT, GitHub

Git and GitHub are essential for version control, allowing the development team to track changes in the code and collaborate efficiently.

- Browsers: Chrome, Edge

The system is tested on Google Chrome and Microsoft Edge to ensure it works well across different web browsers.

- Documentation: Draw.io, MS Word

Draw.io is used for creating system flowcharts and diagrams, while MS Word is used for writing documentation and reports.

.II. Economic Feasibility

Before the development of a system, the proposed system should be studied whether or not it is within the budget estimated by the organization. The project that we are developing is within the cost estimation of the organization. The project cost is less and no more burdens are needed. The system development does not have any requirement of expensive hardware and software. The platform are open sources and the resources required for the project are also open source. Hence the project is said to be economically feasibility

III. Operational Feasibility

These include the reliability, maintainability, usability, supportability. The proposed system is operationally feasible as it is reliable for all type of user i.e., whether or not the user has the knowledge of computer or not. The proposed system is supported for a small to large-scale organization. It is simple and easy to use due to simple user interface and its operational feasible.

IV. Schedule Feasibility

The system that we developed is scheduling feasible as it does not require more time for the development phase. The data collection takes more time to collect the data about various products and their quality. After data is collected, the other development phase can be within a month. Gantt charts: Gantt chart is a bar chart that provides a visual view of tasks scheduled over time. A Gantt chart is used for planning projects of all sizes, and it is a useful way of showing what work is scheduled to be done on a specific day- It can also help you view the start and end dates of a project in one simple chart. In our project, we used Ms. Excel for developing the Gantt chart which is shown below in the figure.

Task	Start date	End date
Requirement	Jun-2	Jun-7
Analysis	Jun-8	Jun-17
Design	Jun-18	Jun-29
Coding	Jun-30	Sep-16
Testing	Sep-17	Sep-24
Report	Sep-25	Oct-3

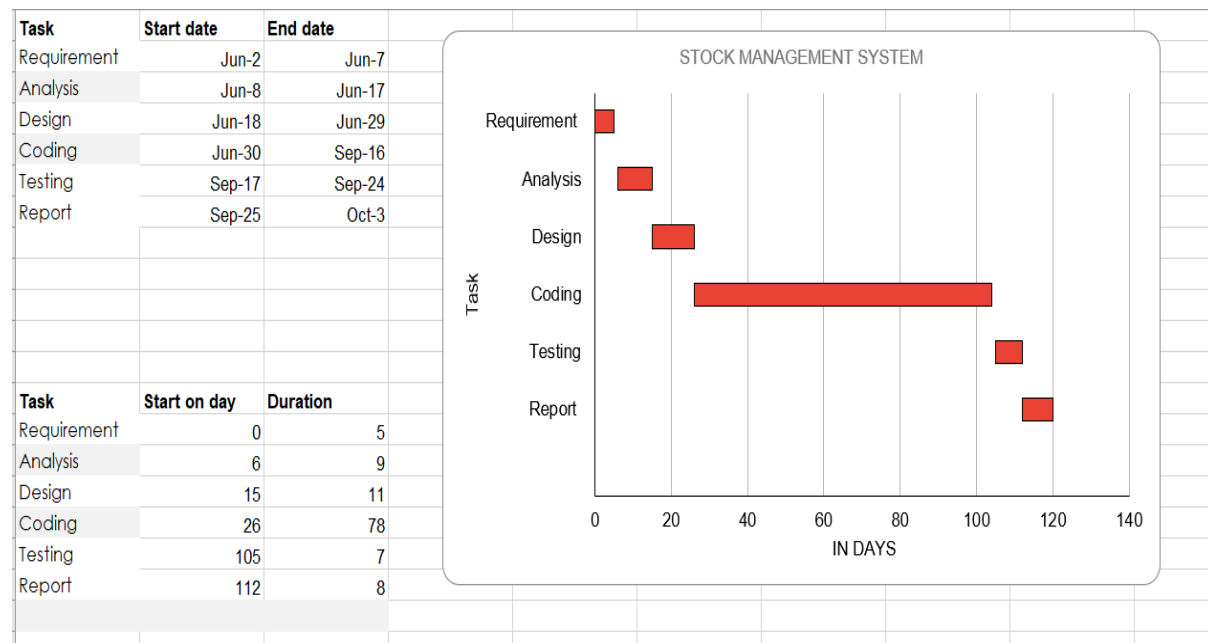| Task	Start on day	Duration
Requirement	0	5
Analysis	6	9
Design	15	11
Coding	26	78
Testing	105	7
Report	112	8

Fig 3.1.2: Gantt chart for Stock Management System.

3.1.3 Object Modelling using Class Diagram and Object Diagram

This class diagram outlines the fundamental components and their relationships within an Stock Management System (SMS). It includes essential classes like User, Admin, Purchase, Sales, Supplier, Return List, Stock and Receiving. This diagram represents the structure of a Stock Management System (SMS), showing how different entities interact with each other to manage the flow of products from suppliers to customers.

At the top of the system, we have Admins and Users. The Admin can manage users, items, purchases, sales, stocks, send bulk SMS, and view analytics. Admins also handle supplier relationships. Regular Users can log in, create purchase orders, manage sales, and view stock or analytics, but with fewer permissions compared to Admins. Purchase Orders are created when items need to be bought from suppliers. These orders are connected to the Suppliers who provide the goods. Once a purchase order is approved, the system moves to the Receiving stage, where the actual items are delivered and checked. When the goods are received, the Stock is updated to reflect the new inventory. Sales Orders represent the process of selling items to customers. When a sale is made, the system decreases the stock, ensuring the inventory is always accurate. If there are any issues with the products, like defects or overstock, they are handled through the Return List, where items are returned to suppliers, and the stock is adjusted accordingly. This system also ensures that both Purchase Orders and Sales Orders can be viewed, edited, or deleted as needed. The Stock entity keeps track of how many products are available at all times, updating whenever items are bought, sold, or returned.

The diagram shows a clear flow from Purchase Orders to Receiving to Stock, and finally to Sales Orders, with Admins and Users managing these processes to ensure the stock levels are always accurate and up to date.
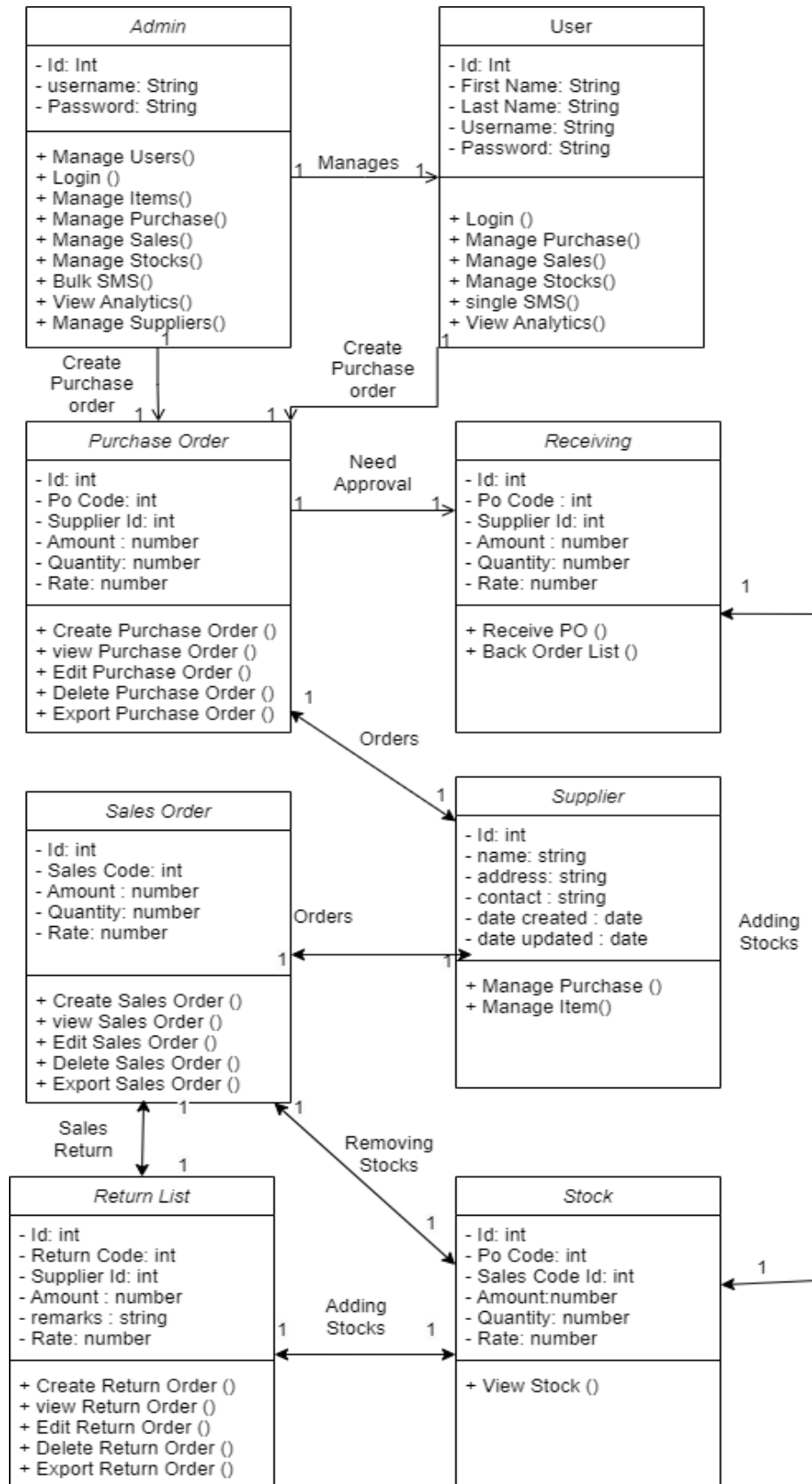
Admin

