INVENTORY MANAGEMENT SYSTEM

Mphoyaone Frekkie: Student Number Cse22-060
Aobakwe Moerane: Student Number Cse22-063
Motheo Thipe: Student Number Cse22-047
Tapologo Sekgorotha: Student Number Cse18-073

March 2024

Contents

Chapter 1

Abstraction

Effective inventory management is essential to the success and profitability of businesses. Financial strategy and operational efficiency are improved by effectively managing inventory levels. Ineffective management can cause delays, stock imbalances, and financial losses, order fulfilment, as well as unhappy clients. The goals, methods, and best practices of inventory management frameworks are the main topics of this chapter. Maintaining a balance between overstocking and understocking is the main objective in order to maximize expenses and guarantee sufficient working capital. To efficiently control and manage inventory, the chapter explores several inventory management strategies. As a best practice, accurate forecasting is emphasized as a way to match supply and demand, save carrying costs, improve cash flow, and simplify the supply chain. The literature is examined, with a focus on the function of inventory management within the larger framework of supply chain management. KEYWORDS: supply chain management, inventory management, cost efficiency, demand forecasting, safety stock, Economic Order Quantity.

Chapter 2

Introduction

The process of effectively managing the continuous movement of units into and out of an existing stock of products is known as inventory management. Controlling the transfer of units is typically part of this process to keep inventory from growing too large or decreasing to the point where it could endanger the business's ability to operate. (Khan, A., Ansari, A., Ghalib, M., Pandit, H. 2021). The main goal of inventory management is to designate the quantity and location of stocked items. It is essentially a science that deals with defining the percentage and form of stocked commodities. In order to safeguard the regular and planned flow of production against the unforeseen disruption of running out of materials or goods, inventory management is necessary at several places inside a facility or within many sites of a supply network. Inventory management encompasses various aspects such as lead times for replenishment, carrying costs, asset management, forecasting, valuation, visibility, future price forecasting, physical inventory, available space for inventory, quality control, replenishment, returns, defective goods, and demand forecasting.(Khan, A., Ansari, A., Ghalib, M., Pandit, H. 2021).

2.0.1 Motivation

This study aims to enhance comprehension of the server center inventory management redesign, moving away from the outdated traditional method of manually maintaining equipment files and generating reports from them. Additionally, the goal is to foster greater transparency among inventory users.

Chapter 3

Background Literature

Most inventory management systems now on the market rely on employees to use online applications to track and record products in the inventory. These web apps were created in response to numerous researchers who encountered issues of a similar nature with their clients when their inventory was handled manually, which resulted in laborious and ineffective work.

One of the findings about inventory management conducted with focus of reducing errors in recording product stock into the warehouse inventory and improve the efficiency of the process of moving products into and out of the warehouse inventory, an inventory management system has been designed for a firm in Indonesia. In order to do that, the researcher added some basic functionality to the system, with the overall feature being the ability to examine inventory information data, including product comprehensive reports within a specified time frame and form templates for managing inventory product information. According to the study's findings, the web-based inventory management application's author's work is productive and efficient, which will reduce some of the effort involved in inventory management. (S Pasaribu, 2021).

Nonetheless, the majority of businesses would want to see a variety of report approaches and information that allows them to swiftly examine their inventory and product status. Students from a technological school carried out a comparable development research study to create an inventory management system for the AIKTC Server Center. Similar to this, the AIKTC company's manual tracking and recording of product stock issues led to the development of the inventory management system. Currently, server inventory management is completed only every few days or weeks. Even though the circumstances and usage differ slightly from those of the most recent research study, AIKTC is still required to generate and provide a variety of report types and statistical data, including inventory transaction, purchase, and maintenance reports that are part of the author system requirements. (Aamir Khan, Aasif Ansari, MD Ghalib, 2019)

Inventory Management System An inventory management system, also known as an inventory system, encompasses the process of tracking goods across your entire supply chain, from procurement to production to final sales. It dictates the approach to inventory management within your business. (Munro, 2021) Efficiency and healthy margins are essential when dealing with substantial inventory levels. An effective inventory management system is designed to assist you in achieving these goals. (Munro, 2021)

3.0.1 Objectives of Inventory Management

Boosting Product Sales through Inventory Management

Utilizing inventory management to enhance sales opportunities is another valuable technique. Effective inventory management guarantees that products are readily available for purchase when customers demand them, thus maximizing revenue potential. For instance, a consumer electronics retailer may strategically stock up on the latest gadgets ahead of the holiday season to capitalize on heightened consumer spending. By implementing a robust inventory management system, businesses can elevate sales by minimizing stockouts and ensuring continuous availability of high-demand products. (Sharma, 2023)

Cost Reduction in Inventory Management

Another crucial objective of inventory management is minimizing the expenses linked to storing and handling inventory. This involves implementing strategies aimed at lowering various costs, including those related to storage facilities, insurance premiums, taxes, and shrinkage. A significant aspect of this endeavor involves optimizing the supply chain to diminish holding times and storage requirements. For instance, a furniture retailer may opt for a drop-shipping approach to decrease reliance on extensive warehouse facilities. Moreover, businesses can embrace Just-In-Time (JIT) inventory practices to curtail the duration goods remain in storage, consequently reducing storage and handling expenses. (Sharma, 2023)

Adequate Inventory Management

One of the key aims of inventory management is maintaining sufficient stock levels. This entails ensuring there's enough inventory to fulfill customer demands without excessive overstocking. Achieving this balance involves managing stock levels carefully to avoid unnecessary holding costs and the risk of items becoming obsolete. For example, a retailer specializing in seasonal clothing must meticulously adjust inventory levels to match shifting fashion trends and seasonal demands. Many businesses rely on inventory optimization tools, which factor in historical sales data, seasonal patterns, and prevailing market conditions to sustain optimal stock levels. (Sharma, 2023)

Ensuring Material Availability

A primary goal of inventory management revolves around maintaining a steady flow of materials necessary for production or sales without any disruptions. This goal holds particular importance for manufacturers, as a shortage of essential materials can bring production lines to a standstill, resulting in delays and financial setbacks. For example, an electronics manufacturer must consistently access components such as microchips and circuit boards to adhere to production timelines. Similarly, automotive manufacturers employ advanced forecasting techniques and robust supplier relationships to secure a continuous supply of crucial parts like engines and body panels. (Sharma, 2023)

3.0.2 Inventory Management Techniques

FIFO — first in, first out

FIFO, or first in, first out, stands as one of the prevalent inventory management methods employed in stock operations. This approach prioritizes the utilization of the oldest products first, thereby mitigating the risk of spoilage or obsolescence. Under FIFO, assets are sold, utilized, or disposed of in the sequence of their acquisition. Primarily utilized for accounting purposes, FIFO offers a more precise assessment of an organization's inventory levels and associated costs. (Katana Technologies, 2024)

LIFO — last-in, first-out

LIFO, or last-in, first-out, serves as a method for calculating the cost of inventory allocated to the cost of goods sold. In line with its name, the LIFO inventory valuation technique dictates that the most recently acquired items are the first to be sold. Consequently, during periods of inflation, this method attributes higher costs to goods sold, as the latest purchases are made at the highest prices. Notably, LIFO can result in reduced tax liabilities since the IRS permits businesses to deduct the elevated inventory costs from their taxable income. (Katana Technologies, 2024)

JIT — just-in-time

Just-in-time (JIT) inventory represents a system wherein businesses procure only enough stock to fulfill current customer demand. This system aims to reduce holding costs and guarantee that businesses maintain necessary products in stock at all times. In JIT, businesses synchronize their supply orders with sales planning. By ordering stock solely as required to satisfy demand, sellers can mitigate expenses linked to holding surplus inventory. This strategy enables them to optimize efficiency and profitability by ensuring the availability of appropriate products at the right time without excess stocking. Nevertheless, implementing and sustaining this technique can pose challenges. (Katana Technologies, 2024)

Average Costing

The average costing method, also known as the weighted-average method, involves assigning a cost to inventory items based on the total cost of goods acquired within a specific period, divided by the total number of items: Average cost formula = Total cost of purchases / Number of units purchased With this method, every unit in inventory is assigned a cost equivalent to the average of all costs incurred to acquire the units available for sale. (Katana Technologies, 2024) The average cost per unit is typically calculated periodically, often at the end of each accounting period. Subsequently, all units in inventory are valued at this computed average cost. (Katana Technologies, 2024)

3.0.3 Challenges in the implementation of Inventory Management System

Inventory management systems are critical components of businesses, facilitating the efficient tracking and control of inventory levels. However, several challenges are encountered in the implementation and operation of such systems. Here are some of the challenges:

1. Inventory Inaccuracy: Maintaining accurate inventory records is essential for effective management. However, discrepancies between recorded inventory levels and actual quantities on hand often occur due to errors in data entry, theft, or damage during transit/storage (Smith, 2018).

2. Supply Chain Complexity: Modern supply chains are highly complex, involving numerous suppliers, distributors, and logistics partners. Coordinating these entities and ensuring seamless inventory flow poses significant challenges, particularly in global supply chains where distances and cultural differences add layers of complexity (Jones Robinson, 2019).

3. Demand Forecasting Uncertainty: Predicting future demand accurately is essential for optimizing inventory levels. However, demand forecasting is inherently uncertain, influenced by factors such as market trends, seasonal variations, and unexpected events (Chopra Meindl, 2020).

4. Technology Integration Issues: Implementing inventory management software often requires integration with existing systems, such as enterprise resource planning (ERP) systems and point-of-sale (POS) terminals. Ensuring compatibility and smooth data exchange between different systems can be challenging and time-consuming (Simchi-Levi et al., 2019).

5. Inventory Optimization: Balancing inventory levels to meet customer demand while minimizing holding costs and stockouts is a perpetual challenge. Traditional inventory optimization models may be inadequate in dynamic environments, requiring continuous refinement and adaptation (Gupta et al., 2018).

6. Risk of Obsolescence: Products in inventory are susceptible to obsolescence, especially in industries with rapidly evolving technology or short product lifecycles. Managing obsolete inventory incurs costs and ties up valuable resources that could be allocated elsewhere (Waters, 2019).

3.0.4 Existing Challenges in Inventory Management Systems

Inaccurate Demand Forecasting:

Traditional inventory systems often rely on historical data and simplistic forecasting methods, leading to inaccuracies in demand prediction.

Manual Data Entry and Processing:

Manual data entry processes are time-consuming and error-prone, increasing the risk of inventory discrepancies and inefficiencies.

Lack of Real-time Visibility:

Many inventory systems lack real-time visibility into inventory levels, leading to stock-outs, overstocking, and inefficient order fulfillment processes.

3.0.5 Limited Scalability and Flexibility:

Legacy inventory systems may lack the scalability and flexibility to adapt to changing business requirements and market dynamics.

Poor Integration with Supply Chain Partners:

Inefficient integration with suppliers, distributors, and other supply chain partners can lead to delays, errors, and inefficiencies in inventory management processes.

3.0.6 Bridging the Gap

To bridge the gap in inventory management systems, our proposed approach focuses on the following key areas:

Advanced Demand Forecasting Techniques:

Leveraging machine learning algorithms and data analytics to improve demand forecasting accuracy and optimize inventory replenishment strategies.

Automation and Robotics:

Implementing automation and robotics technologies to streamline inventory handling, picking, and replenishment processes, reducing manual labor and errors.

Real-time Monitoring and Analytics:

Deploying IoT sensors and cloud-based analytics platforms to provide real-time visibility into inventory levels, enabling proactive decision-making and optimization of inventory flows.

Scalable and Cloud-based Solutions:

Adopting scalable and cloud-based inventory management solutions that offer flexibility, scalability, and seamless integration with existing systems and supply chain partners.

