# 



**Technology Store Chain Management**

**Documentation**

|  |  |
| --- | --- |
| **Authors** | **C2309G – Group 1** |
| **Le Trung Kien (Leader)**  **Tran Bao Viet**  **Nguyen Dinh Chien Nguyen Minh Quang Nguyen Thi Yen** |
| **Date** | **12/11/2024** |
| **Instructor** | **To Hoang Anh** |

Hanoi, November 2024

**Index**

[1. Problem Definition 3](#_Toc180989923)

[1.1. Problem Abstraction 4](#_Toc180989924)

[1.2. The Current System 6](#_Toc180989925)

[1.3. The Proposed System 7](#_Toc180989926)

[1.3.1. Boundaries of the System 7](#_Toc180989927)

[1.3.2. Hardware and Software Requirements 8](#_Toc180989928)

[2. Customer Requirements Specification 10](#_Toc180989929)

[2.1. Users of the System 10](#_Toc180989930)

[2.2. System functions 12](#_Toc180989931)

[3. System Designs 17](#_Toc180989932)

[3.1. Entity Relationship Diagram 17](#_Toc180989933)

[3.2. Database Design 18](#_Toc180989934)

[3.3. Sitemap 25](#_Toc180989935)

[3.4. System fucntions design 26](#_Toc180989936)

[4. Task sheet 48](#_Toc180989937)

[5. Validation Checklists 50](#_Toc180989938)

# Problem Definition

* **Meeting the Needs of Businesses in the E-commerce Environment:**
  + The management system for the technology retail chain is designed to provide comprehensive solutions for businesses, ensuring smooth and efficient buying and selling processes.
  + This system supports technology stores in optimizing direct sales processes, including merchandise management, inventory control, payment, and shipping tracking.
  + At the same time, it provides businesses with tools for managing inventory and products, as well as revenue analysis, helping to ensure timely and efficient product supply.
  + By integrating these features, the system helps technology businesses manage more effectively, thereby enhancing business efficiency.
* **Powerful Tools for Businesses in Inventory and Sales Management:**
  + Businesses can easily manage product catalogs, including adding new items, editing product information, or removing discontinued products. This ensures that product information is always up-to-date and accurate.
  + The system automatically updates inventory levels in real-time whenever there are new orders or stock is received, providing sellers with a comprehensive overview of current inventory status.
  + Businesses receive notifications when products are running low or reach a minimum threshold, allowing them to plan for timely restocking and avoid revenue loss due to product shortages.
  + The purchase management feature from suppliers enables businesses to easily track incoming orders, control supply sources, and efficiently manage the storage process.
  + The system provides reports on sales revenue, helping store managers evaluate sales performance and adjust strategies accordingly.
  + Store managers can monitor the work schedules and performance of cashiers, ensuring that personnel are allocated efficiently and effectively.
  + Reports and data analysis from the system assist businesses in making accurate business decisions, optimizing sales processes, and increasing revenue.
* **Supporting Effective Customer Management:**
  + The system stores detailed customer information, including names, phone numbers, and shopping histories, helping sellers understand the needs of specific customer groups.
  + Businesses can interact and engage with customers through integrated communication channels, enhancing service experience.
  + Businesses can easily analyze customer data to gain insights into shopping trends, allowing them to create appropriate promotions or offers for each customer segment.
* **Creating Competitive Advantages for Businesses:**
  + By integrating data analysis tools and detailed sales reports, the system enables businesses to easily assess the effectiveness of advertising campaigns and promotions and adjust marketing strategies to attract more customers.
  + The system supports businesses in analyzing the performance of product lines, helping to decide on new product development strategies or adjust product catalogs to align with market preferences.
  + The ability to manage inventory and supply chains accurately and effectively helps businesses minimize the risk of stockouts and ensures a stable supply, thereby enhancing customer satisfaction.
  + The system provides comprehensive optimization in business operations, from sales processes to customer and inventory management, creating sustainable competitive advantages for businesses in the highly competitive e-commerce market.

## Problem Abstraction

Currently, the process of managing technology sales for many businesses does not effectively utilize information technology, leading to various limitations and challenges in business operations. Most inventory, order, and customer management activities are still carried out manually or through disconnected management software that are not closely integrated. Specifically, the current process can be described as follows:

* **Inventory Management:**
  + Most inventory management is still done manually, through paper records or spreadsheets like Excel. Managers must track remaining stock by checking each product manually
  + Information about goods is not updated in real-time, making it difficult to monitor inventory status.
  + This leads to inaccuracies in capturing the actual quantity of goods, increasing the risk of stock shortages or unnecessary surplus.
  + When products run out of stock, replenishment is not timely due to the lack of alert support tools.
* **Employee Management:**
  + Employee management processes are often carried out manually, lacking a system to track work schedules and performance for each employee.
  + There are no performance evaluation tools, making it difficult to identify and encourage high-performing employees and to address issues with low-performing staff.
  + The lack of information about employee work history and skills hinders task assignment and career development.
* **Store Management:**
  + Stores often lack a comprehensive system to track and manage all business activities, leading to a lack of transparency and difficulties in making management decisions.
  + The absence of tools to monitor store performance and compare with other stores in the chain reduces competitiveness and does not optimize business operations.
* **Customer Management:**
  + Customer information is not stored systematically. Businesses only record customer order information without a system to manage and categorize customers based on value or shopping history.
  + This results in difficulties in customer care, making it impossible to personalize promotions or build long-term relationships with loyal customers.
  + There are no tools to analyze shopping behavior, making it difficult to predict purchasing trends or recommend products to customers.
* **Supplier and Inventory Management:**
  + The process of purchasing from suppliers is carried out in a fragmented manner, lacking a close connection between inventory levels and incoming orders. This leads to a lack of transparency in supply chain management and creates difficulties in controlling incoming goods.
  + There is no system to evaluate supplier performance or tools to find new suppliers, leading to challenges in selecting and maintaining relationships with quality suppliers, which reduces competitiveness.
  + The lack of clear reports on purchasing costs and information about suppliers' price histories makes it difficult for businesses to control budgets and miss opportunities to negotiate better prices.
* **Revenue Management and Reporting:**
  + Revenue reporting is primarily conducted through simple tools like Excel or even manually, making it difficult to aggregate data and analyze business performance.
  + The lack of detailed reporting tools diminishes the ability to evaluate sales performance and to make timely and accurate business decisions.

## The Current System

The proposed new system is a modern desktop application for managing an online technology retail chain, aimed at improving the shopping process and business management. The primary goal is to create a platform that supports businesses in optimizing inventory, order, and revenue management. Key features of the system include:

* **Powerful Inventory Management Tools:**
  + **Real-Time Inventory Tracking:** The system automatically updates inventory status in real-time, providing sellers with a clear view of their stock levels.
  + **Inventory Alerts:** When stock levels are low, the system automatically sends alerts to sellers, enabling them to proactively reorder items and avoid stockouts or excess inventory.
  + **Supplier Order Management:** The system supports tracking incoming shipments, ensuring seamless and accurate updates to inventory.
* **Revenue Management and Business Reporting:**
  + **Real-Time Revenue Tracking:** Sellers can monitor revenue on a daily, weekly, monthly basis, or over any chosen timeframe, with detailed and easy-to-understand reports that highlight business performance.
  + **Detailed Revenue Analysis:** Revenue reports can be customized by product, customer type, geographic area, or promotional campaign, helping sellers easily grasp the situation and devise strategies.
  + **Business Performance Evaluation:** Managers can quickly assess the performance of individual products or product groups, optimizing the inventory mix and adjusting business strategies based on sales trend analysis over time to maximize profits.
* **Comprehensive Customer Information Management:**
  + **Storage and Analysis of Purchase History:** Detailed customer information, including name, address, phone number, and shopping history, is stored, along with the capability to segment customers based on various criteria such as age, gender, or shopping behavior, facilitating easier management and more effective marketing campaigns.
  + **Tracking and Analysis:** The system automatically records transaction histories and analyzes individual customer shopping habits, offering personalized product recommendations, helping sellers capture demand and improve conversion rates, while enhancing the shopping experience.
* **High Security and Performance:**
  + **Transaction and Customer Information Security:** The system employs modern data encryption methods and multi-factor authentication, ensuring that personal and payment information of customers is kept absolutely secure.
  + **High Performance and Scalability:** The system is built to handle multiple transactions simultaneously with fast page loading speeds, ensuring a smooth experience even during peak hours, and is scalable to accommodate future business growth.
* **Integration and Extensibility:**
  + **Integration with Other Platforms:** The system can integrate with shipping services, CRM systems, or other payment platforms, optimizing the process from purchasing to delivery and customer management.

## The Proposed System

## Boundaries of the System

The project for managing a chain of technology retail stores aims to build an integrated system to manage and optimize the business processes of small and medium-sized tech stores. This system will include several basic functions such as inventory management, revenue tracking, stock control, and customer support.