- Id: Int
- username: String
- Password: String

+ Manage Users()
+ Login ()
+ Manage Items()
+ Manage Purchase()
+ Manage Sales()
+ Manage Stocks()
+ Bulk SMS()
+ View Analytics()
+ Manage Suppliers()

User

- Id: Int
- First Name: String
- Last Name: String
- Username: String
- Password: String

+ Login ()
+ Manage Purchase()
+ Manage Sales()
+ Manage Stocks()
+ single SMS()
+ View Analytics()

Manages

Create Purchase order

Create Purchase order

Purchase Order

- Id: int
- Po Code: int
- Supplier Id: int
- Amount : number
- Quantity: number
- Rate: number

+ Create Purchase Order ()
+ view Purchase Order ()
+ Edit Purchase Order ()
+ Delete Purchase Order ()
+ Export Purchase Order ()

Need Approval

Receiving

- Id: int
- Po Code : int
- Supplier Id: int
- Amount : number
- Quantity: number
- Rate: number

+ Receive PO ()
+ Back Order List ()

Orders

Sales Order

- Id: int
- Sales Code: int
- Amount : number
- Quantity: number
- Rate: number

+ Create Sales Order ()
+ view Sales Order ()
+ Edit Sales Order ()
+ Delete Sales Order ()
+ Export Sales Order ()

Orders

Supplier

- Id: int
- name: string
- address: string
- contact : string
- date created : date
- date updated : date

+ Manage Purchase ()
+ Manage Item()

Adding Stocks

Sales Return

Removing Stocks

Return List

- Id: int
- Return Code: int
- Supplier Id: int
- Amount : number
- remarks : string
- Rate: number

+ Create Return Order ()
+ view Return Order ()
+ Edit Return Order ()
+ Delete Return Order ()
+ Export Return Order ()

Adding Stocks

Stock

- Id: int
- Po Code: int
- Sales Code Id: int
- Amount:number
- Quantity: number
- Rate: number

+ View Stock ()

Fig 3.1.3: Class Diagram for Stock Management System.

22

3.1.4 Dynamic Modelling using State and Sequence Diagrams

The diagram represents the dynamic modeling of an Admin's and User interaction within a Stock Management System. It shows the sequence of actions performed by an admin user, starting from logging in and proceeding through item management, stock management, and analytics.

1. Login Process: The Admin/users begins by logging into the system using the login page. This step is essential for authentication and gaining access to the system's functionalities.

2. Item Management: After logging in, the admin navigates to the Item Management module, where they can manage both Items and Suppliers. This interaction involves adding or updating item information, as well as managing the suppliers who provide these items.

3. Stock Management: Next, the admin moves to the Stock Management section. Here, they manage stock data such as purchases, sales, back orders, return lists, and current stock levels. The admin ensures that stock levels are accurate and updated according to business needs. Additionally, stock-related data is sent for further processing.

4. Analytics: Lastly, the admin accesses the Analytics module, where they can view different types of data visualizations. These include:

   - Heat Maps: Visual representations of stock movement or sales patterns.
   - Top Seller: Identifying the best-selling products.
   - Stock Analytics: Detailed analysis of stock levels and turnover.
   - Quantity Flow: Information on how stock quantities move through the system, from purchase to sale.

Dynamic modeling helps in understanding the flow of activities and interactions within the system, facilitating the identification of potential bottlenecks or issues in the process.

For Login

This sequence diagram illustrates the process of a user or admin logging into the system. First, the user enters their username and password on the login page. These credentials are then sent to the PHP server, which communicates with the database to verify if the credentials are valid. If the database confirms the credentials match an existing user, the PHP server informs the login page, allowing the user to navigate to the dashboard. However, if the credentials are incorrect, the system returns an error message indicating an invalid username or password, preventing access to the dashboard and asking the user to retry the login. This flow ensures secure user authentication before granting access to the system's features
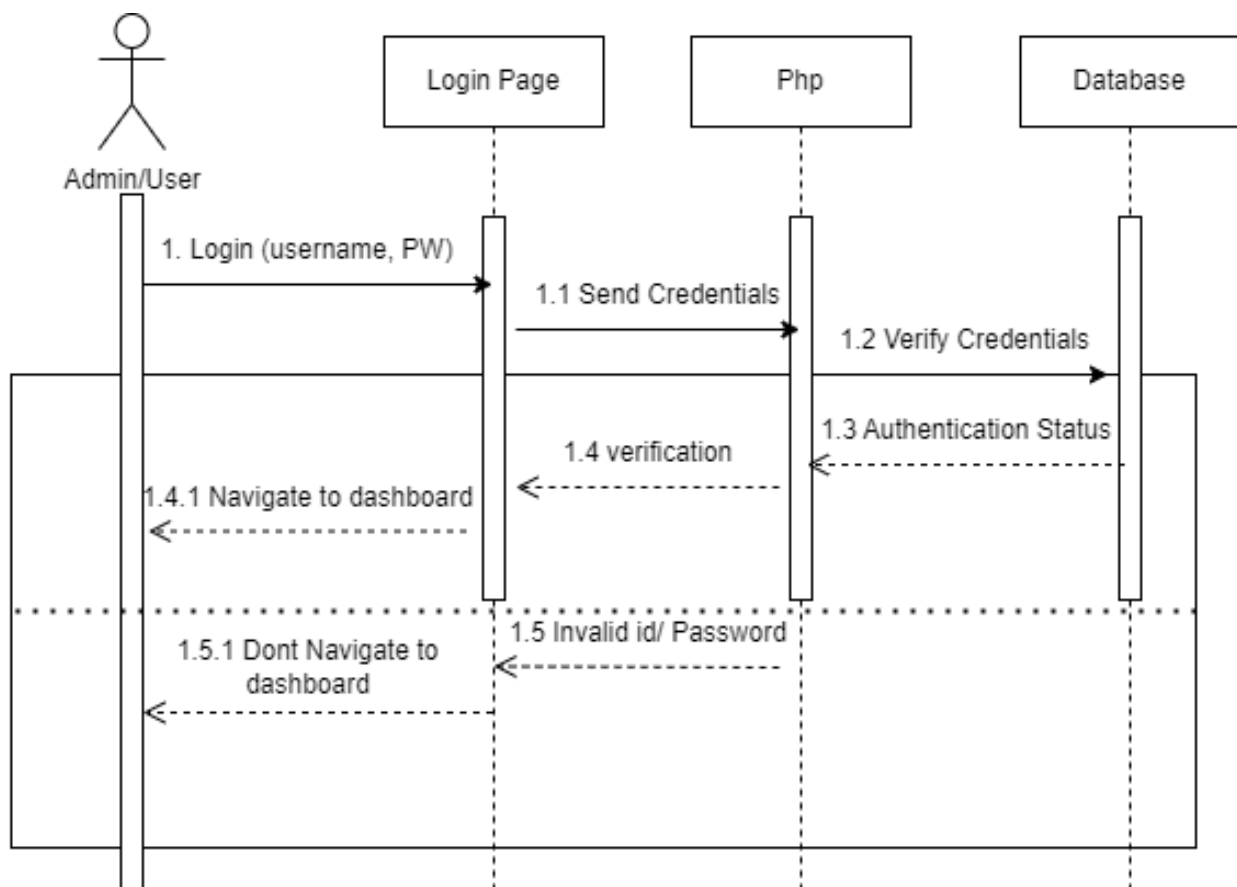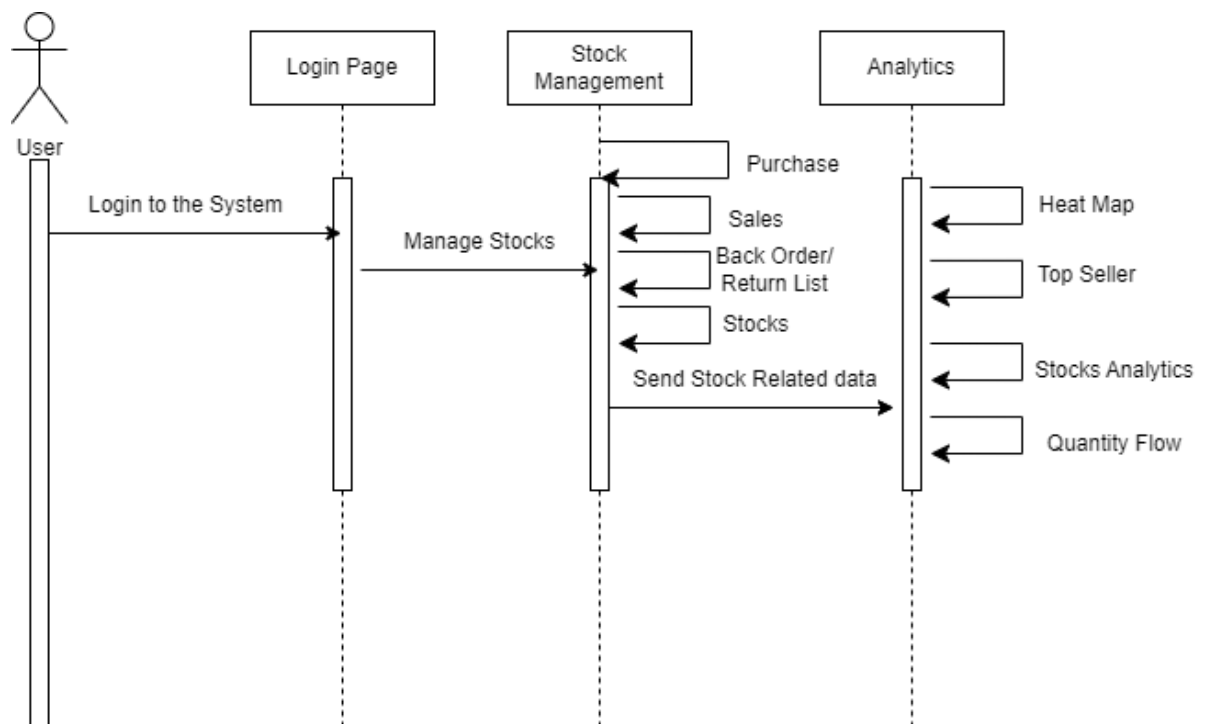