Collaborative Supply Chain Platforms:

Implementing collaborative supply chain platforms that facilitate seamless communication and collaboration among supply chain partners, enabling end-to-end visibility and coordination of inventory management processes.

3.0.7 Borrowing from Existing Ideas

Inventory management systems play a pivotal role in efficiently handling the flow of goods and products within an organization. Drawing from existing ideas in inventory management systems offers valuable insights and strategies for enhancing the effectiveness and functionality of new systems. This section explores key concepts and practices borrowed from previous inventory management systems, incorporating them into the development of our own system.

One significant concept borrowed from existing inventory management systems is the utilization of a centralized database for storing product information, supplier details, and order records. By centralizing data management, our system ensures data integrity, consistency, and accessibility across different modules. This approach, inspired by systems implemented by renowned organizations such as Amazon and Walmart, facilitates real-time inventory tracking, efficient order processing, and seamless integration with other business processes.

Furthermore, the implementation of automated stock level monitoring, inspired by systems like SAP and Oracle Inventory Management, enables proactive inventory control and optimization. Through automated alerts and notifications, our system helps prevent stockouts, minimize excess inventory, and streamline replenishment processes. Leveraging algorithms for demand forecasting and trend analysis, our system dynamically adjusts inventory levels to meet fluctuating customer demands and market trends, enhancing operational efficiency and customer satisfaction.

Another key feature borrowed from existing systems is the integration of user-friendly interfaces and intuitive navigation, influenced by user-centric designs adopted by industry leaders like Shopify and Square. By prioritizing simplicity, accessibility, and responsiveness, our system ensures ease of use for both frontline staff and management personnel. Intuitive dashboards, customizable reports, and interactive data visualization tools empower users to make informed decisions, monitor performance metrics, and identify areas for improvement effectively.

Moreover, inspired by the agile methodologies embraced by modern software development teams, our system adopts an iterative and collaborative approach to development and implementation. Drawing from frameworks like Scrum and Kanban, our team emphasizes continuous feedback, rapid prototyping, and incremental enhancements, ensuring the system evolves in response to evolving business needs and user feedback.

3.0.8 Conclusion

The factor in consideration is how efficient inventory management is for businesses aiming to optimize their operations and achieve healthy profit margins. Munro (2021) underscores the significance of an effective inventory management system in overseeing the flow of goods throughout the supply chain. Sharma (2023) further elaborates on the objectives of inventory management, emphasizing its role in boosting product sales, reducing costs, maintaining adequate stock levels, and ensuring material availability. These objectives align with the overarching goal of enhancing operational efficiency and profitability.

To achieve these objectives, businesses employ various inventory management techniques, as outlined by Katana Technologies (2024). FIFO (first in, first out) and LIFO (last in, first out) methods are commonly utilized to manage inventory valuation and cost of goods sold. Additionally, the JIT (just-in-time) approach enables businesses to minimize holding costs by procuring stock based on current demand, ensuring timely availability of products without excess stocking. Moreover, the average costing method provides a systematic way to assign costs to inventory items, facilitating accurate valuation and financial reporting.

Implementing these techniques requires careful consideration of factors such as demand variability, supply chain dynamics, and financial objectives. While each method offers distinct advantages, businesses must assess their suitability based on specific operational requirements and market conditions. Furthermore, integrating these techniques into a comprehensive inventory management system enables businesses to achieve synergy and maximize efficiency across the supply chain.

In conclusion, effective inventory management is essential for businesses to enhance sales, reduce costs, maintain optimal stock levels, and ensure uninterrupted material availability. By incorporating various inventory management techniques into their operations, businesses can streamline processes, optimize resource utilization, and ultimately, improve overall performance and competitiveness in the marketplace.

3.0.9 Recommendations

In order for businesses to optimize inventory management practices,they are highly recommended to prioritize the adoption of a comprehensive inventory management system that encompasses varuious techniques such as JIT and average costing.In addition embracing technological advancements solutions such as inventory management systems software enables businesses to manage processes automatically and improvement of data accuracy.By adopting technological advancements solutions,businesses can drive innovation and enhance their ability to meet customer demands while minimizing costs.

Furthermore,it is important for businesses to find a balance between cutting costs while still offering outstanding customer service.By conducting regular assessments of inventory management procedures and adopting technological advancements businesses can enhance their operational effectiveness,boost their competitive position and achieve long-term sustainable growth by putting these methods into practice.

Chapter 4

ANALYSIS and DESIGN

4.1 Requirement Analysis

A type of behavioral diagram within the Unified Modeling Language (UML) is the use case diagram, specifically designed to provide a visual representation of the functionality offered by a system, such as an inventory management system. It showcases actors involved, their objectives (presented as use cases), and any interdependencies. This diagram aids in identifying the primary elements—actors and use cases—that constitute the system's functionality. In the realm of software and systems engineering, the use case technique is instrumental in documenting the functional requirements of systems, especially from a user's perspective. In the context of an inventory management system, the use case diagram illustrates how different actors, such as "Administrator," "Warehouse Staff," and "Customer," engage in specific tasks within the application.

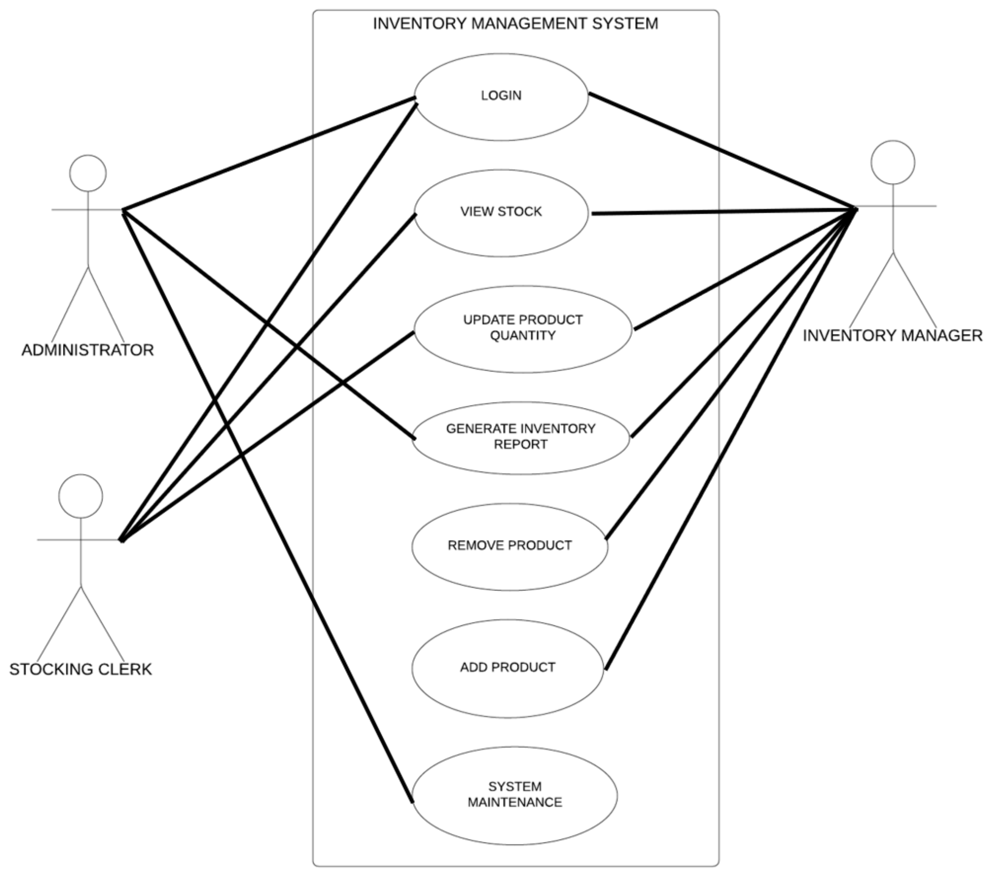

Figure 4.1: Use Case Diagram

4.2 Class Diagram

A class diagram in UML represents the structure of a system by illustrating the classes, attributes, methods, and their relationships. In an inventory management system, this diagram would showcase the various classes such as "Product," "Supplier," and "Inventory," along with their properties and associations, providing a clear picture of the system's structure.

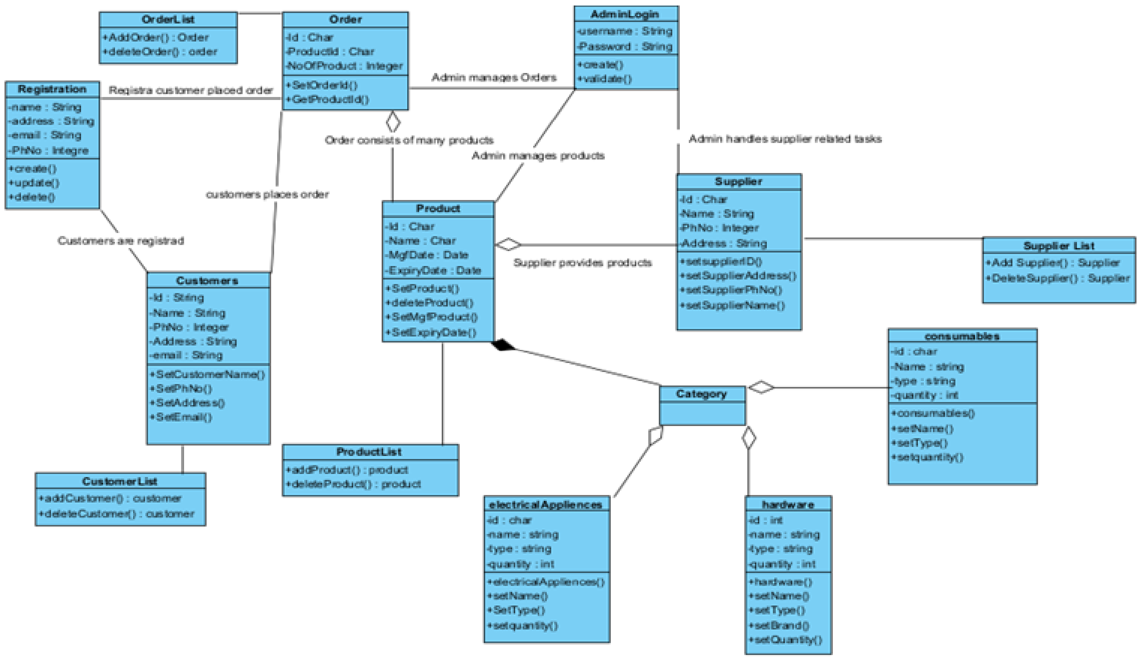

Figure 4.2: Class Diagram

4.3 ERD Diagram

An ERD is a visual representation of the relationships between entities in a database. For an inventory management system, the ERD would display entities like "Product," "Supplier," and "Category," along with their relationships, such as "Supplier supplies Product" and "Order contains Product," helping to understand how data is organized and related within the system.