1. **Primary Users:**
   1. **General Director:** A high-level decision-maker with access to reports and performance analysis data for the retail chain.
   2. **Store Management:** Manages daily operations and personnel of the store, including product management and inventory control.
   3. **Warehouse Management:** Oversees and tracks inventory, coordinating the inbound/outbound processes.
   4. **Cashier:** Responsible for payment processing, invoice management, and daily sales revenue tracking
2. **Scope of the System:**
   1. **In-Store Sales Management:**
      * Automatically updates inventory in real-time when sales or returns occur, helping managers accurately control stock levels.
      * The system supports sales transactions and payment processing, including invoice management and promotional programs.
   2. **Inventory Management:**
      * The system helps track the quantity of goods in stock, plan restocking, and alert when inventory levels are low.
      * Assigns permissions to warehouse managers to update product statuses and handle internal orders within the retail chain.
   3. **Chain Store Management:**
      * Integrates reports on revenue, stock levels, and sales performance for each store, providing real-time analytical information to leadership.
      * The system will store detailed information about stores, such as sales history, monthly sales figures, and best-selling products to support decision-making.
   4. **User Management and Role Assignment:**
      * The system will assign roles to users to secure and protect internal information. Each role will have access only to appropriate functions, ensuring that workflows are optimized and secure.

## Hardware and Software Requirements

* + - * **Programming Languages**: Java and JavaFX
        + **Java:** Java is a powerful object-oriented programming language with high compatibility across platforms (cross-platform). Java provides security, performance, and high scalability, making it suitable for building mid-sized and large e-commerce systems. It also has a large development community, which makes it easy to find resources and support when needed.

**Application in the project:** Java will be used to build the entire server-side logic, including data management, order processing, and control of functions related to products, customers, and transactions.

* + - * + **JavaFX:** This is a powerful framework for developing user interfaces (UI) using Java. JavaFX allows for the creation of visually appealing graphics, good interactivity, and easy customization according to user requirements. It supports various UI components such as tables, input forms, menus, and other graphical elements, enhancing the user experience.

**Application in the project:** JavaFX will be used to develop the management interface for admins and sales staff, allowing them to manage products, orders, and customers in an intuitive and straightforward manner. This user-friendly interface will also help end users (customers) easily search for products, make purchases, and complete payments.

* + - * **Database:** SQL
        + **SQL:** SQL is a popular database management system, suitable for small and medium-sized enterprises, helping to organize data efficiently and facilitate easy querying.

**Application in the project:** SQL will be used to store data for products, customers, orders, and inventory. The data tables will be designed with clear relationships, supporting quick querying and report generation.

* + - * **Framework**: JDBC (Java Database Connectivity)
        + **JDBC:** This is the standard API for Java used to connect and interact with database management systems. JDBC allows Java to interact directly with MySQL or PostgreSQL, including operations such as adding, updating, deleting, and querying data.

**Application in the project:** JDBC will be used to connect the Java application with the database. When users perform actions such as placing orders, adding products to the cart, or managing inventory, JDBC will help update information in the database in real time. This ensures data consistency and accuracy, which is especially important when the system needs to handle multiple simultaneous transactions.

* + - * **CSS** (Cascading Style Sheets)
        + **CSS:** CSS is a language used to format and present the appearance of web pages. It helps manage colors, layouts, fonts, and other interface elements, creating a user-friendly (UI) and visually appealing interface.

**Application in the project:** CSS will be used to design and format the user interface in JavaFX, optimizing the interface Tech store desktop app for user-friendliness on both desktop and mobile devices. This ensures the system not only functions well but also creates a smooth, intuitive, and easy-to-use shopping experience.

* + - * **Development Tools**: IntelliJ IDEA & MySQL Workbench
        + **IntelliJ IDEA:** IntelliJ IDEA is a powerful IDE (Integrated Development Environment) that fully supports features such as code completion, real-time error checking, and easy integration with development tools like Maven, Git, and database management systems. IntelliJ IDEA also has a user-friendly interface, helping programmers increase productivity and manage source code efficiently. With its intelligent features and ability to integrate various tools, IntelliJ IDEA will be the ideal IDE for the store management project.

**Application in the project:** IntelliJ IDEA will be used to develop the entire project, from backend programming (server) to user interface design and database integration. Supporting features like error checking and code suggestions will help optimize the development process.

* + - * + **MySQL Workbench:** MySQL Workbench is a visual tool for designing, managing, and modeling databases, making it easy for administrators to interact with MySQL databases.

**Application in the project:** MySQL Workbench will assist administrators in designing and managing the database, from creating tables and relationships to checking integrity and executing data queries.