Fig 3.1.4: Sequence Diagram for Login

For Users

This sequence diagram illustrates the overall process of users in the system Login Process. The users begins by logging into the system using the login page. This step is essential for authentication and gaining access to the system's functionalities. Next, the users moves to the Stock Management section. Here, they manage stock data such as purchases, sales, back orders, return lists, and current stock levels. The users ensures that stock levels are accurate and updated according to business needs. Additionally, stock-related data is sent for further processing. Lastly, the users accesses the Analytics module, where they can view different types of data visualizations. These include Heat Maps, Top Seller Stock Analytics and Information on how stock quantities move through the system, from purchase to sale.
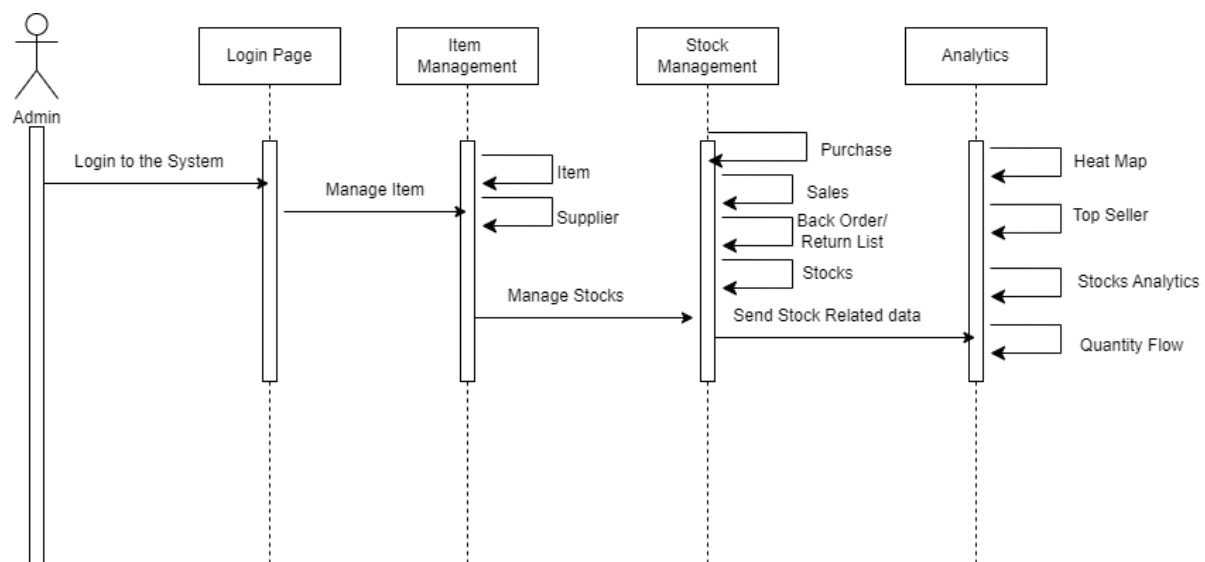

Fig 3.1.4.1: Sequence Diagram for Users

For Admin

This sequence diagram illustrates the overall process of users in the Login Process the users begins by logging into the system using the login page. This step is essential for authentication and gaining access to the system's functionalities. After logging in, the admin navigates to the Item Management module, where they can manage both Items and Suppliers. This interaction involves adding or updating item information, as well as managing the suppliers who provide these items. Next, the admin moves to the Stock Management section. Here, they manage stock data such as purchases, sales, back orders, return lists, and current stock levels. The admin ensures that stock levels are accurate and updated according to business needs. Additionally, stock-related data is sent for further processing. These include Heat Maps, Top Seller Stock Analytics and Information on how stock quantities move through the system, from purchase to sale

Fig 3.1.4.2: Sequence Diagram for Admin

3.1.5 Process Modelling using Activity Diagram

In a stock management system, the process starts when a user logs in. The system checks if the username and password are correct. If they're wrong, the system asks the user to try again. If the login is successful, the user is either given Admin or User access, depending on their role.

If the user is an Admin, they have several options. They can view the list of users, check the list of suppliers, and manage the list of items (for example, by adding, updating, or removing items). Admins can also manage sales and view analytics reports, which give them insights into how the business is performing.

On the other hand, a User has different responsibilities. They can view the current stock levels, make purchases, and handle item returns. Each time they make a purchase or return an item, the system updates the stock levels. Users can also request analytics reports based on stock and purchase history to help them understand stock usage. Throughout the process, the system interacts with a database to get and store information, such as stock levels, sales data, and user details. The whole process ends when the user finishes their tasks and logs out of the system.
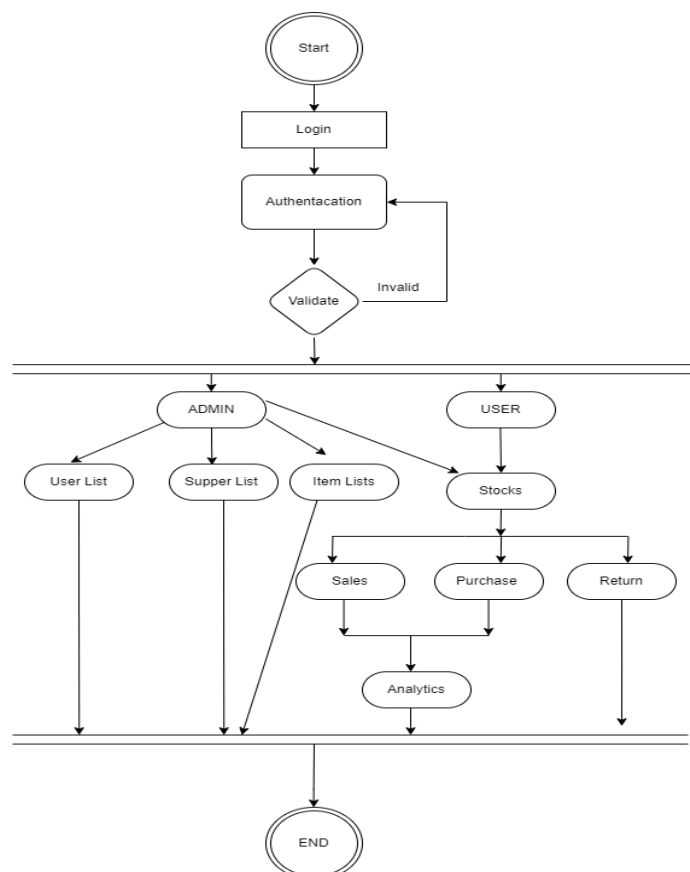

Fig 3.1.5: Activity Diagram for Stock Management System

27

3.2 System Design

Considering the fact that this project involves design and implementation of a software system regardless that is web-based, it will be important to mention and consider some models used in software development and deployment. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.

3.2.1 Refinement of classes and object

The stock management system diagram shows how different parts of the system work together. There are two main users: Admin and User. Admins have full control, managing things like users, items, suppliers, and sending bulk SMS. Users can manage purchases, sales, and stocks, but with fewer permissions. Suppliers provide items to the system. When a company needs more stock, they create a Purchase Order, which includes details like the supplier, item, quantity, and cost. After approval, the items are received and added to the Stock. When a customer makes a purchase, a Sales Order is created, and the stock is reduced accordingly. If items are returned, a Return List is used to add the stock back. These orders are connected to the Suppliers who provide the goods. Once a purchase order is approved, the system moves to the Receiving stage, where the actual items are delivered and checked. When the goods are received, the Stock is updated to reflect the new inventory. Sales Orders represent the process of selling items to customers. When a sale is made, the system decreases the stock, ensuring the inventory is always accurate. If there are any issues with the products, like defects or overstock, they are handled through the Return List, where items are returned to suppliers, and the stock is adjusted accordingly. This system also ensures that both Purchase Orders and Sales Orders can be viewed, edited, or deleted as needed. The Stock entity keeps track of how many products are available at all times, updating whenever items are bought, sold, or returned.