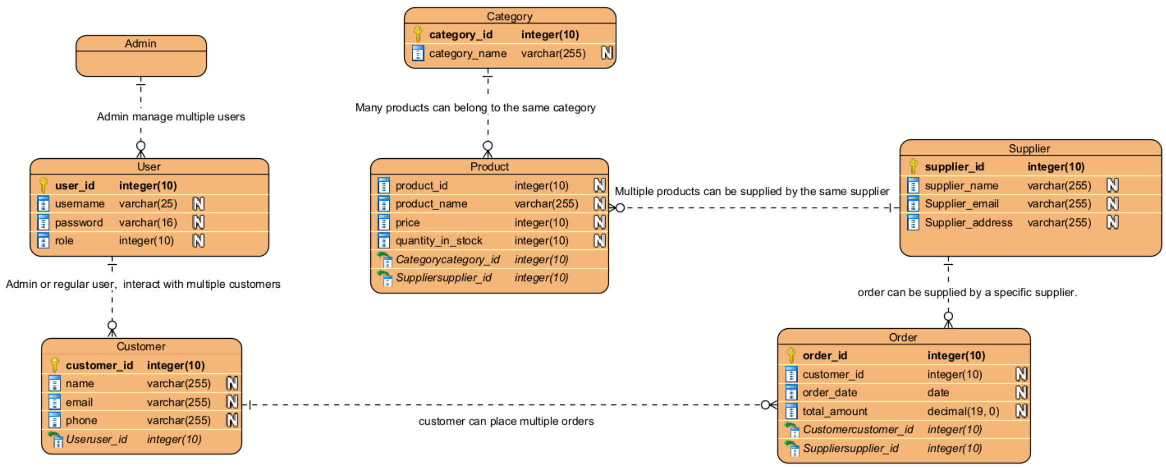

Figure 4.3: Entity Relationship Diagram

4.4 Context Diagram

A context diagram illustrates the external entities (actors) interacting with a system and the system itself. In the context of an inventory management system, it would depict the system at the center and external entities like "Customer," "Supplier," and "Administrator" surrounding it, demonstrating how these entities interact with the system.

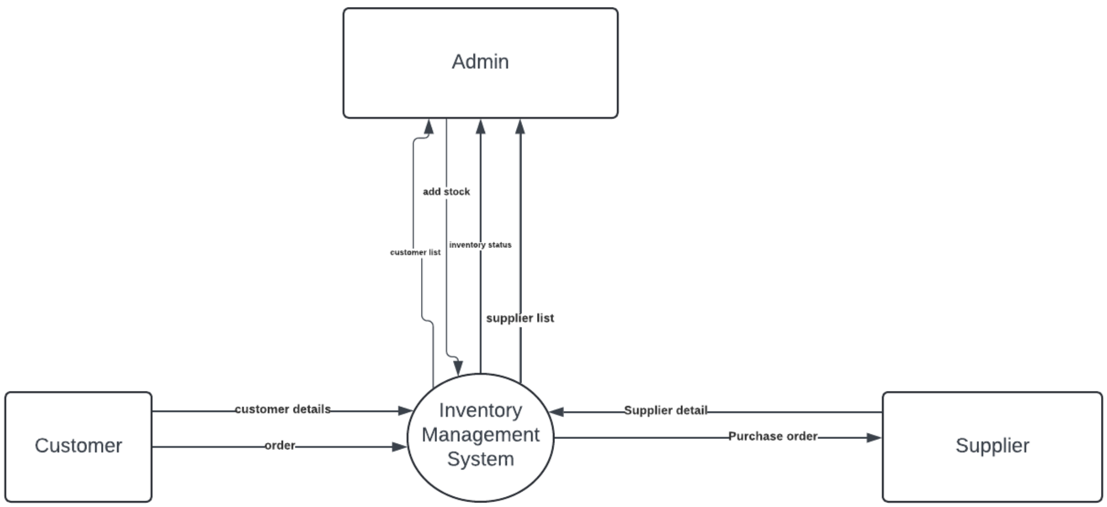

Figure 4.4: Context Diagram

4.5 Activity Diagram

An activity diagram models the workflow or process flow of a system, showing the sequence of activities and decisions. In an inventory management system, this diagram would outline the steps involved in tasks such as "Adding Inventory," "Updating Inventory," and "Removing Inventory," providing a visual guide to the system's operational flow.

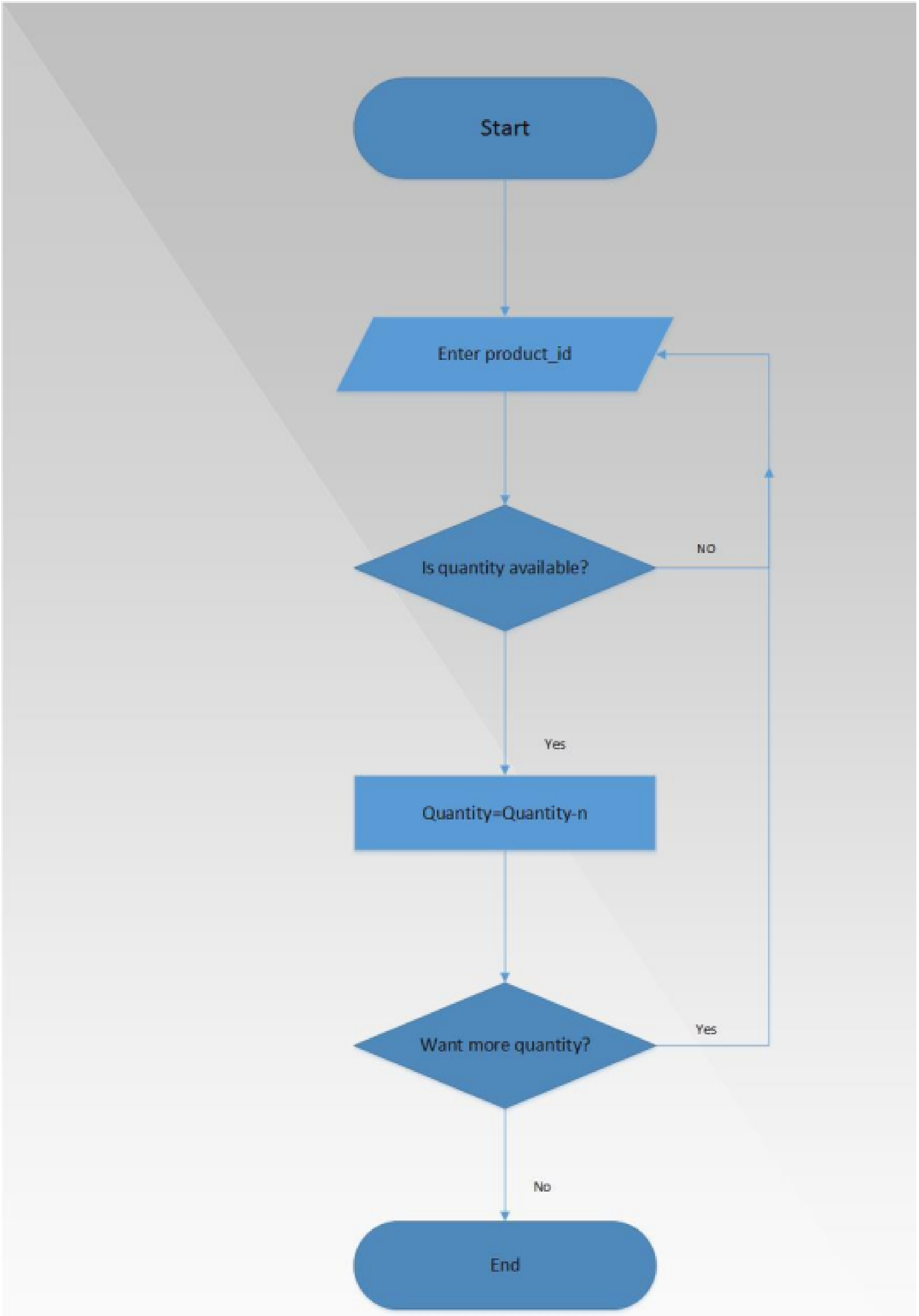

Figure 4.5: Activity Diagram

4.6 Overview of Design

To create a system that is efficient, dependable, and maintained is the aim of systems design. The system needs to meet the specified requirements in order to be effective and limitations. Users who use it to further the business goals of the firm are also required to adopt the system.

If a system can effectively handle defects—such as input errors, processing problems, hardware malfunctions, or human error—it can be considered reliable. Ideally, mistakes can all be avoided.

Planning for errors, identifying them as soon as feasible, enabling their rectification, and preventing them from causing system damage is a more practical strategy for creating a dependable system.

If a system is well-designed, adaptable, and created with future updates in mind, it will be maintainable. No matter how carefully thought out and executed a system is, changes will eventually need to be made. Changes will be required to address issues, adjust to evolving user needs, improve the system, or capitalize on.

System Architecture

Two-tier architecture has been implemented in the system development. Three-tier is a client–server architecture in which consists of presentation logic and database logic.

Two-tier architecture then will have two processing nodes. Layers refer to a logical grouping of components which may or may not be physically located on one processing node.

IMS application will be a desktop application. It will consist of a database server. These server components can exist in the same sever computer or can be different computers such as Database Server.

Component Architecture

The IMS application will have 2 layers which are as:

1. Presentation Layer

2. Data Access Layer

1. Presentation Layer This layer is in charge of giving consumers access to data and information in a format they are familiar with—high level language. It is also accountable for gathering information, reactions, and events from different users. After obtaining the necessary result from the application layer, the presentation layer displays the information in a predefined format in response to the user's request. Java will be used in the development of this layer.

2. Data Access Layer

This layer is in charge of interacting with the database for data storage and retrieval, as well as the presentation layer to obtain additional information. It gathers data from the display layer, processes it as needed, and, if necessary, connects to the database to carry out additional processing. The database result is once more shown to the display layer.

Below is a pictorial representation of the system design cycle:

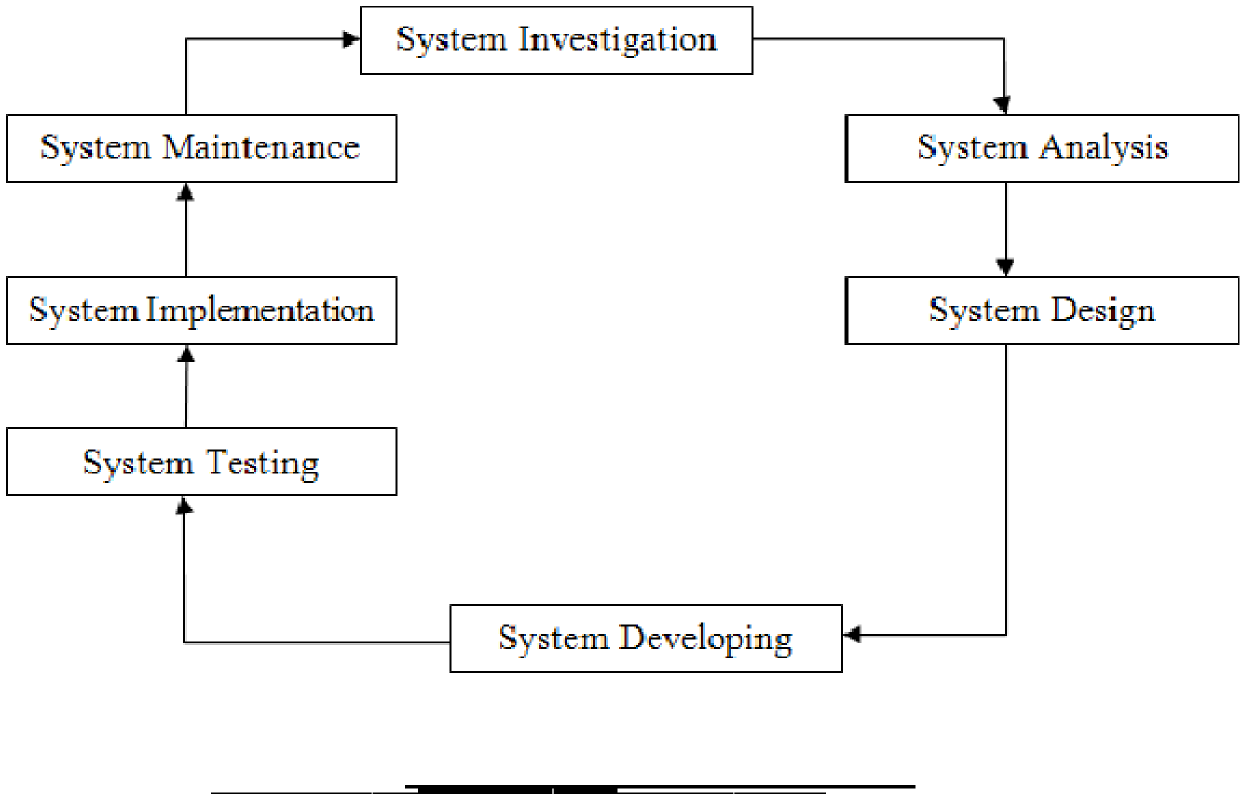

Figure 4.6: System Design Cycle

4.7 Sequence Diagram

An inventory management system's sequence diagram shows the successive interactions that take place during normal operations between the administrator and system components. The administrator starts it by logging in and performing other activities via the user interface. These operations start conversations with the inventory database, which is where things like processing orders, adding or updating products, and changing stock levels happen. Furthermore, interactions with external systems for activities like processing payments or confirming product availability might be shown. Overall, the sequence diagram provides a clear visual representation of the inventory management system administrator's operation.