# Customer Requirements Specification

## Users of the System

In the context of the tech store management system, the system will serve four main user groups: Cashier, Store Management, Warehouse Management, and General Director. Below are the detailed requirements for each group:

1. **Cashier:** Cashiers play a crucial role in the store's sales operations, ensuring transactions proceed smoothly and accurately. The main tasks of cashiers include:
   * **Processing Sales Transactions:** Executing payment transactions for customers, including scanning product codes and entering payment information.
   * **Managing Invoices:** Issuing invoices to customers and handling return or exchange requests.
   * **Price Checking:** Searching for and verifying product prices in the system to ensure accuracy.
   * **Maintaining Cash Fund:** Managing cash in the register, ensuring that the amount matches the recorded figures at the end of the shift.
2. **Store Management:** Store managers play a vital role in maintaining efficient operations and optimizing revenue. The system needs to provide the necessary tools for them to manage inventory, track employee performance, and analyze revenue effectively.
   * **Monitoring Daily Operations:** Ensuring the store operates smoothly, from opening, serving customers, to closing.
   * **Managing Inventory:** Tracking the quantity of goods in stock, placing new orders as necessary, and ensuring products are always available.
   * **Tracking Employee Performance:** Evaluating employee performance, scheduling work hours, and managing personnel-related issues.
   * **Analyzing Sales Reports:** Reviewing revenue reports over time, identifying customer shopping trends, and adjusting sales strategies accordingly.
3. **Warehouse Management:** Warehouse managers are responsible for storing and supplying goods to the store. The main tasks include:
   * **Managing Inbound and Inventory:** Updating inventory status, receiving new shipments, controlling outbound goods, and ensuring products are stored correctly.
   * **Monitoring Inventory Levels:** Tracking product quantities, identifying low stock to support timely reordering, and avoiding shortages.
   * **Managing Purchase Orders from Suppliers:** Tracking purchase orders from suppliers, checking the quality of received goods, and handling arising issues.
   * **Conducting Periodic Inventory Checks:** Regularly auditing inventory to verify actual stock quantities and minimize losses.
4. **General Director:** The General Director has overall access to the entire system to support strategic decision-making. The main tasks include:
   * **Reviewing Overview Reports:** Monitoring revenue reports, sales performance, and analyzing business trends to adjust strategies.
   * **Making Strategic Decisions:** Using data from the system to develop business expansion strategies, manage costs, and optimize operational processes.
   * **Managing Supplier Relationships:** Approving orders with suppliers, reviewing pricing and product quality to maintain good relationships with the best suppliers.
   * **Monitoring Store Performance:** Tracking the operational effectiveness of each store in the chain to evaluate and make necessary adjustments.

## System functions

* + - **Login**
      * Input Information:
        + Username
        + Password
      * Output Information:
        + Success or failure authentication message
        + Redirect to the dashboard if authentication is successful
      * Processing Method:
        + Validate the username and password.
        + Connect to the database to verify the credentials.
        + If the information is correct, store the user's login state.
      * Data to be Stored:
        + Username and password (with password hashed) in the accounts table.
    - **Search**
      * Input Information:
        + Search keyword
      * Output Information:
        + List of matching entities
      * Processing Method:
        + Filter data from the database based on the keyword.
        + Display search results to the user.
      * Data to be Stored:
        + No data storage needed, only retrieval from relevant tables.
    - **Filter product**
      * Input Information:
        + Filtering criteria
      * Output Information:
        + List of filtered entities
      * Processing Method:
        + Execute an SQL query with conditions based on the selected criteria.
        + Display the filtered results to the user.
      * Data to be Stored:
        + No data storage needed, only retrieval from relevant tables.
    - **Save Customer, Invoice**
      * Input Information:
        + Customer information
        + Invoice information
      * Output Information:
        + Confirmation message of successful or failed save operation
      * Processing Method:
        + Validate the input information.
        + Save customer information in the customers table.
        + Save invoice information in the receipts table along with invoice details in the products\_receipt table.
      * Data to be Stored:
        + Customer information in the customers table.
        + Invoice information in the receipts table and invoice details in the products\_receipt table.
    - **Dashboard**
      * Input Information:
        + None (data fetched automatically from the database)
      * Output Information:
        + Statistical information such as revenue, number of orders, best-selling products, number of customers
      * Processing Method:
        + Execute SQL queries to compute and retrieve statistical data.
        + Display statistical information on the dashboard interface.
      * Data to be Stored:
        + No data storage needed, only retrieval from relevant tables.
    - **CRUD for Employee, Product, Stores, Warehouse, Account**
      * Input Information:
        + Necessary information for each entity (employee, product, store, warehouse, account)
      * Output Information:
        + Confirmation message of successful or failed CRUD (Create, Read, Update, Delete) operation
      * Processing Method:
        + Create, read, update, or delete an entity in the database.
        + Validate the information before performing the operation.
      * Data to be Stored:
        + Corresponding information in the employees, products, stores, warehouses, and accounts tables.
    - **Create Account**
      * Input Information:
        + ComboBox to select employee (list of employees)
        + Input field for username
        + Input field for password
        + Input field for confirm password
      * Output Information:
        + Confirmation message indicating success or failure of account creation
        + Any error messages if validation fails (e.g., username already exists, passwords do not match)
      * Processing Method:
        + Validate Input:

Ensure the username is not empty and adheres to any specific format requirements.