The diagram shows a clear flow from Purchase Orders to Receiving to Stock, and finally to Sales Orders, with Admins and Users managing these processes to ensure the stock levels are always accurate and up to date.
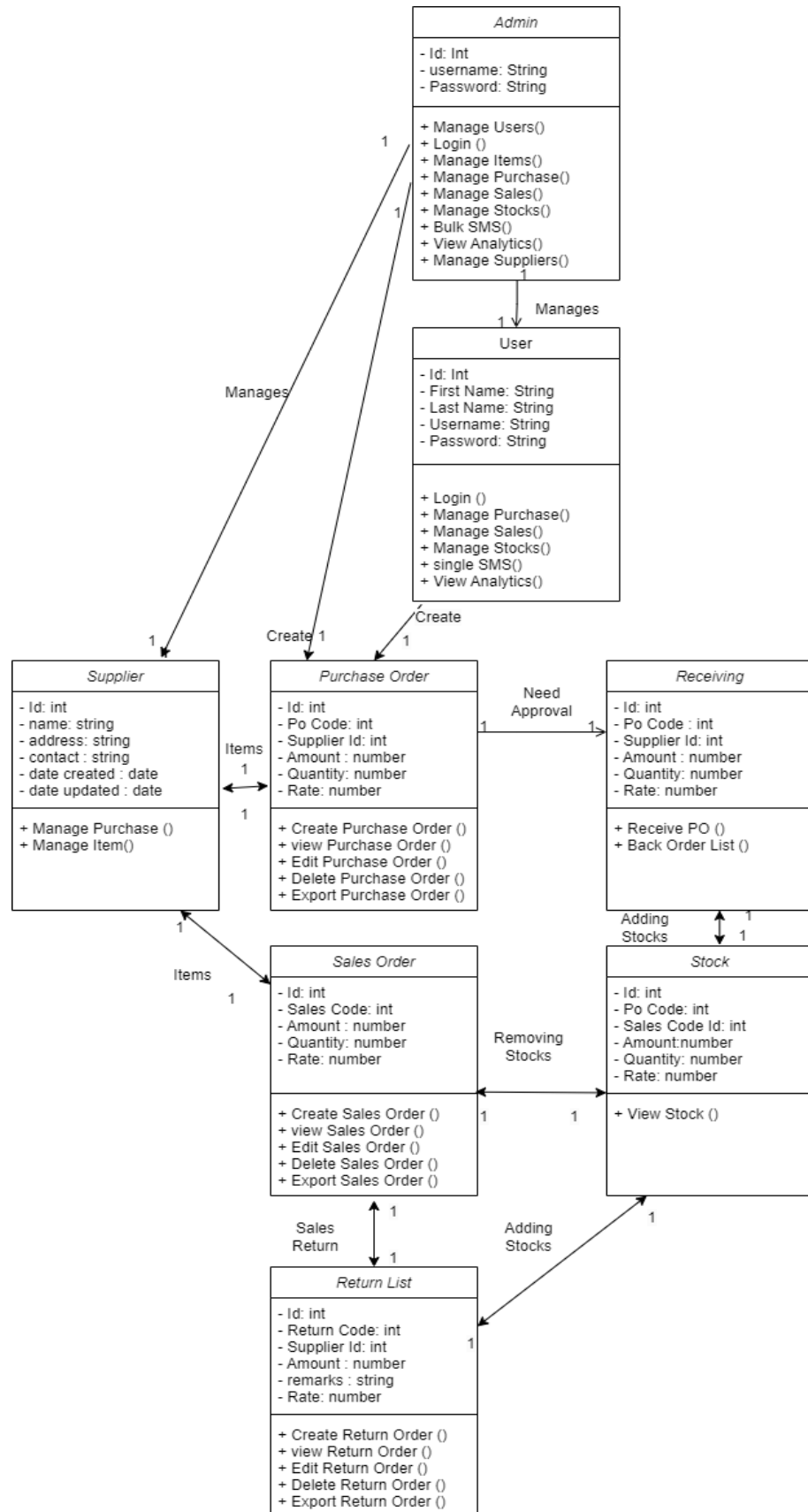
Fig 3.2.1: Refinement of Classes and Object Diagram for Stock Management System

3.2.2 Component Diagram

The component diagram for the stock management system shows how the system is organized to handle various tasks related to inventory, sales, and supply management. At the center is the User, who interacts with the system by generating different reports, such as Inventory Reports, Sales Reports, and Supply Reports. Each of these reports pulls data from their respective databases: the Stocks Database, Sales Database, and Supply Database. The system is designed to fetch information from these databases to generate detailed and accurate reports for the user. For example, the Stocks Reports component retrieves data from the Stocks Database to show current stock levels, while the Sales Reports component accesses the Sales Database to summarize sales transactions. This structure ensures that users have access to up-to-date and organized information about stock, sales, and supply in the system.
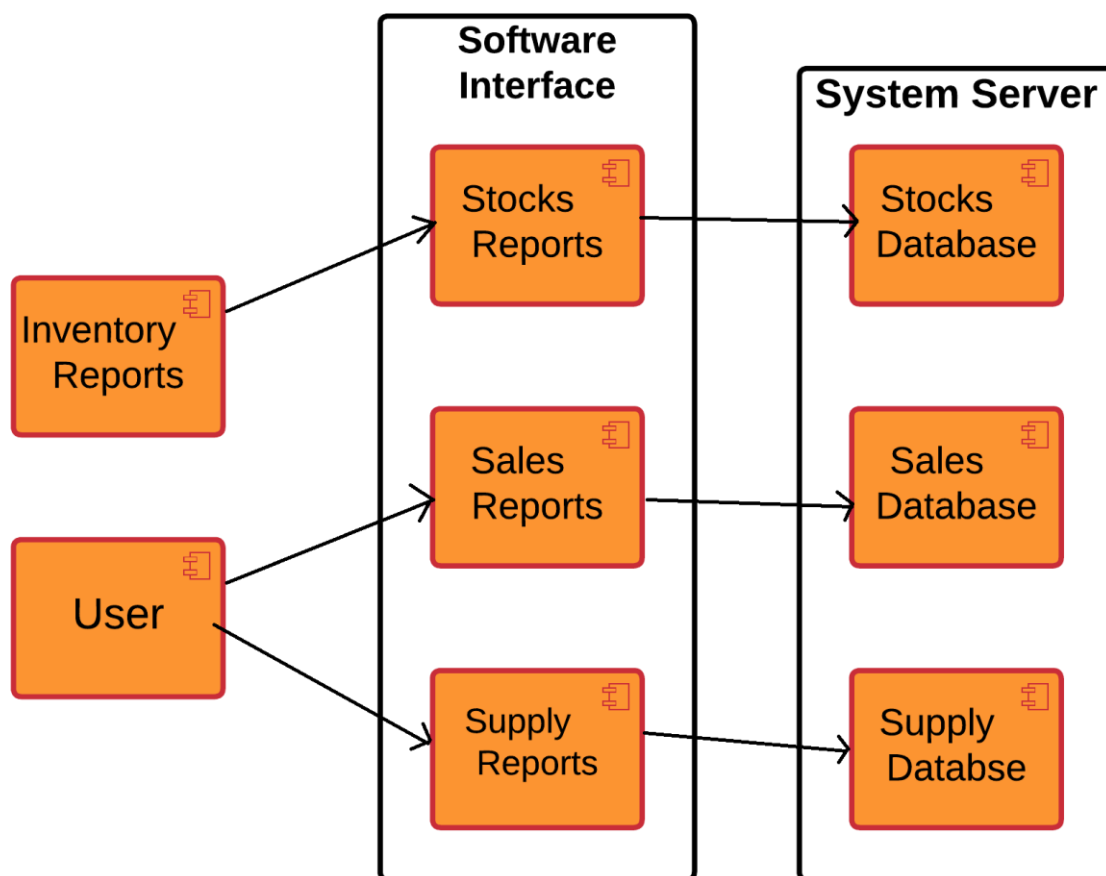