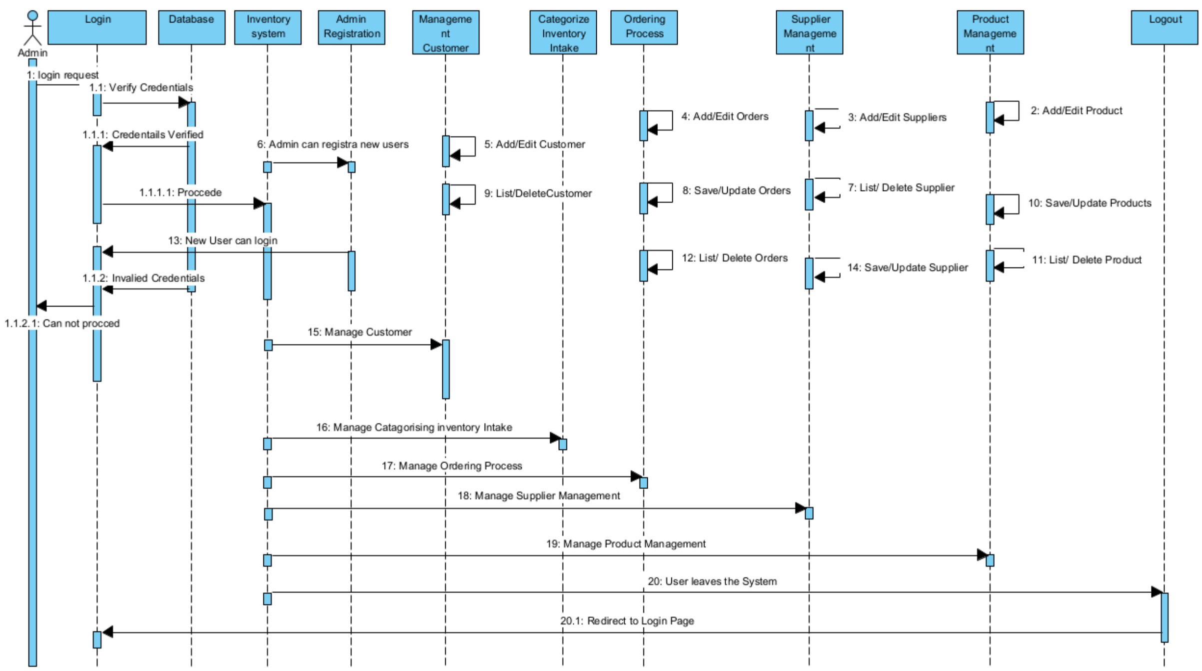

Figure 4.7: Sequence Diagram

Chapter 5

Implementation

LOGIN

This snippet defines an ActionListener for a login button. It retrieves the entered username and password, queries the database to verify credentials, and displays a success message if the login is successful. Otherwise, it shows an error message.

```
public void actionPerformed(ActionEvent e) {
  if (e.getSource() == loginButton) {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());

    try {
      PreparedStatement preparedStatement = connection.prepareStatement("SELECT * FROM
users WHERE username=? AND password=?");
      preparedStatement.setString(1, username);
      preparedStatement.setString(2, password);

      ResultSet resultSet = preparedStatement.executeQuery();

      if (resultSet.next()) {
        JOptionPane.showMessageDialog(this, "Login Successful!");
        // Open the main menu or perform other actions upon successful login
      } else {
        JOptionPane.showMessageDialog(this, "Invalid Username or Password. Please try again.");
      }
    } catch (SQLException ex) {
      ex.printStackTrace();
      JOptionPane.showMessageDialog(this, "Error occurred while attempting to login.");
    }
  }
}
```

ADD PRODUCT

This function prompts the user to input details of a new product, such as name, description, and price. It then inserts this information into the database using a PreparedStatement. Success or failure of the insertion is communicated via JOptionPane messages.

```
private void addNewProduct() {
  String productName = JOptionPane.showInputDialog(this , "Enter Product Name.");
  String productDescription = JOptionPane.showInputDialog(this, "Enter Product Description.");
  double productPrice = Double.parseDouble(JOptionPane.showInputDialog(this, "Enter Product Price.."));

  try{
    PreparedStatement preparedStatement = connection.prepareStatement("INSERT INTO products (name, description, price)
     VALUES (?, ?, ?)");
    preparedStatement.setString(1, productName);
    preparedStatement.setString(2, productDescription);
    preparedStatement.setDouble(3, productPrice);

    int rowsAffected = preparedStatement.executeUpdate();

      if (rowsAffected > 0) {
        JOptionPane.showMessageDialog(this , "Product added successfully.");

      }else {
        JOptionPane.showMessageDialog(this, "Failed to add product. Please try again.");
      }
    }catch (SQLException ex ) {
    ex.printStackTrace();
  }

}
```

Figure 5.1: Login

PRODUCT LIST

In this case, a method gets a product list out of the database. After running a query to retrieve product details, iterating through the result set, creating Product objects, and adding them to a list are all done. For error handling, any SQLException is caught and reported.tatement. Success or failure of the insertion is communicated via JOptionPane messages.

```
private List<Product> retrieveProductList() {
   List<Product> productList = new ArrayList<>();

   try {
      // Establish connection and create a statement
      Statement statement = connection.createStatement();

      // Execute a query to retrieve product information
      ResultSet resultSet = statement.executeQuery("SELECT productId, name FROM products");

      // Process the result set and populate the productList
      while (resultSet.next()) {
         int productId = resultSet.getInt("id");
         String name = resultSet.getString("name");

         // Create a new Product object and add it to the productList
         Product product = new Product(productId, name);
         productList.add(product);
      }
   } catch (SQLException ex) {
      ex.printStackTrace(); // Handle exceptions appropriately
   }

   return productList;

}
```

Figure 5.2: Login

DATABASE CONNECTION

This code establishes a database connection using JDBC to a MySQL database. It includes a method to obtain a connection and another to close it. The connection details such as URL, username, and password are provided as constants.

```
public class DatabaseUtil {
   private static final String DB_URL = "jdbc:mysql://localhost:3306/dbase";
   private static final String USER = "root";
   private static final String PASSWORD = "password";

   // Method to establish database connection
   public static Connection getConnection() throws SQLException {
      return DriverManager.getConnection(DB_URL, USER, PASSWORD);
   }

   // Method to close database connection
   public static void closeConnection(Connection conn) {
      if (conn != null) {
         try {
            conn.close();
         } catch (SQLException e) {
            e.printStackTrace();
         }
      }
   }

   // Additional utility methods for executing queries to be added here
}
```

Figure 5.3: Login

Chapter 6

Testing Discussions

Login
Login screen for accepting user credentials.