Check if the username already exists in the accounts table.

Validate that the password meets security criteria (e.g., length, complexity).

Confirm that the password matches the confirmation password.

* + - * + Store Information:

If validation passes, create a new account entry in the accounts table, linking it to the selected employee from the ComboBox.

Hash the password before storing it for security purposes.

* + - * + Feedback:

Display a success message or any relevant error messages based on the validation checks.

* + - * Data to be Stored:
        + Employee ID (from ComboBox) in the accounts table.
        + Username (if unique) and hashed password.
    - **Edit Account**
      * Input Information:
        + ComboBox to select employee (pre-populated with existing employees)
        + Input field for username
        + Input field for old password
        + Input field for new password
      * Output Information:
        + Confirmation message indicating success or failure of account update
        + Any error messages if validation fails (e.g., old password incorrect, username already exists)
      * Processing Method:
        + Validate Input:

Ensure the username is not empty and adheres to any specific format requirements.

Verify that the old password is correct by comparing it to the stored hashed password in the accounts table.

Validate that the new password meets security criteria.

Check if the new username already exists (if the username is being changed).

* + - * + Update Information:

If validation passes, update the username (if changed) and the password (hashing the new password) in the accounts table.

* + - * + Feedback:

Display a success message or any relevant error messages based on the validation checks.

* + - * Data to be Stored:
        + Updated employee ID (if necessary) in the accounts table.
        + Updated username and hashed new password.
    - **Logout**
      * Input Information:
        + None (logout action)
      * Output Information:
        + Redirect to the login page
      * Processing Method:
        + Clear the user’s login state.
        + Redirect to the login interface.
      * Data to be Stored:
        + No data storage needed, only changes in the login state.
    - **Create Purchase Invoice**
      * Input Information:
        + ComboBox to select warehouse or store
        + List products
        + Input field for quantity
        + Input field for price per unit
        + Date picker for invoice date
      * Output Information:
        + Confirmation message indicating success or failure of invoice creation
        + Any error messages if validation fails
      * Processing Method:
        + Validate Input:

Ensure that the selected warehouse/store is valid.

Validate that the selected product exists in the database.

Check that the quantity is a positive integer.

Ensure the price per unit is a positive number.

Validate the invoice date to ensure it is not a future date.

* + - * + Store Information:

If validation passes, create an entry in the import\_warehouse or import\_store table (depending on where the products are being imported).

Also, create corresponding entries in the import\_warehouse\_details or import\_store\_details table, including product ID, quantity, and price.

* + - * + Feedback:

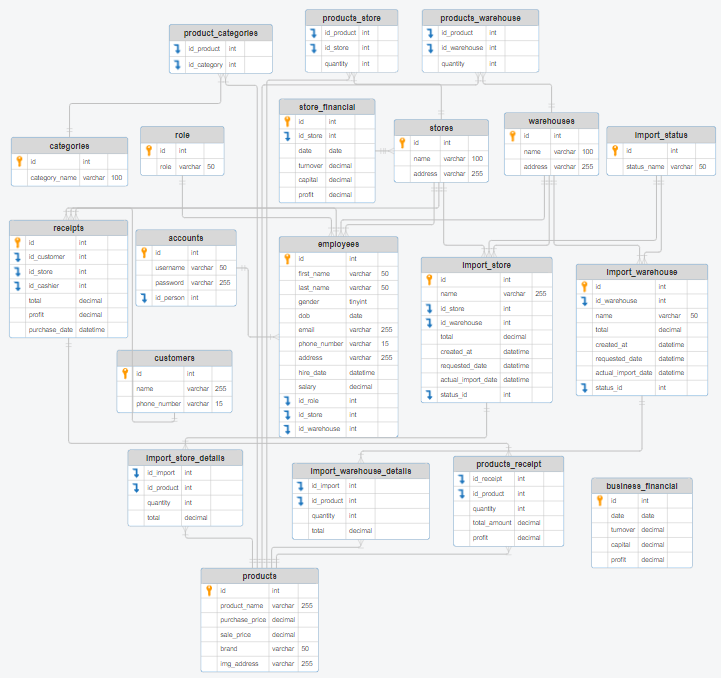
Display a success message indicating the invoice has been created or any relevant error messages based on the validation checks.