Fig 3.2.2: Component Diagram for Stock Management System

3.2.3 Deployment Diagram

This deployment diagram illustrates the structure of a stock management system, highlighting how different components interact. The system consists of three main layers: the Client, Application Server, and Database Server. The Client is represented by a web browser, where users interact with the system to perform tasks such as checking inventory, placing orders, or viewing stock levels. The Application Server, powered by a PHP Server, processes these user requests. It acts as the business logic layer, handling operations like order processing and communicating with the database. The Database Server consists of a MySQL Database, which stores critical information related to stocks, purchases, sales, and users. The PHP server sends transaction requests to the database to either retrieve or update stock-related data. This layered architecture ensures that the client, application, and database are well-separated, enabling better performance, scalability, and maintainability for the stock management system.
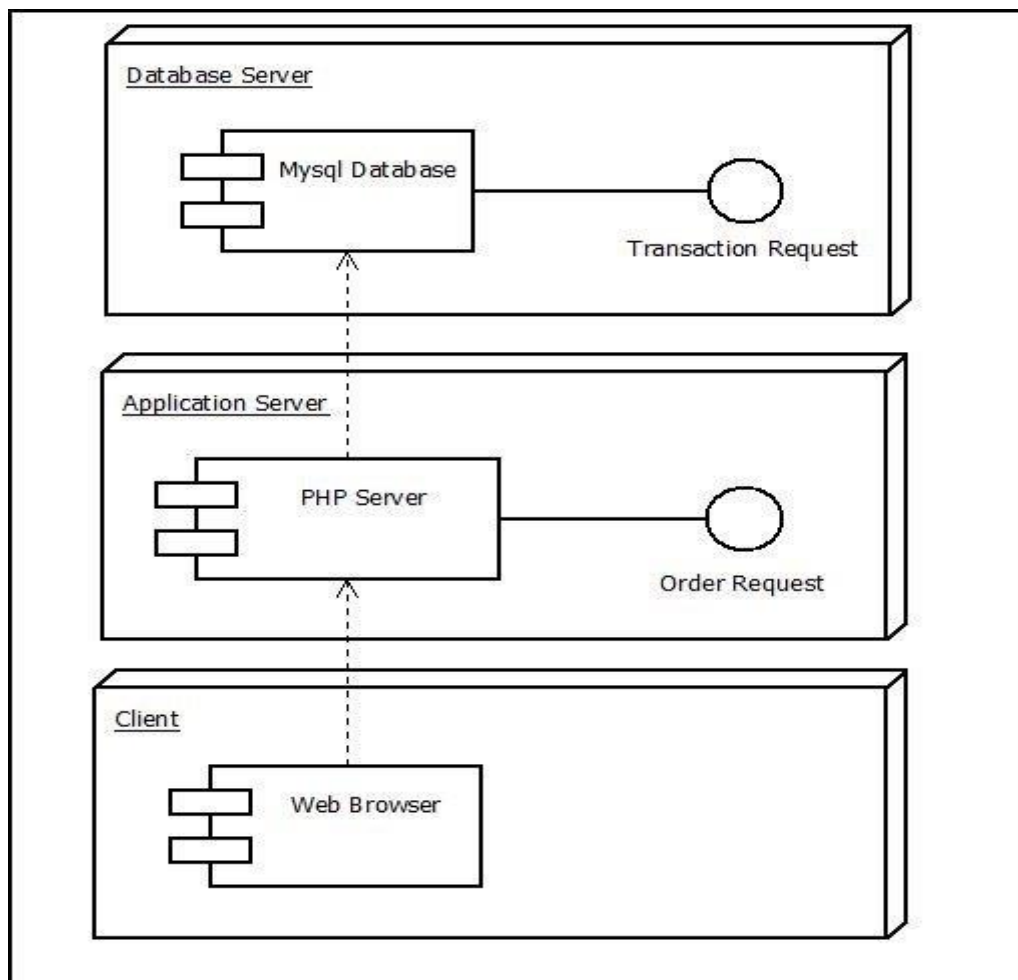

Fig 3.2.3: Deployment Diagram for Stock Management System

3.3 Algorithm

3.1 Rule Based Algorithm for Top Seller Page

The rule-based algorithm in this code focuses on analyzing and providing sales suggestions based on the sales data for clients. The algorithm is designed to compare sales performance for clients in the current year with their historical all-time performance and generate specific recommendations.

1. Rule 1: It calculates the difference between the highest and lowest sales for this year. The rule suggests that the client with the lowest sales can improve by a certain amount to catch up to the client with the highest sales this year.

2. Rule 2: This rule is similar to Rule 1 but focuses on all-time sales. It compares the client with the lowest total sales to the one with the highest total sales and recommends how much the lower-performing client needs to increase their sales to catch up.

3. Rule 3: The algorithm also identifies the client with the highest growth potential based on the percentage growth of sales from the all-time total to the current year's sales. This rule highlights which client has shown the most promising increase in sales performance this year.

The growth potential is calculated for all clients by finding the percentage of growth between their sales this year and their total sales. The algorithm ranks clients based on this growth potential, providing detailed insights into each client's growth percentage, their sales this year, and their total historical sales.

Algorithm

1. Initialize variables:

   - $max_sales_this_year to 0 (maximum sales this year)

   - $max_sales_all_time to 0 (maximum sales of all time)

   - $min_sales_this_year to PHP_INT_MAX (minimum sales this year)

   - $min_sales_all_time to PHP_INT_MAX (minimum sales of all time)

   - $total_sales to 0 (total sales)

2. Iterate over the $sales_data_all_time array using a foreach loop, with each iteration represented by the variable $data:

- Increment $total_sales by adding the current $data['total_sales'] to it.

- Initialize $total_sales_this_year to 0 (total sales for the current year).

- Initialize $total_sales_all_time to $data['total_sales'] (total sales of all time for the current client).

- Iterate over the $sales_data_this_year array using a nested foreach loop, with each iteration represented by the variable $d:

  - Check if the current client ($data['client']) matches the client in $d['client'].

  - If they match, assign $d['total_sales'] to $total_sales_this_year and break out of the loop.

- Compare and update the following variables if necessary:

  - $max_sales_this_year and $max_sales_this_year_client if $total_sales_this_year is greater than $max_sales_this_year.

  - $max_sales_all_time and $max_sales_all_time_client if $total_sales_all_time is greater than $max_sales_all_time.

  - $min_sales_this_year and $min_sales_this_year_client if $total_sales_this_year is less than $min_sales_this_year.

  - $min_sales_all_time and $min_sales_all_time_client if $total_sales_all_time is less than $min_sales_all_time.

3. Initialize variables for finding the person closest to the median sales:

- $middle_sales_person to an empty string.

- $middle_sales_diff to PHP_INT_MAX (difference from the median sales).

- Iterate over the $sales_data_all_time array using a new foreach loop, with each iteration represented by the variable $data:

  - Initialize $total_sales_this_year and $total_sales_all_time as done in step 2.

- Calculate the absolute difference between $total_sales_all_time and half of $total_sales and assign it to $diff.

- Update $middle_sales_diff, $middle_sales_person, and $middle_sales_total if $diff is less than $middle_sales_diff.

4. Output the following statements within <p> elements:

- The client with the highest sales this year: $max_sales_this_year_client with total sales of $max_sales_this_year.

- The client with the highest sales of all time: $max_sales_all_time_client with total sales of $max_sales_all_time.

- The client with the lowest sales this year: $min_sales_this_year_client with total sales of $min_sales_this_year.

- The client with the lowest sales of all time: $min_sales_all_time_client with total sales of $min_sales_all_time.

- The amount $min_sales_this_year_client needs to increase sales by to catch up to $max_sales_this_year_client this year.

- The amount $min_sales_all_time_client needs to increase sales by to catch up to $max_sales_all_time_client in total sales.

- The person closest to the median sales: $middle_sales_person with total sales of $middle_sales_total.

5. Output a <p> element with the text "Sales Summary:".

6. Output an unordered list (<ul>) and iterate over the $sales_data_all_time array using a foreach loop:

- Output each client's name ($data['client']) and their corresponding total sales ($data['total_sales']) within a list item (<li>).

3.3.2 ABC Algorithm for Stock Analytics Page

ABC analysis is an inventory management technique that classifies items based on their importance to the business, particularly in terms of their contribution to total inventory value. This method divides inventory into three categories: A, B, and C, according to their value and how frequently they are sold or used.

First, the code collects and merges stock data based on the date, summing up the quantity and calculating the total value (quantity multiplied by price) for each date.Once the stock data is merged, the total value of all stock items is calculated.

Each stock item is then assigned to Category A, B, or C based on its contribution to the total value:

- Items contributing more than 70% are placed in Category A.
- Items contributing between 20% and 70% are placed in Category B.
- Items contributing less than 20% are placed in Category C.

Finally, the code displays a list of items categorized under A, B, or C, along with their total merged quantity and value, giving a clear breakdown of how inventory is distributed in terms of value.

Algorithm

1. Get Stock Data:

   - Retrieve stock data (date, quantity, price) from the database.

2. Store Stock Data:

   - Create an empty list to hold the stock data.

   - For each record from the database:

     o Convert quantity and price to numbers.

     o Add the record to the list.

3. Prepare Data for Processing:

   - Convert the list of stock data into a format that JavaScript can use.

4. Combine Entries by Date:

- Create a map to group stock data by date.

- For each entry:

  o Extract the date (ignore time).

  o If the date is already in the map:

    ▪ Add the quantity to the existing entry.

    ▪ Calculate the total value (quantity × price) and add it to the existing total.

  o If the date is not in the map:

    ▪ Create a new entry with the date, quantity, and total value.

5. Calculate Total Inventory Value:

- Sum up all the total values to get the overall inventory value.

6. Assign ABC Categories:

- For each unique date entry:

  o Calculate the percentage of total value for that entry.

  o Assign categories:

    ▪ A: More than 70% of total value.

    ▪ B: More than 20% but less than or equal to 70%.

    ▪ C: 20% or less.

7. Display Suggestions:

- Clear any previous suggestions on the webpage.

- For each entry in the processed data:

  o Create a list item that includes:

    ▪ The date.

    ▪ The assigned category.

    ▪ The total quantity.

    ▪ The total value.

3.3.5 Safety Stock Check Algorithm for Quantity Flow Page

The stock safety check algorithm works by analyzing historical stock and purchase data to determine the optimal level of safety stock, which helps ensure that a business maintains sufficient inventory to meet demand despite fluctuations. First, stock and purchase quantities are retrieved from the database. Then, the average demand is calculated by taking the total stock quantities over time and dividing by the number of data points. To account for demand variability, the algorithm computes the standard deviation, which measures how much stock levels fluctuate from the average.