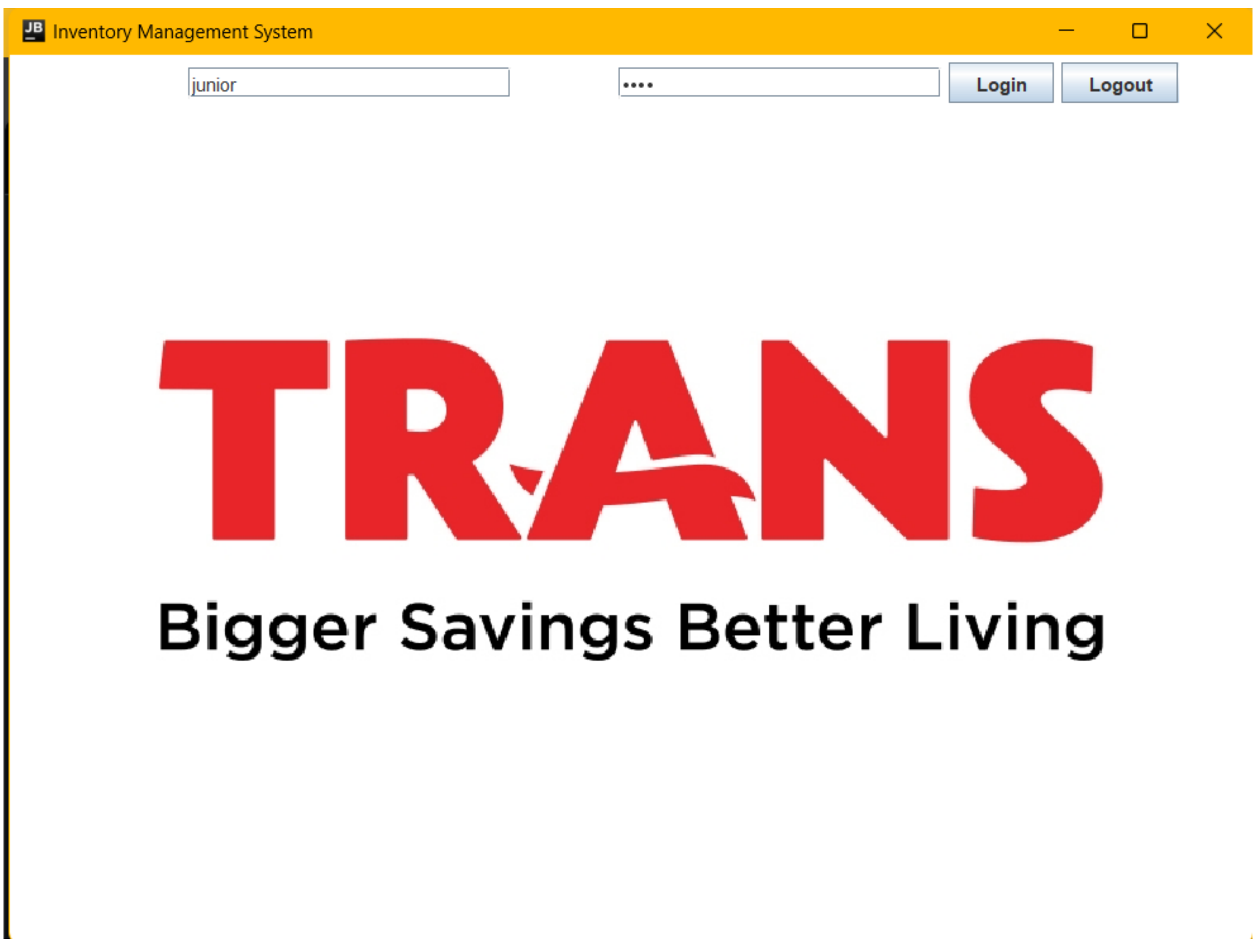

Figure 6.1: login

Search Product
The search where users can find products at ease

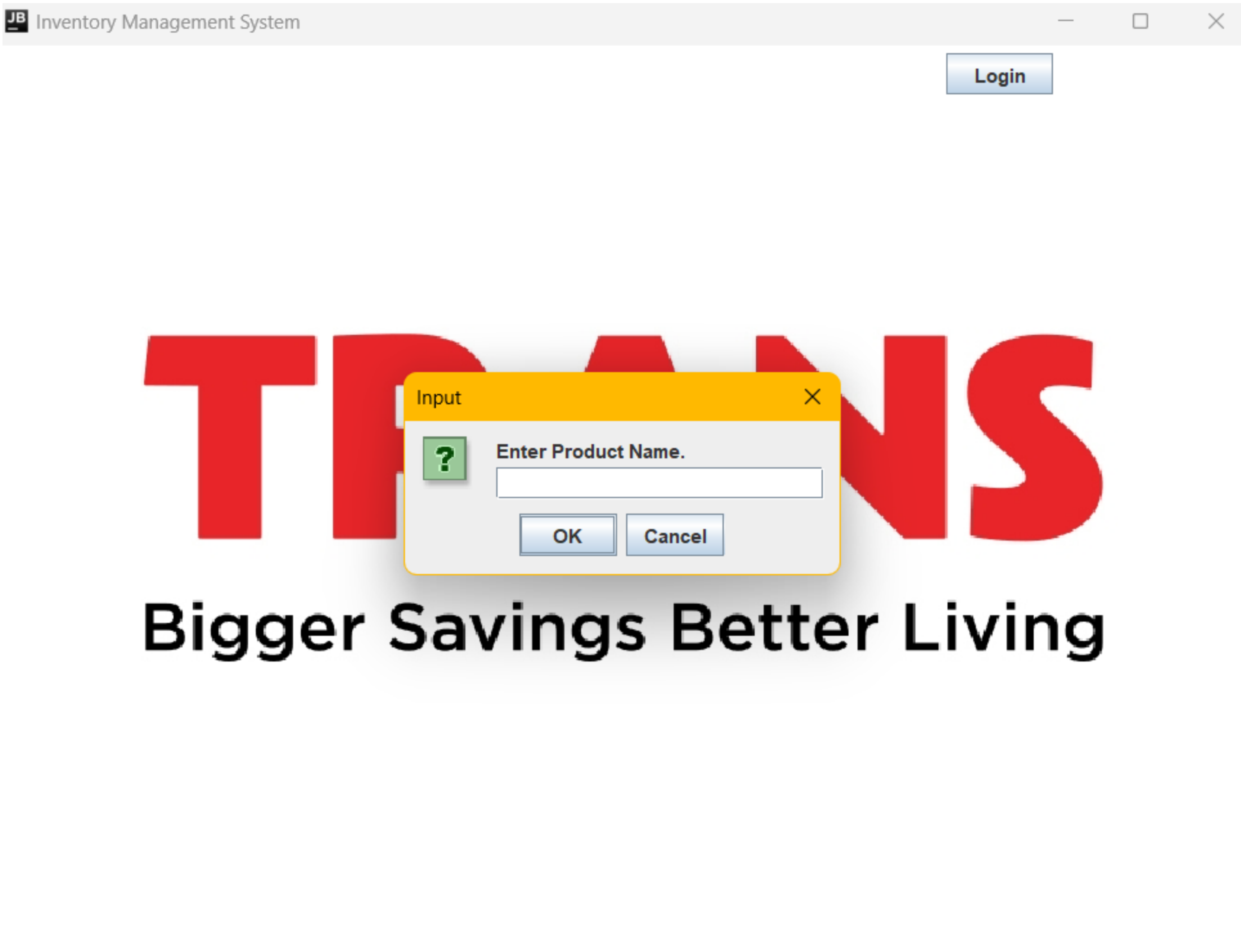

Figure 6.2: Product search

Product List
Screen for displaying products

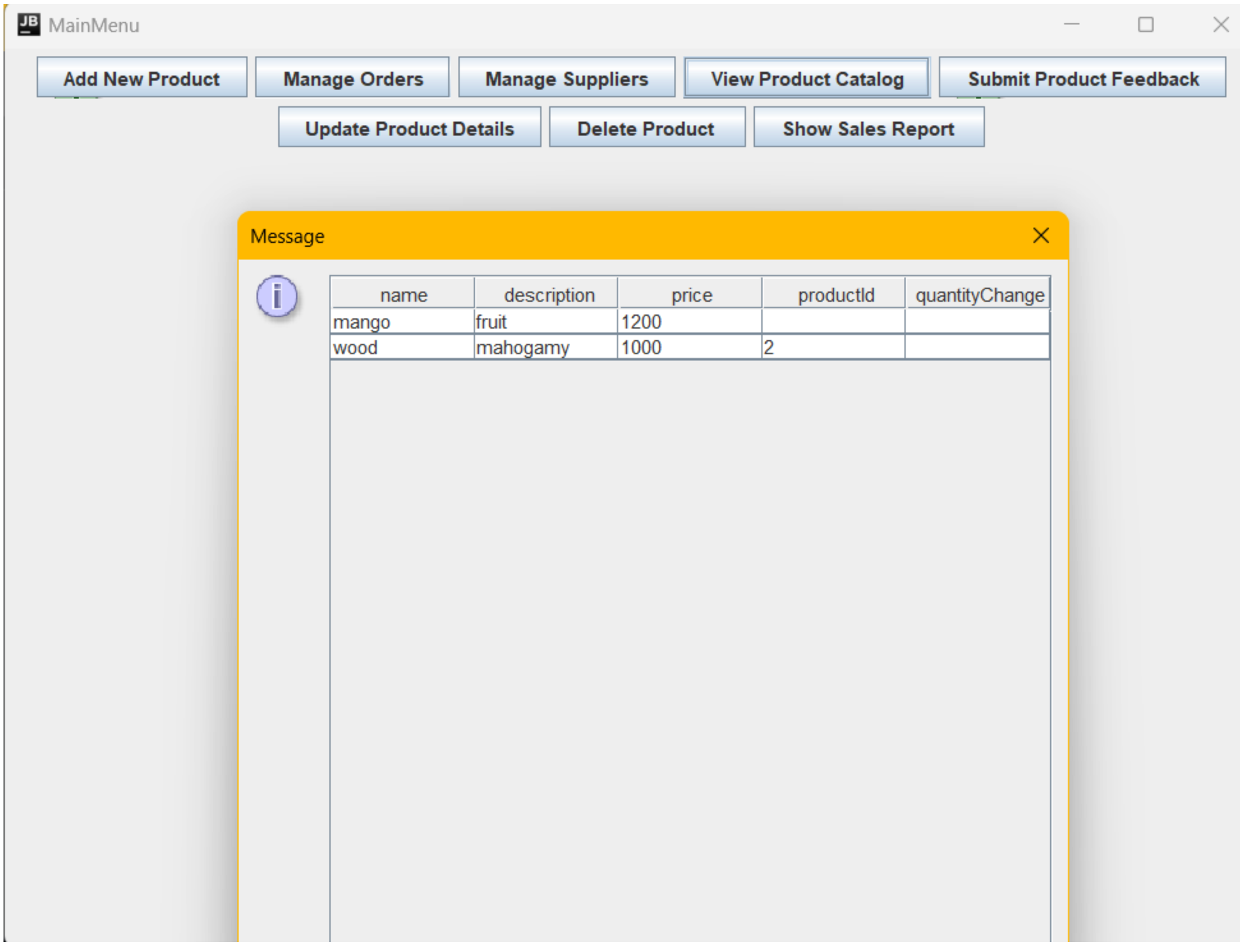

Figure 6.3: product list

Main Menu
Main menu screen displaying navigation to various parts of the system

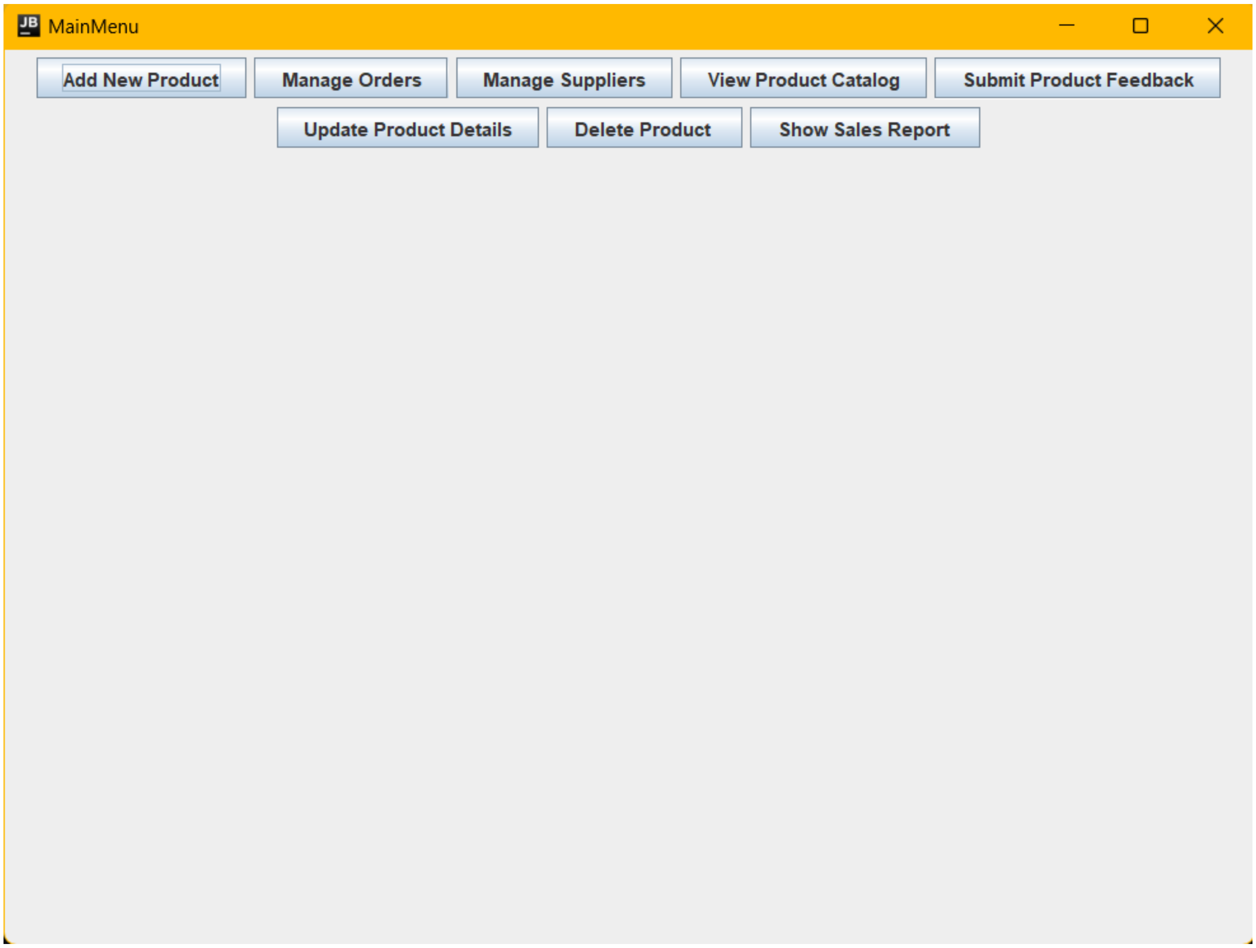

Figure 6.4: main menu

Feedback

Hub where users can provide feedback about products

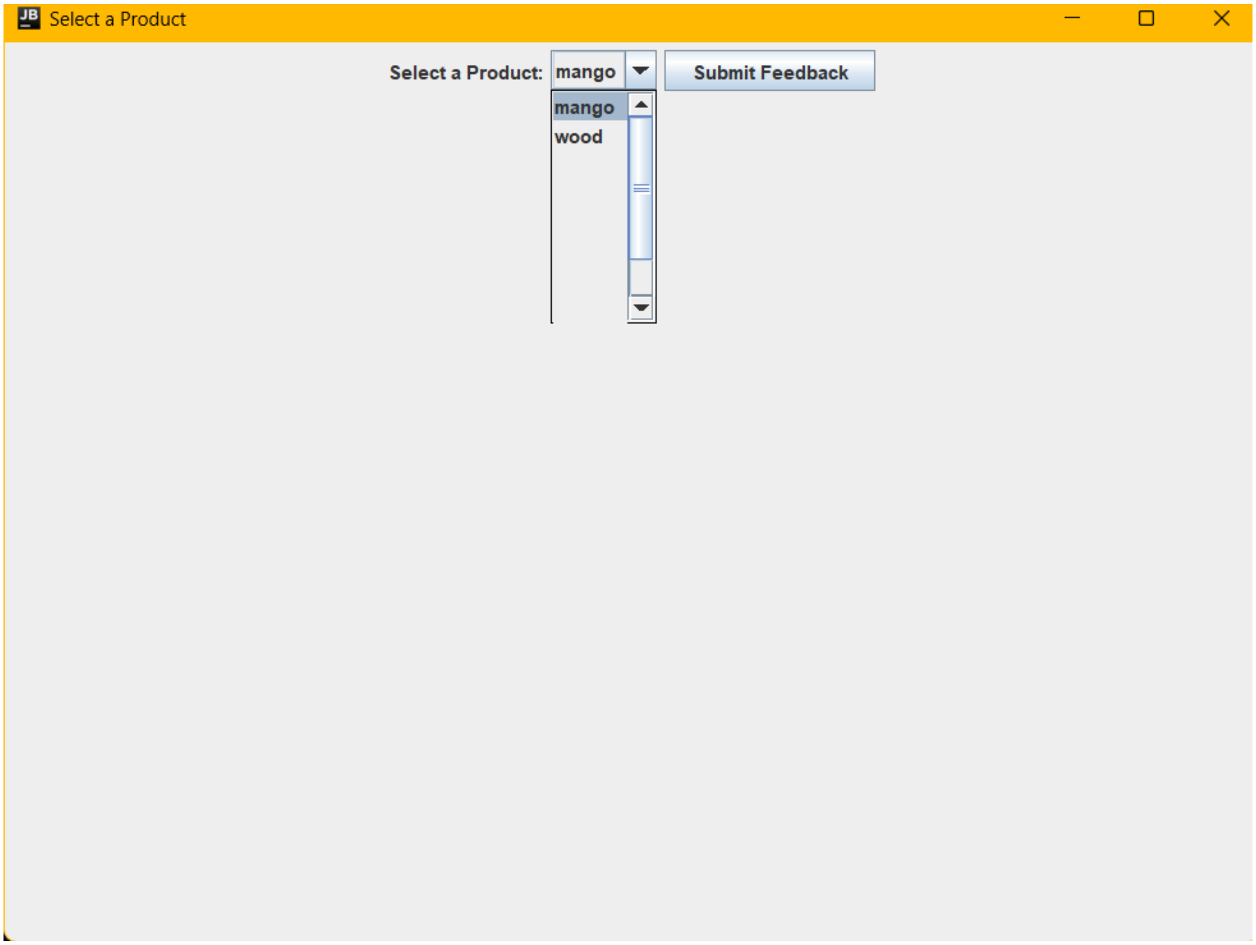

Figure 6.5: feedback

Chapter 7

Conclusion

Inventory Management System, a crucial tool for recording and managing an organization's inventory. Our journey involved continuous modification, rigorous testing, and thorough debugging. Here are the key takeaways:

System Adaptability: The developed system demonstrates remarkable adaptability to changing organizational conditions. While it cannot entirely replace human efforts, it serves as a valuable asset across various projects.

Theoretical Knowledge in Action: The experience we gathered form study about different inventory systems existing , allowed us to bridge the gap between theoretical knowledge and real-world implementation. The system embodies the principles we acquired form our pervious modules such as Database Design and Development and Object Oriented Programming.

Lessons Learned: Teamwork and Cooperation: We sharpened our knowledge base by collaborating effectively within a group.Time Management: Valuing time and workplace efficiency became second nature.Communication and Leadership: We honed our communication skills, leadership qualities, and built strong public relations.Flexibility: Recognizing that institutional requirements may evolve, we ensured our system remains adaptable.

In summary, the Inventory Management System stands as a testament to our learning journey, combining theory and practice. As we move forward, let's continue refining our systems and embracing growth.

Chapter 8

References

- Khan, A., Ansari, A., Ghalib, M., Pandit, H. (2021). A Web Application on Inventory Management System for Server Center of AIKTC. International Research Journal of Engineering and Technology
- S Pasaribu, J. (2021). Development of a Web Based Inventory InformationSystem. International Journal of Engineering, Science andInformation Technology, 1(2). https://doi.org/10.52088/ijesty.v1i2. (PDF) Inventory Management Systems (IMS). Available from: https://www.researchgate.net/publication/37

- Aamir Khan, Aasif Ansari, MD Ghalib, H. P. (2019). IRJET- A WebApplication on Inventory Management System for Server Center ofAIKTC. International Research Journal of Engineering andTechnology, 06(02), 2203–2205.https://www.irjet.net/archives/V6/i2/IRJET-V6I2436.pdf
- Chopra, S., Meindl, P. (2020). Supply Chain Management: Strategy, Planning, and Operation (7th ed.). Pearson Education.
- Gupta, S., Maranas, C. D., Shmoys, D. B. (2018). Inventory Management Under Demand Uncertainty: Structural and Policy Implications. Management Science, 64(6), 2649–2666.
- Jones, P., Robinson, P. (2019). Operations Management (8th ed.). Oxford University Press.
- Simchi-Levi, D., Schmidt, W., Wei, Y. (2019). Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies (4th ed.). McGraw-Hill Education.
- Smith, G. (2018). Fundamentals of Inventory Management and Control. Routledge. Waters, D. (2019). Essentials of Inventory Management (2nd ed.). Routledge.
- Anderson, D., Smith, J. (2019). Inventory Management: A Systematic Approach. Harvard Business Review.
- Chopra, S., Meindl, P. (2016). Supply Chain Management: Strategy, Planning, and Operation. Pearson.
- Lee, H. L. (2004). The Triple-A Supply Chain. Harvard Business Review.
- Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E. (2008). Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies. McGraw-Hill Education.
- Wisner, J. D., Tan, K. C., Leong, G. K. (2011). Principles of Supply Chain Management: A Balanced Approach. Cengage Learning.