**Validation Considerations:**

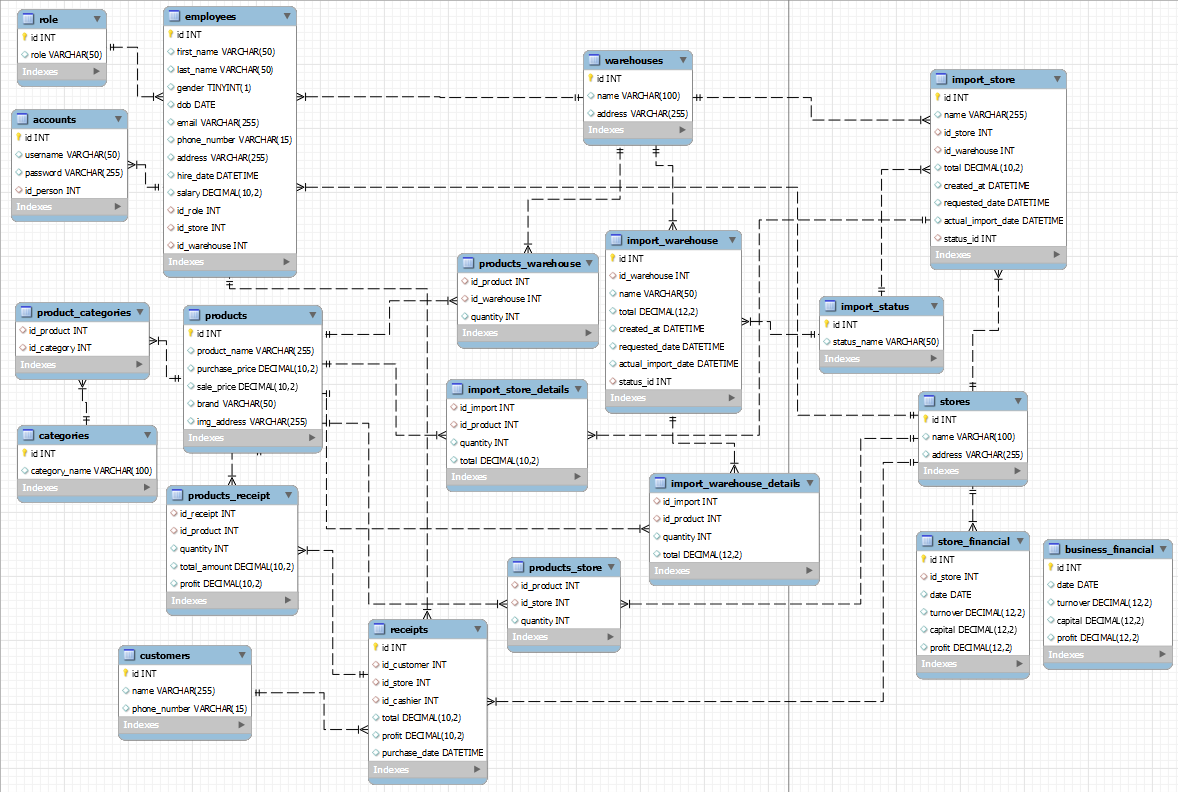
1. Username Validation:
   * + Must not be empty.
     + Must follow specified format
     + Must be unique in the accounts table.
2. Password Validation:
   * + Minimum length requirement
     + Must include a mix of uppercase letters, lowercase letters, numbers, and special characters (if required).
     + New password must differ from the old password during edits.
3. Confirmation Validation:
   * + The confirm password field must match the password field in the creation process.

# System Designs

## Entity Relationship Diagram



## Database Design

* + - ***Database Diagram***
    - ***Description of the detailed tables*** 
      * ***products***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment Primary key |
| product\_name | Varchar (255) |  |
| purchase\_price | Decimal (10,2) |  |
| sale\_price | Decimal (10,2) |  |
| brand | Varchar (50) |  |
| img\_address | Varchar (255) |  |

* + - * ***categories***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment Primary key |
| category\_name | Varchar (100) |  |

* + - * ***import\_status***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment Primary key |
| import\_name | Varchar (50) |  |

* + - * ***product\_categories***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id\_product | Int | Foreign Key  produucts (id) |
| id\_category | Int | Foreign Key  categories (id) |

* + - * ***warehouse***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment Primary key |
| name | Varchar (100) |  |
| address | Varchar (255) |  |

* + - * ***stores***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment Primary key |
| name | Varchar (100) |  |
| address | Varchar (255) |  |

* + - * ***Customers***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment Primary key |
| name | Varchar (100) |  |
| phone\_number | Varchar (255) |  |

* + - * ***products\_warehouse***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| Id\_product | int | Foreign Key (products.id) |
| Id\_warehouse | int | Foreign Key (warehouses.id) |
| quantity | int |  |

* + - * ***import\_warehouse***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | int | Auto increment, Primary key |
| id\_warehouse | int | Foreign Key (warehouses.id) |
| name | Varchar (50) |  |
| total | Decimal (12, 2) |  |
| created\_at | datetime |  |
| requested\_date | datetime |  |
| actual\_import\_date | datetime |  |
| status\_id | int | Foreign Key (import\_status.id) |