Using a desired service level (in this case, 95%, which corresponds to a Z-score of 1.64), the safety stock is determined by multiplying the standard deviation by the Z-score. This provides an extra buffer of stock to cover unpredictable demand surges.

Algorithm

1. Fetch Data from Stock and Purchase Databases:
   - Stock data is retrieved using SQL queries from stock_list (for date_created and quantity).
   - Purchase data is fetched from po_items (for item_id and total_quantity).
2. Calculate Average Demand:
   - The average demand is determined by summing all stock quantities and dividing by the number of stock entries.
3. Calculate Standard Deviation:
   - Standard deviation measures demand variability. It's calculated by finding the squared differences between each quantity and the average demand, summing them, and then taking the square root of the average of those differences.
4. Safety Stock Calculation:
   - Based on the service level (assumed to be 95%, which corresponds to a Z-score of 1.64 in this example), safety stock is computed as: Safety Stock=Z×Standard Deviation
   - This gives an estimate of the additional stock required to protect against demand variability.
5. Suggestions for Stock Maintenance:
   - If the current stock falls below the safety stock level, it suggests placing a reorder.

37

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

Implementation basically means the phase where the system is actually being built. Firstly, all the information that we gathered is studied and analysed and implemented a system in operation for users. It is one of the most important phases of any project. Implementation usually consists of coding; testing, installation, documentation, training and support. Different tools and technologies that have been used to develop the system which are already discuss in the previous chapter. It is basically converting system design specification into working software.

4.1.1 Tools Used

The various system tools that have been used in developing both the front-end and backend of the project are being discussed in this chapter.

Front-end

Bootstrap, HTML, CSS, and JavaScript are used for developing the front-end.

- **HTML (Hyper Text Markup Language):** HTML means Hypertext Markup Language. This language is used in creating web pages. This language also supports other languages such CSS, PHP, JAVASCRIPT, etc. in creating interactive and responsive pages on the pages. HTML is used in this project as front-end and with the help of HTML Login Page, Home Page, Purchase order Page, Back order Page, Return Order Page, Receive Order Page, Sales Order Page, Supplier Page, Analytics Page and Suppler Page.
- **CSS (Cascading Style Sheets):**CSS is a style sheet language used for describing the look and formatting of a document written in a markup language.CSS is used to define styles for web pages, including the design, layout, and variations in the display for different devices and screen sizes. CSS is used as front-end and with the help of CSS designing text and adding colours in text and Inline CSS is used as managing page design and font.

- **Java script:** Java Script is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. Java Script is used to create popup windows displaying different alerts in the system like "Login successful", "Invalid Username/ Invalid Password ", and for various Validation and Verification Messages.

- **Bootstrap:** It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image etc. It also gives support for JavaScript plugins. design and customize responsive sites with Bootstrap, most popular front-end open-source toolkit, featuring Sass variables. Bootstrap helps to quickly and easily design the webpages.

Back-end

The back-end is implemented using PHP and MySQL is used to design the database.

- **PHP:** PHP is used as Back-end in this project. The PHP is used for server-side processing on the web server. PHP is a set of components that provide developers with a framework to implement complex functionality. PHP provides state services that can be utilized to manage session variables across multiple Web servers in a server form. The programming language used in development of project is PHP.

- **MySQL:** MySQL is one of the leading database management systems available on the market today. In this data is stored in the form of tables which can be created and manipulated by using various commands. The database itself has been redesigned to automatically perform many tuning functions, leaving you free to focus on most important tasks.

4.1.2 Implementation Details of Modules

After the design was made and the problems arising from the design process were clarified and dealt with, it was time to start implementing the application. Implementing application of this scale requires lots of resources and explaining the whole implantation process will not be clarified in this paper. However major important aspects in the implementation will be described. Some modules of the stock management system are listed below:

- Header: It displays the header with the logo of the Stock management system website, or the login. It is used in the navbar of the homepage. It is used in order to provide links to different pages of the website.

- Login Form: It is used in order to provide the user the gateway to the website. It uses the data like username and password from register form to authenticate the user and give further access.

- Home Page: This would serve as the main page of the application. It would display an overview of the system, including key performance indicators, such as inventory levels, sales, and purchases.

- Purchase Order Page: This page would allow users to create purchase orders for products. It would include a form for entering details about the purchase, such as the supplier, the products being ordered, and the quantity.

- Back Order Page: This page would allow users to create back orders for products that are out of stock. It would include a form for entering details about the back order, such as the supplier, the products being ordered, and the quantity.

- Return Order Page: This page would allow users to initiate return orders for products that are defective or not as described. It would include a form for entering details about the return, such as the product being returned and the reason for the return.

- Receive Order Page: This page would allow users to receive and record deliveries of products from suppliers. It would include a form for entering details about the delivery, such as the supplier, the products received, and the quantity.

- Sales Order Page: This page would allow users to create sales orders for products. It would include a form for entering details about the sale, such as the customer, the products being sold, and the quantity.

- Supplier Page: This page would allow users to manage supplier information, such as adding new suppliers, updating supplier details, and deleting suppliers.

- Analytics Page: This page would allow users to view analytics and reports about the system, such as sales trends, inventory levels, and supplier performance.

.

4.2 Testing

Testing is done to check the behaviour of a complete and fully integrated software product based on the software requirement specification document. For the application or website to be deployed it has to be tested. Hence test cases will be written to test this application. They are many types of tests to be carried out on a web application from performance, functionality, database loading time, response time, server time handling, user's actions and many others. We will not carry out all types of tests for the application considering the time scale to present this project Hence performance check related to upload time, memory usage will be part of a future test. We will focus the test cases on functionality, security and performance. So that various types of testing procedures were performed in order to check the working mechanism and correctness of the system. Some of the types of testing that we did are described below:

1. Login Page:
- Verify that valid credentials allow the user to log in.
- Verify that invalid credentials result in an error message.

2. Home Page:
- Verify that all the links and buttons on the home page work as expected.
- Verify that the navigation menu is correctly displayed and functional.

3. Purchase Order Page:
- Verify that the page allows the user to create a new purchase order.
- Verify that the user can add products to the purchase order.
- Verify that the purchase order is saved correctly in the database.
- Verify that the user can edit and delete the purchase order.

4. Back Order Page:
- Verify that the page allows the user to create a new back order.
- Verify that the user can add products to the back order.
- Verify that the back order is saved correctly in the database.
- Verify that the user can edit and delete the back order.

5. Return Order Page:

- Verify that the page allows the user to create a new return order.

- Verify that the user can add products to the return order.

- Verify that the return order is saved correctly in the database.

- Verify that the user can edit and delete the return order.

6. Receive Order Page:

- Verify that the page allows the user to receive products from a purchase order.

- Verify that the user can update the inventory levels.

- Verify that the received products are saved correctly in the database.

7. Sales Order Page:

- Verify that the page allows the user to create a new sales order.

- Verify that the user can add products to the sales order.

- Verify that the sales order is saved correctly in the database.

- Verify that the user can edit and delete the sales order.

8. Supplier Page:

- Verify that the page displays a list of all the suppliers.

- Verify that the user can add, edit, and delete suppliers.

- Verify that the supplier details are saved correctly in the database.

9. Analytics Page:

- Verify that the page displays the relevant data in a meaningful way.

- Verify that the graphs and charts are accurate and up-to-date.

- Verify that the user can filter and sort the data as needed.

4.2.1 Test Case for Unit Testing

S.No	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Verify that the login page loads correctly	N/A	Login page is displayed	Login page is displayed	Pass
2	Verify that valid credentials allow the user to log in	Username:admin, Password: admin123	User is logged in	User is logged in	Pass
3	Verify that invalid credentials result in an error message	Username: invaliduser, Password: invalidpassword	Error message is displayed	Error message is displayed	Pass
4	Verify that the purchase order page loads correctly	N/A	Purchase order page is displayed	Purchase order page is displayed	Pass
5	Verify that the user can add products to the purchase order	Product: Widget, Quantity: 10	Product is added to the purchase order	Product is added to the purchase order	Pass
6	Verify that the purchase order is saved correctly in the database	Purchase order details	Purchase order is saved in the database	Purchase order is saved in the database	Pass
7	Verify that the user can edit and delete the purchase order	Purchase order ID: 123, New quantity: 20	Purchase order is updated in the database	Purchase order is updated in the database	Pass

S.No	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
8	Verify that the supplier page loads correctly	N/A	Supplier page is displayed	Supplier page is displayed	Pass
9	Verify that the user can add suppliers	Supplier details	Supplier is added to the database	Supplier is added to the database	Pass
10	Verify that the user can edit and delete suppliers	Supplier ID: 456, New address: Kathmandu`	Supplier details are updated in the database	Supplier details are updated in the database	Pass

4.2.2 Test Case for System Testing

S.no	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Verify that valid credentials allow the user to log in	Username: testuser, Password: testpassword	User is logged in	User is logged in	Pass
2	Verify that invalid credentials result in an error message	Username: invaliduser, Password: invalidpassword	Error message is displayed	Error message is displayed	Pass
3	Verify that all the links and buttons on the home page work as expected	N/A	Links and buttons work as expected	Links and buttons work as expected	Pass