- Wang, Z., Mahmood, A. (2012). Supply chain management capabilities, competitive advantage, and firm performance in the context of Taiwan's electronics industry. Journal of Business Research, 65(2), 207-214.
- Lee, H. L., Padmanabhan, V., Whang, S. (2004). Information distortion in a supply chain: The bullwhip effect. Management science, 50(12 supplement), 1875-1886.
- Chopra, S., Meindl, P. (2004). Supply chain management: Strategy, planning, and operation. Prentice Hall.
- Sharma, R. (2007). Supply chain management: concepts, practices, and implementation. Oxford University Press, USA.

Chapter 9

Appendices

9.1 Requirement Catalogue

Requirement ID	Description
REQ-001	The system should allow users to login with a username and password.
REQ-002	The system shall provide separate functionalities for managing orders, suppliers, and product list.
REQ-003	Users with administrative privileges shall be able to add, edit, and delete products from the list.
REQ-004	The system shall maintain a centralized database to store product information, supplier details, and order records.
REQ-005	Users shall be able to view detailed product information including name, description, and price.
REQ-006	The system shall generate alerts for low stock levels and notify users to replenish inventory.
REQ-007	Users shall be able to submit feedback for products including ratings and comments.
REQ-008	The system shall support automated stock monitoring and adjustments.
REQ-009	The user interface shall be intuitive, user-friendly, and accessible across different devices.
REQ-010	The system shall provide customizable reports and analytics dashboards.
REQ-011	Security features shall include user authentication, authorization, and data encryption.
REQ-012	The system shall be scalable to accommodate increasing data volumes and user loads.
REQ-013	Error handling mechanisms shall be implemented to detect and handle exceptions gracefully.
REQ-014	The system shall maintain logs of system activities, user interactions, and errors for auditing purposes.
REQ-015	Integration components shall facilitate seamless data exchange with external systems and services.
REQ-016	The system shall support multiple user roles with different access levels including administrators, managers, and regular users.
REQ-017	Users shall be able to place orders for products specifying quantities.
REQ-018	The system shall track order statuses and provide updates to users regarding order processing.
REQ-019	nventory updates shall be synchronized and reflected immediately after any transaction change.
REQ-020	The system shall provide data backup and recovery mechanisms to ensure the integrity and availability of stored information.

Table 9.1: Table with Requirement IDs and Descriptions

System Requirement

Hardware requirement

- Pentium 3 or above
- 512 MB RAM

Software Requirement

- Windowws operating system
- Java Virtual Machine
- Java Development Kit

User Requirement

- Basic Computer knowledge
- File Browsing Skill

9.2 Report Writing

Chapter	Section	Author
Introduction	1	Aobakwe Moerane
Background Literature	2	All members
Requirement Analysis	3	Mphoyaone Frekkie
Class Diagram	4	Tapologo Sekgorotha
Erd Diagram	5	Motheo Thipe
Context Diagram	6	Tapologo Sekgorotha
Activity Diagram	7	Aobakwe Moerane
Overview of Design	8	Mphoyaone Frekkie
Implementation	9	All
Testing Discussions	10	All
Conclusion	11	Aobakwe Moerane
References	12	Mphoyaone Frekkie
Appendices	13	All

Table 9.2: Table with Chapter, Section, and Author

9.3 Project Schedule

It is critical to comprehend the processes and deadlines involved in job execution when aiming for operational excellence. Gantt charts are used to create a visual representation that highlights the fine details of task fulfillment. These charts show the actions in chronological sequence, but they also include important information about how long each task takes. Organizations can uncover possible bottlenecks, enhance resource allocation, and obtain a thorough grasp of process efficiencies by utilizing Gantt charts. To put it simply, they are a pillar in the effort to improve efficiency and simplify project management procedures.