* + - * ***import\_warehouse\_details***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id\_import | int | Foreign Key (import\_warehouse.id) |
| id\_product | int | Foreign Key (products.id) |
| quantity | int |  |
| total | Decimal (12, 2) |  |

* + - * ***store\_financial***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | int | Auto increment, Primary key |
| id\_store | int | Foreign Key (stores.id) |
| date | date |  |
| turnover | Decimal (12, 2) |  |
| capital | Decimal (12, 2) |  |
| profit | Decimal (12, 2) |  |

* + - * ***product\_store***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id\_product | int | Foreign Key (products.id) |
| id\_store | int | Foreign Key (stores.id) |
| quantity | int |  |

* + - * ***import\_store***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | int | Auto increment, Primary key |
| id\_store | int | Foreign Key (stores.id) |
| id\_warehouse | int | Foreign Key (warehouses.id) |
| total | Decimal (10, 2) |  |
| created\_at | datetime |  |
| requested\_date | datetime |  |
| actual\_import\_date | datetime |  |
| status\_id | int | Foreign Key (import\_status.id) |

* + - * ***import\_store\_details***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id\_import | int | Foreign Key (import\_store.id) |
| id\_product | int | Foreign Key (products.id) |
| quantity | int |  |
| total | Decimal (10, 2) |  |

* + - * ***role***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | int | Auto increment, Primary key |
| role | varchar(50) |  |

* + - * ***employees***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | int | Auto increment, Primary key |
| first\_name | Varchar (50) |  |
| last\_name | Varchar (50) |  |
| gender | Boolean |  |
| dob | Date |  |
| email | Varchar (255) |  |
| phone\_number | Varchar (15) |  |
| address | Varchar (255) |  |
| hire\_date | datetime |  |
| salary | Decimal (10, 2) |  |
| id\_role | Int | Foreign Key (role.id) |
| id\_store | Int | Foreign Key (stores.id) |
| id\_warehouse | Int | Foreign Key (warehouses.id) |

* + - * ***account***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| Id | int | Auto increment, Primary key |
| username | Varchar (50) |  |
| password | Varchar (255) |  |
| id\_person | Int | Foreign Key (employees.id) |

* + - * ***receipts***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | Int | Auto increment, Primary key |
| id\_customer | Int | Foreign Key (customers.id) |
| id\_store | Int | Foreign Key (stores.id) |
| id\_cashier | Int | Foreign Key (employees.id) |
| total | Decimal (10, 2) |  |
| profit | Decimal (10, 2) |  |
| datetime | Datetime |  |