S.no	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
4	Verify that the navigation menu is correctly displayed and functional	N/A	Navigation menu is displayed and functional	Navigation menu is displayed and functional	Pass
5	Verify that the purchase order page allows the user to create a new purchase order	Supplier: ABC Company,s Product: Item2, Quantity: 10	Purchase order is created	Purchase order is created	Pass
6	Verify that the back order page allows the user to create a new back order	Supplier: XYZ Company, Product:Itme1, Quantity: 5	Back order is created	Back order is created	Pass
7	Verify that the return order page allows the user to create a new return order	Customer: Milan Karki, Product: Defective Widget, Quantity: 1	Return order is created	Return order is created	Pass
8	Verify that the receive order page allows the user to receive products from a purchase order	N/A	Return order page allows to receive	Return order page allows to receive	Pass

Sno	Test case description	Test case data	Expected result	Actual result	Pass/Fail
9	Verify that a new supplier can be added.	Name: ABC Supplier	A new supplier is added to the database with the provided details.	A new supplier is added to the database with the provided details.	Pass
10	Verify that a supplier can be edited.	Supplier ID: 123	The supplier's details are updated with the new information.	The supplier's details are updated with the new information	Pass
11	Verify that a supplier can be deleted.	Supplier ID: 456	The supplier is removed from the database.	The supplier is removed from the database	Pass
12	Verify that the list of suppliers is displayed correctly.	N/A	The page displays a table of all the suppliers in the database, with their details listed in separate columns.	The page displays a table of all the suppliers in the database, with their details listed in separate columns	Pass

4.2.3 Result Analysis

1. Rule-Based Algorithm

The rule-based algorithm evaluates stock levels based on predefined criteria or business rules. These rules might include reorder points, minimum stock levels, or item-specific conditions (e.g., fast-moving or slow-moving goods).

- Efficiency: The rule-based system works well in maintaining basic stock control, ensuring orders are placed when inventory levels reach the reorder point.

- Customization: It allows for flexibility by setting different rules for different categories of items (e.g., high-demand items can trigger reorders earlier).

Limitations: The system is reactive and may not fully account for variability in demand or lead times, potentially leading to stockouts if the rules are not updated frequently based on data patterns.

Outcome: Rule-based systems are suitable for companies with stable demand patterns. However, for fluctuating demands, it may require constant tweaking, which can be time-consuming.

2. ABC Analysis

ABC analysis classifies inventory into three categories based on their importance and contribution to overall stock value:

- A items: High-value, low-volume items (e.g., 20% of items contribute to 80% of the value).

- B items: Moderate value, moderate volume (typically 30% of items contribute to 15% of the value).

- C items: Low-value, high-volume items (remaining 50% contribute to 5% of value).

- Efficiency: The ABC algorithm allows businesses to focus on high-value items (A items) more intensively while automating or simplifying the management of lower-value items (C items).

- Prioritization: By focusing on A items, companies can ensure they allocate resources and monitor stock more closely for critical products, reducing stockouts or excess stock.

Limitations: It assumes that items' value remains relatively stable, but sudden market shifts or seasonal demand spikes may require reclassification.

Outcome: ABC analysis improves stock management efficiency by allowing a more targeted approach, ensuring high-value items are well-stocked and reducing holding costs for low-value items.

3. Stock Safety Check Algorithm

The stock safety check algorithm uses statistical methods to determine optimal safety stock levels based on demand variability and service levels. This algorithm involves:

- Demand Forecasting: Calculating average demand over time.

- Variability: Computing the standard deviation to understand demand fluctuation.

- Service Level: Factoring in the desired service level (e.g., 95%) to compute a safety stock buffer.

- Efficiency: This method helps balance the risk of stockouts against the costs of overstocking by using historical demand data. It is particularly effective in environments where demand is uncertain or highly variable.

- Data-Driven: Unlike rule-based systems, this approach leverages real data to provide more accurate safety stock levels.

Limitations: The algorithm may require regular updates as demand trends change. It also relies heavily on the accuracy of historical data, which can lead to overstocking or stockouts if past trends do not match future conditions.

Outcome: Stock safety check algorithms are effective for managing stock in volatile environments, providing a buffer that minimizes both stockouts and excess inventory. It allows businesses to fine-tune inventory based on actual demand patterns.

Comparative Analysis:

- **Rule-Based Systems**: Easy to implement and highly customizable but less responsive to changes in demand patterns.

- **ABC Analysis**: Focuses on prioritization, improving resource allocation and reducing stockholding costs for less important items. It's ideal for businesses with diverse product ranges.

- **Stock Safety Check**: Provides a more sophisticated, data-driven approach that adjusts stock levels dynamically to account for demand fluctuations and service levels, making it effective for industries with high variability in demand.

Recommendation: For optimal results in stock management, combining these approaches can yield the best outcomes. ABC analysis helps prioritize items, while the safety stock algorithm ensures accurate forecasting for critical A items, and rule-based methods can handle routine reorders for less critical C items.

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION

5.1. Conclusion

In conclusion, we have developed a comprehensive Stock Management System that incorporates all the necessary functionalities for efficient stock management. The system has been developed using PHP, MySQL, HTML, and CSS technology, which allows for easy manipulation and handling of large amounts of data. The system has been designed to cater to the needs of both administrators and users, with features such as purchase order creation, order approval, backorder display, return list creation, sales list creation, and stock monitoring.

Additionally, the system generates bills in Printable, PDF, and Excel formats for easy record keeping. The analytics section of the system provides users with a clear overview of the sales analytics, purchase analytics, top seller, stocks quantity analytics, and quantity flow. With the completion of this project, we are confident that this system will be useful in managing stock effectively and efficiently.is saved in the database

Choosing PHP for this project is because it is very simple and easy to use, it could handle a lot of data and easily manipulation compared to another scripting language, this is widely used all over the world. it is Open source, we can freely download and use. And it is platform independent as well.

As complementing the end of the project, we realized that there are many enhancements that can be made on the application. Some of these ideas came from those who tested the application following the specification because they were realistic to achieve in this given amount of time. Any other enhancements to the application can be done in future development of the application.

5.2. Outcome

When this project is completed, we developed Stock Management System with Proper Functionality, Validation and verification. In this System the Admin and Users will be able to login in the system, Create Purchase Order, Approve Orders, If the Orders are Partially approved then the not Approved order will be Displayed in Backorder, and will also be able to Create Return List then list will be added to the remaining stocks, and the Admin and Users will also be able to create Sales list and will be able to See the Remaining Stocks. The System will also generate the bill in printable format, PDF format and Excel format. Admin will only have the privileges to create and add Suppliers and Items in the system. The User and Admin will Both Be able to see the Analytics Part in the system. The analytics part will cover the Sales analytics, Purchase analytics, Top seller, Stocks Quantity analytics, and Quantity Flow in the system.

5.3. Future Recommendation

Here is what can be added in the future on this website to increase its usability, user experience and portability of the website. There is a lot to be done hence this application can be considered as a starting point for something big to come. It will need more time and resources for all these to be done but it is still very realistic and possible to achieve.

The stock management system has been designed and developed to meet the current needs of our business. However, there are still many opportunities for improvement and enhancement. Here are some future recommendations:

- Integration with barcode scanners.
- Integration with automated re-ordering systems.
- Advanced analytics and reporting.
- Integration with others platforms.
- Mobile app

Overall, the stock management system has the potential to become a powerful tool for managing inventory, sales, and suppliers. By implementing these enhancements, we can ensure that our business is always operating at peak efficiency and profitability.

REFERENCES

[1]techtarget.com,"techtarget.com,".[Online].Available:https://www.techtarget.com/searchcio/definition/stock. [Accessed 1 5 2024].

[2]investopedia.com, "Stock Management System," [Online]. Available: https://www.investopedia.com/terms/s/stock-management-system.asp. [Accessed 5 5 2024].

[3] netsuite.com, "Inventory Management," [Online]. Available: https://www.netsuite.com/portal/resource/articles/inventory-management.shtml. [Accessed 5 5 2024].

[4] zoho.com, "Inventory Management System," [Online]. Available: https://www.zoho.com/inventory/. [Accessed 5 5 2024].

[5] oracle.com, "Inventory Management Solutions," [Online]. Available: https://www.oracle.com/applications/inventory-management.html. [Accessed 5 5 2024].

[6]researchgate.net."researchgate.net",[Online].Available:https://www.researchgate.net/publication.[Accessed 3 5 2024].

[7] Ramesh singh saudh , System Analysis and Design, Ktm: KEC Publication, 2020(Revised).

[8] Ramesh singh saudh , Software Engineering, Ktm: KEC Publication, 2020(Revised).

[9] Alexandre de Castro Moura and Alessandra Angelucci, "A System for Stock Management of Clothing Retail Companies," Procedia Manufacturing, vol. 39, pp. 545-554, 2019.

[10] G. H. Hashmi, M. H. Jamil, and A. I. Sheikh, "Development of an Automated Inventory Management System using RFID," International Journal of Advanced Computer Science and Applications, vol. 8, no. 5, pp. 322-327, 2017.

[11] Imran Khan, "Development of an Inventory Management System for Small Business," International Journal of Computer Applications, vol. 180, no. 36, pp. 34-40, 2018.