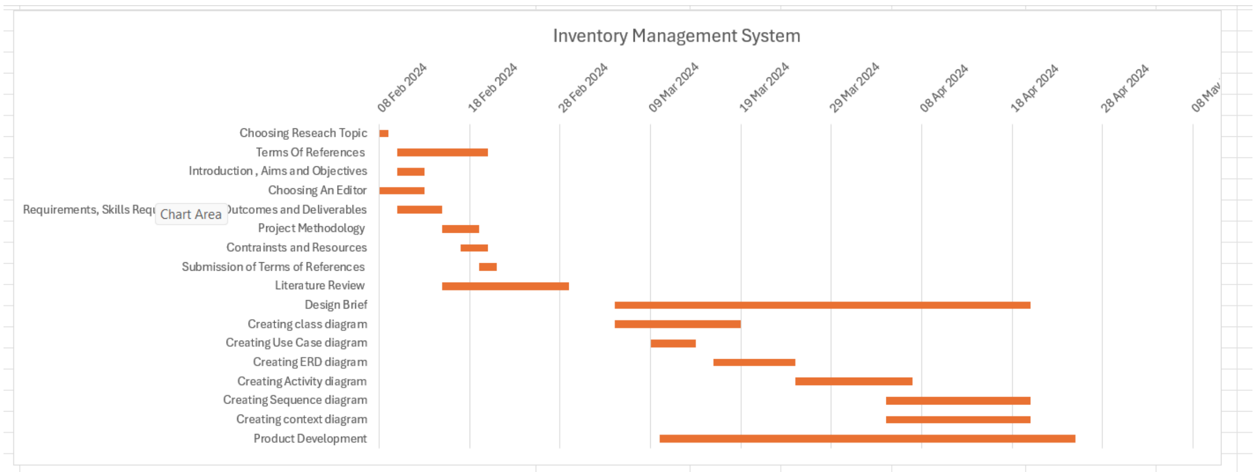

Figure 9.1: Gantt initial Phase

Task No	Tasks	Responsible Perso	Start Date	Days Needed	completion Date	Adjusted Lengt	Progress	Column1
1	Choosing Reseach Topic	All group memebers	08 Feb 2024	5	11/02/2024	3	60%	
2	Terms Of References	All group memebers	11 Feb 2024	15	20/02/2024	9	60%	
2.1	Introduction , Aims and Objectives	All group memebers	11 Feb 2024	5	13/02/2024	2	40%	
2.2	Choosing An Editor	All group memebers	12 Feb 2024	4	15/02/2024	3	75%	
2.3	**Requirements, Skills Required and the Outcomes and Deliverables**	All group memebers	11 Feb 2024	7	15/02/2024	4	57%	
2.4	Project Methodology	All group memebers	15 Feb 2024	7	19/02/2024	4	57%	
2.5	Contrainsts and Resources	All group memebers	19 Feb 2024	3	20/02/2024	1	33%	
2.6	Submission of Terms of References	Aobakwe Moerane	20 Feb 2024	1	21/02/2024	1	100%	
3	Literature Review	All group memebers	07 Mar 2024	30	15/02/2024	21	70%	
4	Design Brief		01 Apr 2024		25/04/2024	24		
4.1	Creating class diagram	Aobakwe Moerane	04 Apr 2024	20	24/04/2024	18	90%	
4.2	Creating Use Case diagram	Mphonyaone	16 Apr 2024	10	23/04/2024	7	70%	
4.3	Creating ERD diagram	Motheo Thipe	16 Apr 2024	10	25/04/2024	9	90%	
4.4	Creating Activity diagram	Motheo Thipe	16 Apr 2024	8	23/04/2024	7	88%	
4.5	Creating Sequence diagram	Aobakwe Moerane	16 Apr 2024	6	23/04/2024	7	117%	
4.6	Creating context diagram	Tapologo	16 Apr 2024	12	25/04/2024	9	75%	
5	Product Development	All group memebers le	11 Apr 2024	27	28/04/2024	17	63%	
6	Testing	Tapologo	25/04/2024	4	28/04/2024	3	75%	
7	Referencing	Mphonyaone	25/04/2024	4	28/04/2024	3	75%	

Figure 9.2: Gantt initial table

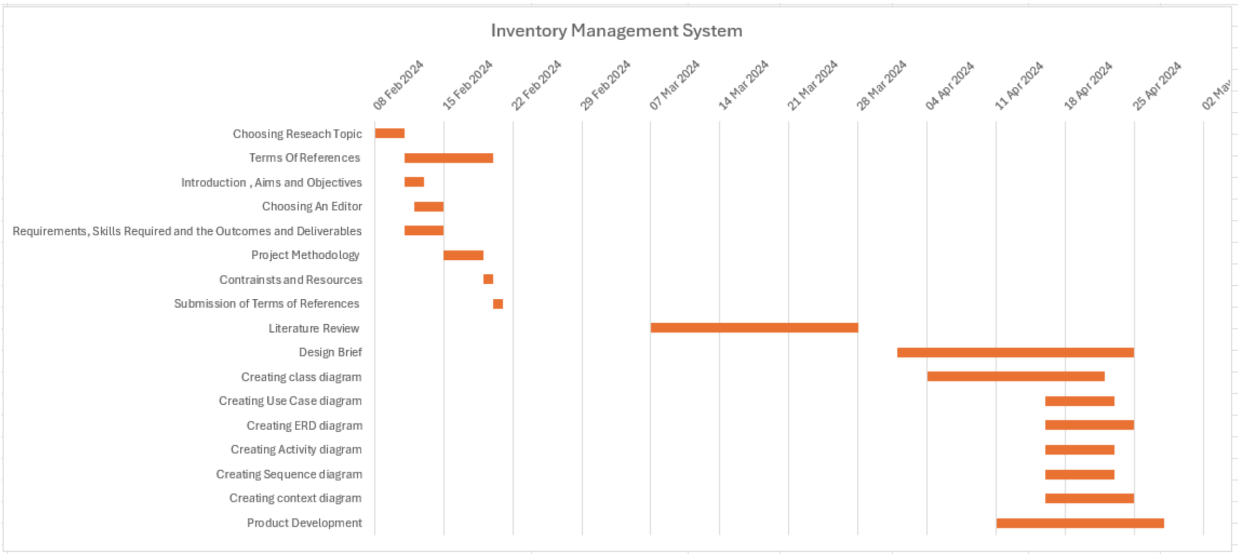

Figure 9.3: Gantt final Phase

Task No	Tasks	Responsible Person	Start Date	Days Needed	completion Date	Adjusted Length	Progress
1	Choosing Reseach Topic	All group memebers	08 Feb 2024	4	09/02/2024	1	25%
2	Terms Of References	All group memebers	10 Feb 2024	15	20/02/2024	10	67%
2.1	Introduction , Aims and Objectives	All group memebers	10 Feb 2024	5	13/02/2024	3	60%
2.2	Choosing An Editor	All group memebers	08 Feb 2024	3	10/02/2024	2	67%
2.3	**Requirements, Skills Required and the Outcomes and Deliverables**	All group memebers	10 Feb 2024	8	15/02/2024	5	63%
2.4	Project Methodology	All group memebers	15 Feb 2024	6	19/02/2024	4	67%
2.5	Contrainsts and Resources	All group memebers	17 Feb 2024	5	20/02/2024	3	60%
2.6	Submission of Terms of References	Aobakwe Moerane	19 Feb 2024	3	21/02/2024	2	67%
3	Literature Review	All group memebers	15 Feb 2024	20	29/02/2024	14	70%
4	Design Brief		05 Mar 2024		20/04/2024	46	
4.1	Creating class diagram	Aobakwe Moerane	05 Mar 2024	18	19/03/2024	14	78%
4.2	Creating Use Case diagram	Mphonyaone	09 Mar 2024	10	14/03/2024	5	50%
4.3	Creating ERD diagram	Motheo Thipe	16 Mar 2024	15	25/03/2024	9	60%
4.4	Creating Activity diagram	Motheo Thipe	25 Mar 2024	17	07/04/2024	13	76%
4.5	Creating Sequence diagram	Aobakwe Moerane	04 Apr 2024	16	15/04/2024	11	69%
4.6	Creating context diagram	Tapologo	04 Apr 2024	20	20/04/2024	16	80%
5	Product Development	All group memebers lead by Motheo Thipe	10 Mar 2024	40	18/04/2024	39	98%
6	Testing	Tapologo	25/04/2024	4	27/04/2024	2	50%
7	Referencing	Mphonyaone	25/04/2024	4	27/04/2024	2	50%

Figure 9.4: Final Gantt table

9.4 ERD diagrams

Entity-Relationship Diagram (ERD) Explanation

Entities and Their Purposes:

1.Category: Purpose: Represents different categories of products. Attributes:

- category Id: Primary key (unique identifier for each category).
- category name: Stores the name of the category.
- Data Types:category id(numeric),category name(text).

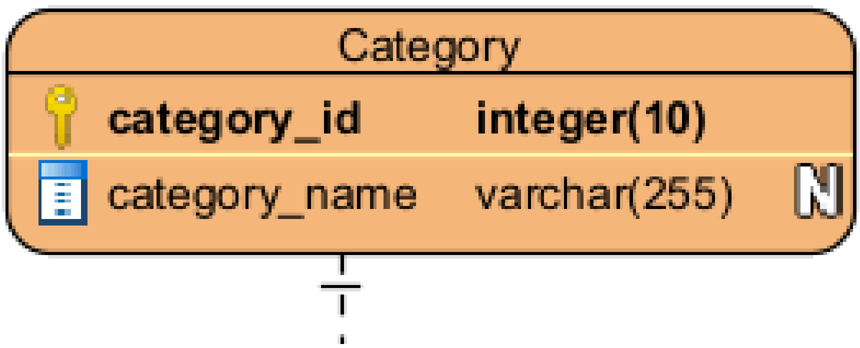

Figure 9.5: CATEGORY

2.Product:
Purpose: Stores details about individual products. Attributes:

- product Id: Primary key (unique identifier for each product).
- product Name: Name of the product.
- price: price of the product.
- quantity in stock: Available stock quantity
- category id: foreign key to category.
- suppliersupplier id: foreign key to supplier.
- DataTypes: product id (numeric), product name (text), description(text),quantity in stock(numeric),price(nu

Relationships: Many products belong to the same category (indicated by the line connecting Product and Category).

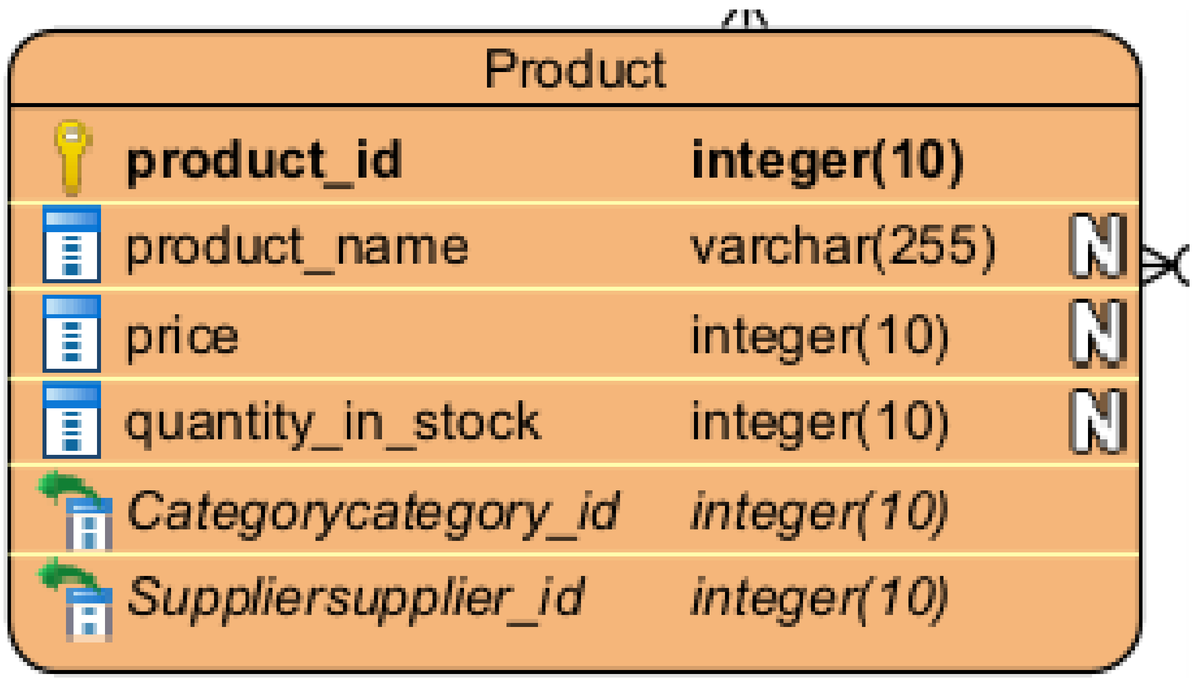

Figure 9.6: PRODUCT

3.Supplier:
Purpose: Contains information about suppliers.
Attributes:

- supplier id: Primary key (unique identifier for each supplier).
- supplier name: Name of the supplier.
- Supplier email: Supplier's email address.
- Supplier address: supplier's address.

Data Types:supplier id(numeric),supplier name(text),contact info(text).

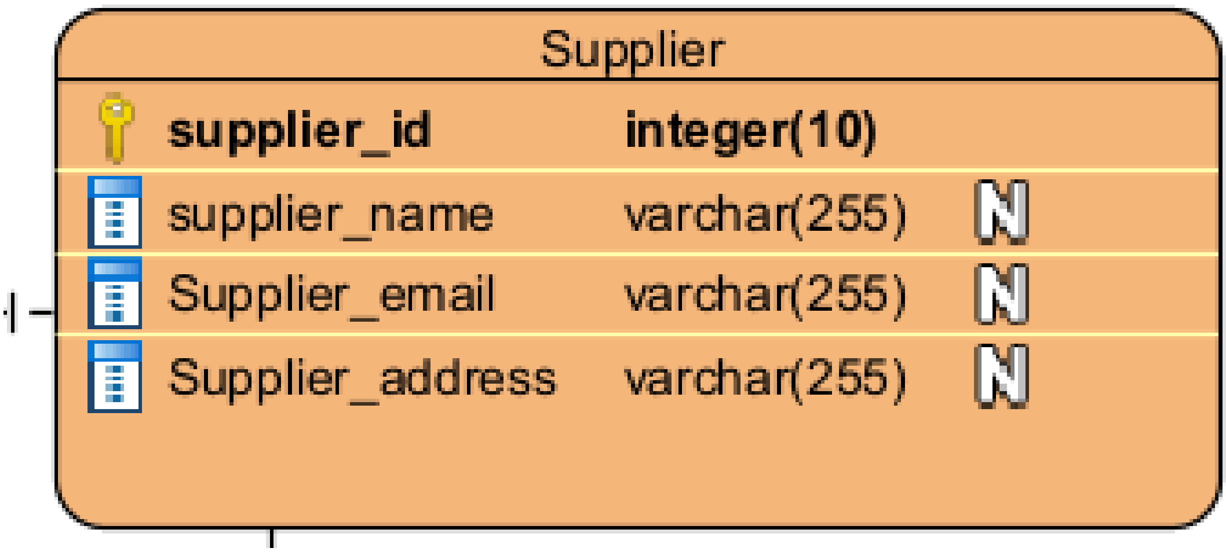

Figure 9.7: PRODUCT

4.Order:
Purpose: Represents customer orders.
Attributes:

- order id: Primary key (unique identifier for each order).
- order date: Date when the order was placed.
- customer id: Foreign key (linked to Customer entity).
- total amount: Total cost of the order.
- Suppliersupplier id : foreign key to supplier

Data Types:order id(numeric),order date(date),customer id(numeric),total amount(numeric).
Relationships:
One customer can place multiple orders (indicated by the line connecting Order and Customer).

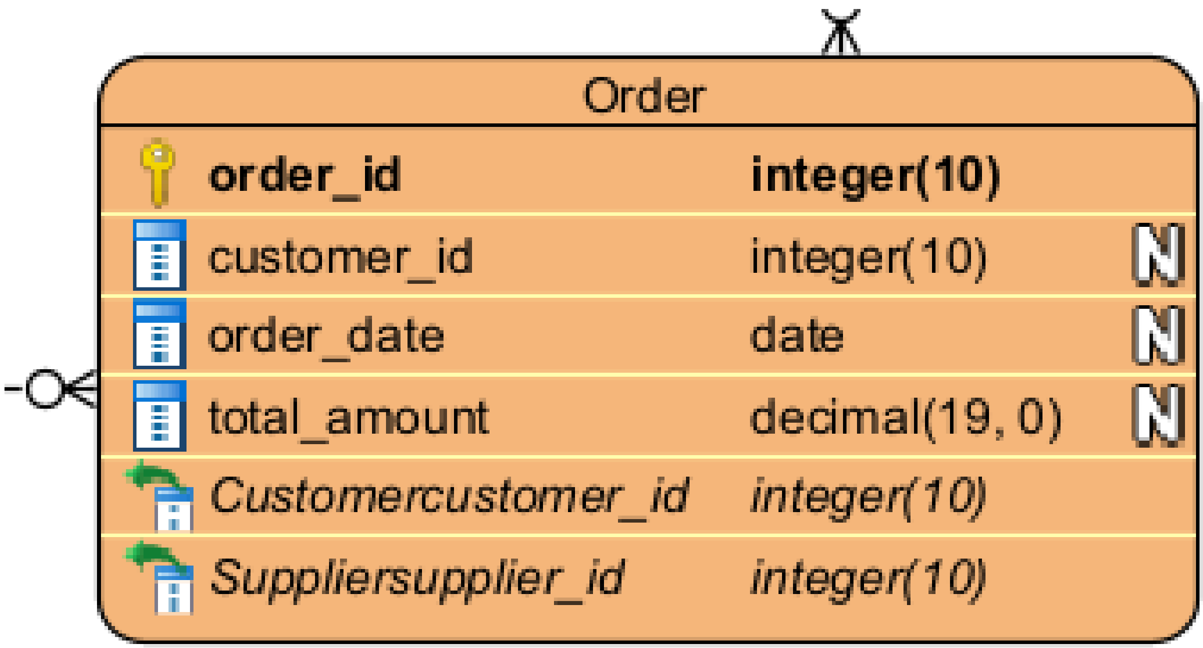

Figure 9.8: PRODUCT

5.Customer:
Purpose: Stores customer-related data.
Attributes:
customer id: Primary key (unique identifier for each customer).

- name: Customer's name.
- email: Customer's email address.
- phone number: Customer's phone number.
- Useruser id: is the foreign key for user id.

Data Types:customer id(numeric),name(text),email(text (text),phone number(text) Useruser id(numeric).

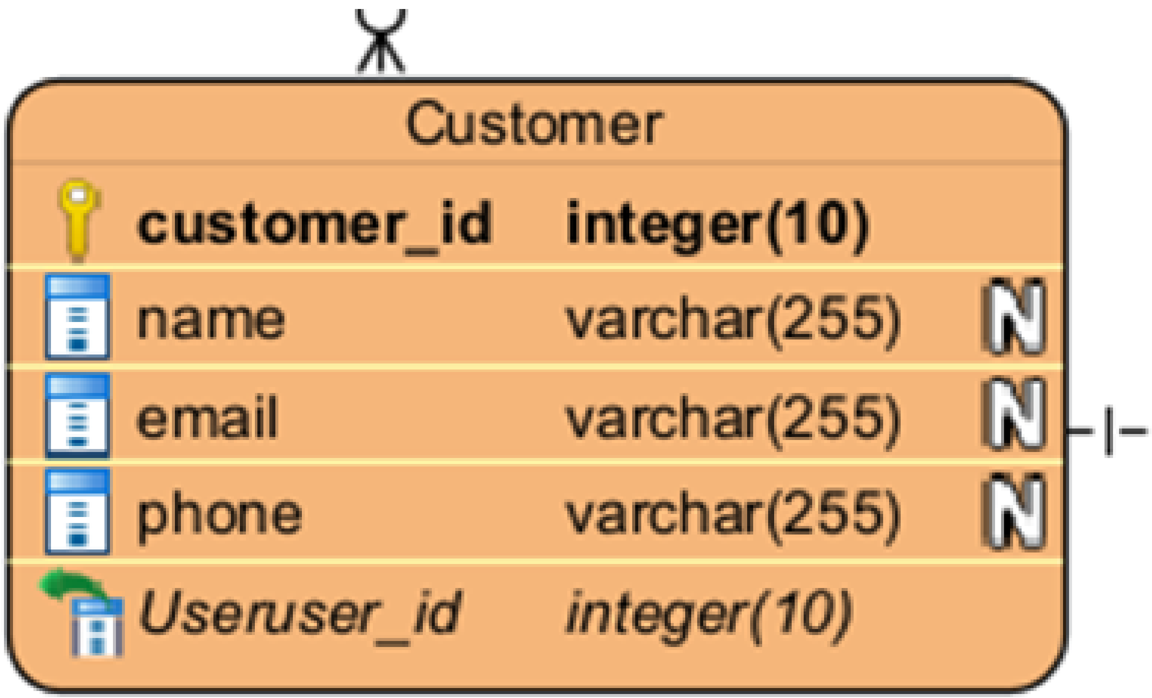

Figure 9.9: PRODUCT

6. User Entity:

Purpose: The User entity represents individuals who interact with the system (administrators, or employees).

Attributes:

- user id: Primary key (unique identifier for each user).
- username :used to login
- password : Securely stored password hash.
- Role : who interact with the system at that time.

Data Types:user id(numeric),name(text),password (text),role (text).

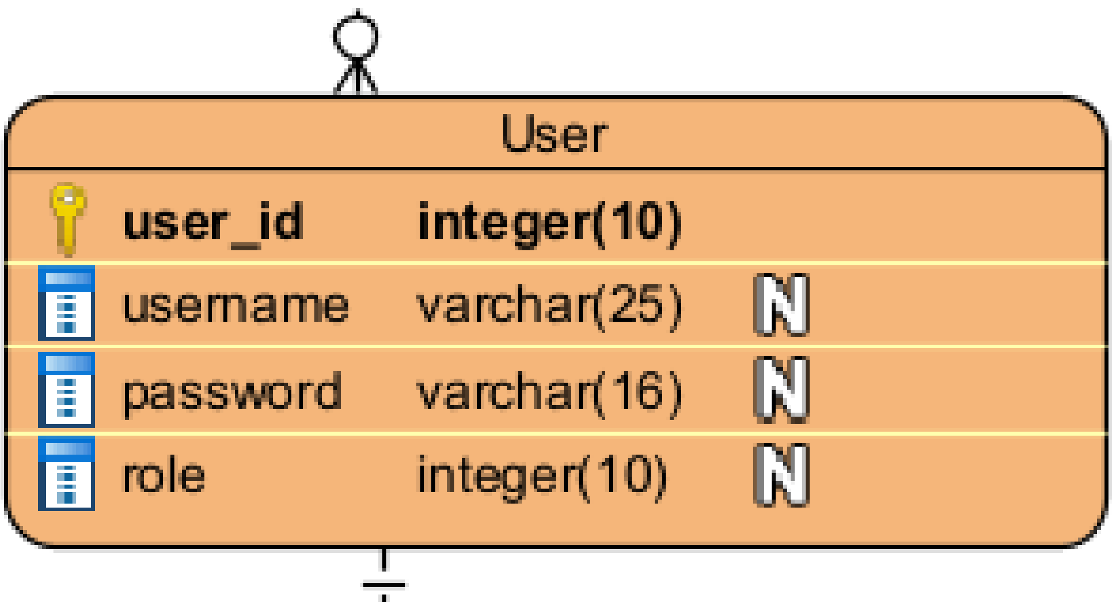

Figure 9.10: PRODUCT