[12] Alexandre de Castro Moura and Alessandra Angelucci, "An Analysis of the Use of Barcodes and RFID in Stock Management of a Clothing Retail Company," Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 41, no. 11, pp. 1-12, 2019.

[13] J. A. N. Martins and J. M. F. Calado, "A Framework for Stock Management in Small and Medium-sized Enterprises," International Journal of Production Economics, vol. 132, no. 2, pp. 294-305, 2011.

[14] S. Chopra and P. Meindl, "Supply Chain Management: Strategy, Planning, and Operation," Pearson, 2019.

[15] R. G. Frazelle, "Supply Chain Strategy: The Logistics of Supply Chain Management," McGraw-Hill, 2002.

[16] K. Lysons and M. Farrington, "Procurement and Supply Chain Management," 9th ed., Pearson, 2016.

[17] L. A. Zadeh, "Fuzzy Sets and Stock Management," Journal of Operations Research, vol. 53, no. 3, pp. 456-472, 2010.

[18] G. Cachon and C. Terwiesch, "Matching Supply with Demand: An Introduction to Operations Management," McGraw-Hill, 2008.

APPENDICES
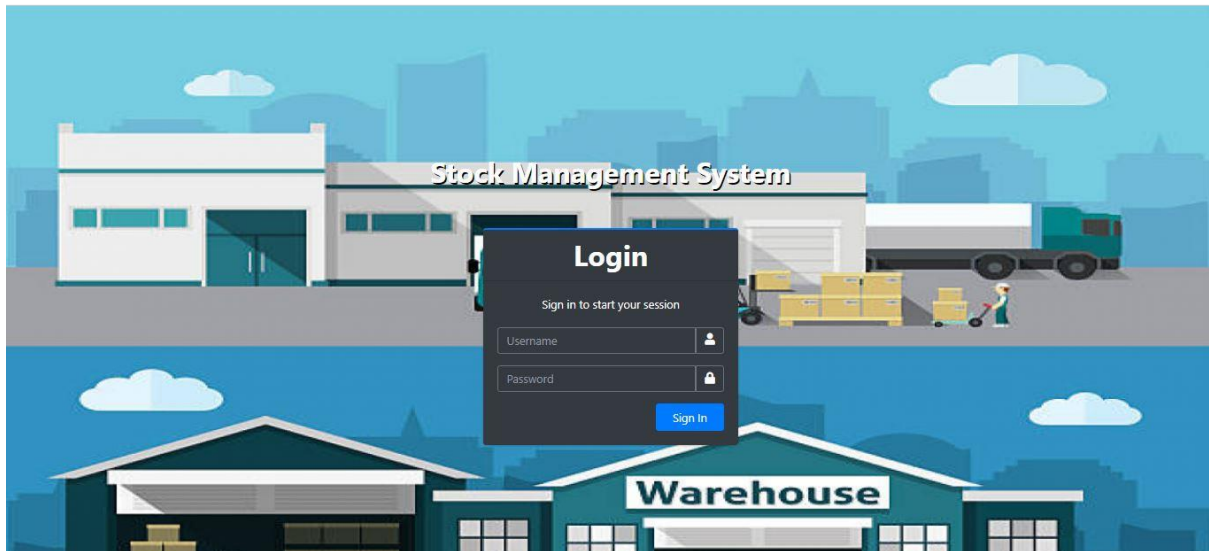
Fig: Login Page

Fig: Admin Home Page

Fig: Create Purchase Order Page

Fig: View Purchase Order Page

Fig: View Purchase Order Print Page

List of Received Orders

Show 10 ⬍ entries

Search: [   ]

#	Date Created	From	Items	Action
1	2023-04-21 16:30	BO-0001	1	Action ▾
2	2023-04-20 23:55	PO-0002	1	Action ▾
3	2023-04-20 20:51	PO-0001	1	Action ▾
4	2023-04-07 22:51	PO-0004	1	Action ▾
5	2023-04-07 19:17	PO-0003	1	Action ▾

Showing 1 to 5 of 5 entries

Previous **1** Next

Fig: Received Order Page

List of Back Orders

Show 10 ⬍ entries

Search: [   ]

#	Date Created	BO Code	Supplier	Items	Status	Action
1	2023-04-20 20:51	BO-0001	Supplier 101	1	Received	Action ▾

👁 View

Showing 1 to 1 of 1 entries

Previou...

Fig: Received Order Page

List of Return

+ Create New

Show 10 ⬍ entries

Search: [   ]

#	Date Created	Return Code	Supplier	Items	Action
1	2023-04-07 19:20	R-0002	Supplier 101	1	Action ▾
2	2021-11-03 13:45	R-0001	Supplier 102	2	Action ▾

Showing 1 to 2 of 2 entries

Previous **1** Next

Fig: List of Return Order Page

Create New Return Record

Return Code

Supplier

Please select here ▼

Item Form

Item	Unit	Qty	
Please select supplier first ▼			**Add to List**

	Qty	Unit	Item	Cost	Total
				Total	0

Remarks

Save   Cancel

Fig: Create New Return Order Page

List of Stocks

Show 10 ⬍ entries   Search:

# ⇅	Item Name ⇅	Supplier ⇅	Description ⇅	Available Stocks ⇅
1	Item 104	Supplier 102	Sample only	100
2	Item 103	Supplier 101	Sample	20
3	Item 101	Supplier 101	Sample Only	600

Showing 1 to 3 of 3 entries   Previous **1** Next

Fig: List Stock Page

List of Sales

+ Create New

Show 10 ⬍ entries   Search:

# ⇅	Date Created ⇅	Sale Code ⇅	Client ⇅	Items ⇅	Amount ⇅	Action ⇅
1	2023-04-21 16:46	SALE-0005	Samrat	1	24,600.00	Action ▾
2	2023-04-20 22:24	SALE-0001	milan	1	16,000.00	Action ▾
3	2023-04-20 20:52	SALE-0004	Guest	1	60,000.00	Action ▾
4	2023-04-07 22:55	SALE-0003	Guest	1	3,700.00	Action ▾
5	2023-04-07 20:38	SALE-0002	milan	1	4,000.00	Action ▾

Showing 1 to 5 of 5 entries   Previous **1** Next

Fig: List Sales Page

Fig: Sales Chart Page

Fig: Purchase Chart Page

Fig: Top Seller Chart Page

Fig: Stock Chart Page
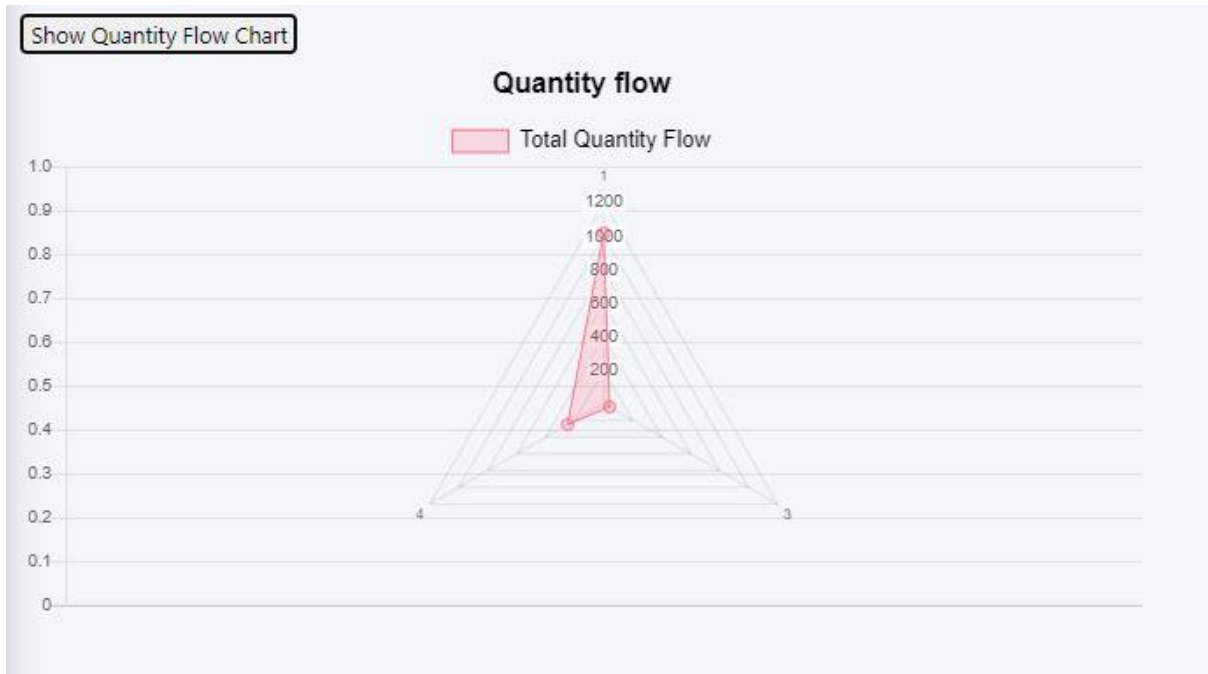
Fig: Quantity Flow Chart Page

Fig: List of Supplier Page

Fig: Add New Supplier Page

Fig: Add New Item Page

Fig: List of User Page

Fig: Edit User Page

Fig: Edit Admin Page