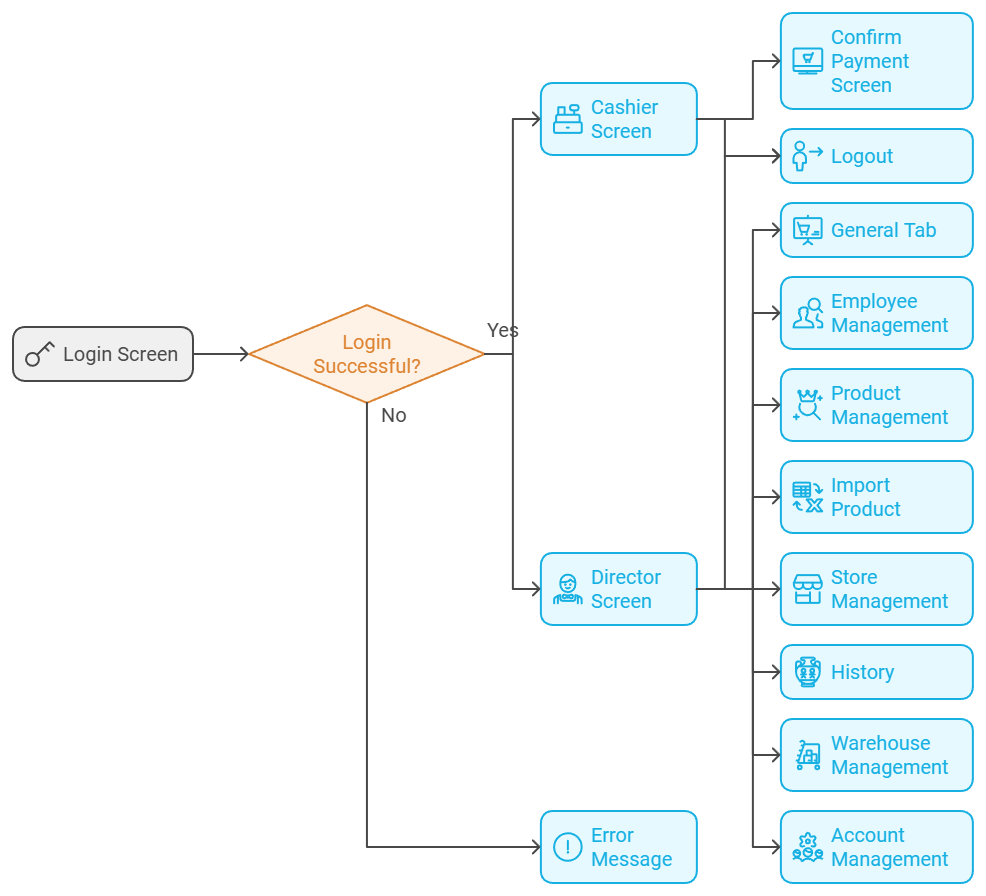
* + - * ***products\_receipt***

|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id\_receipt | int | Foreign Key (receipts.id) |
| id\_product | int | Foreign Key (products.id) |
| quantity | int |  |
| total\_amount | Decimal (10, 2) |  |
| profit | Decimal (10, 2) |  |

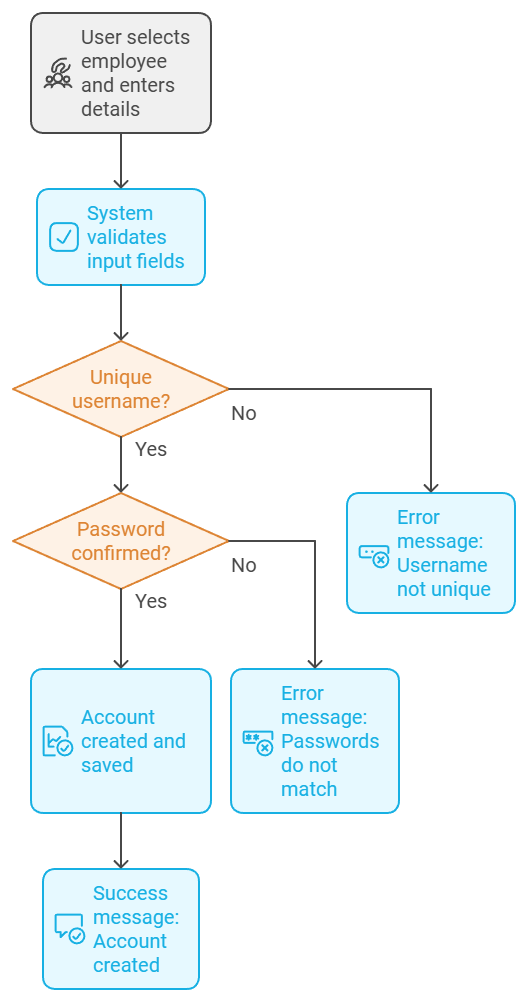
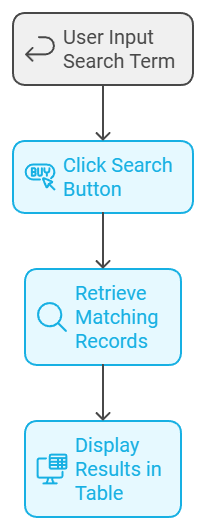
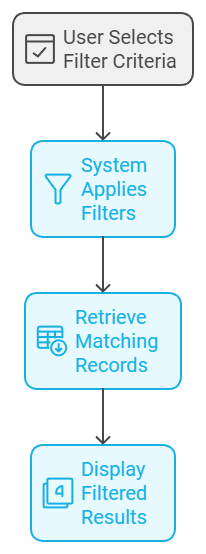
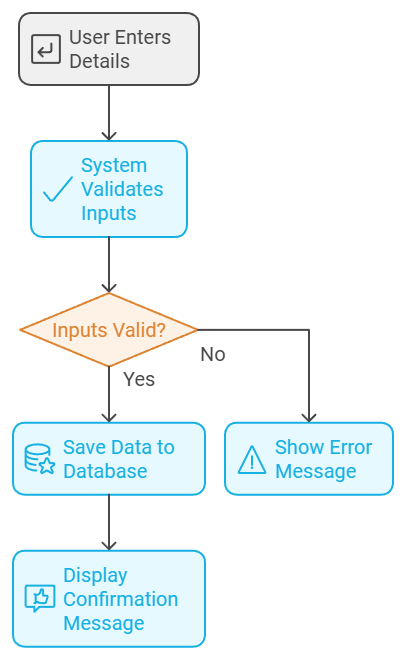
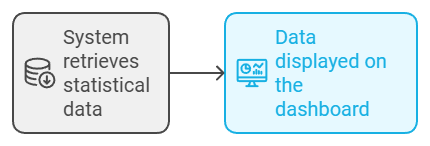
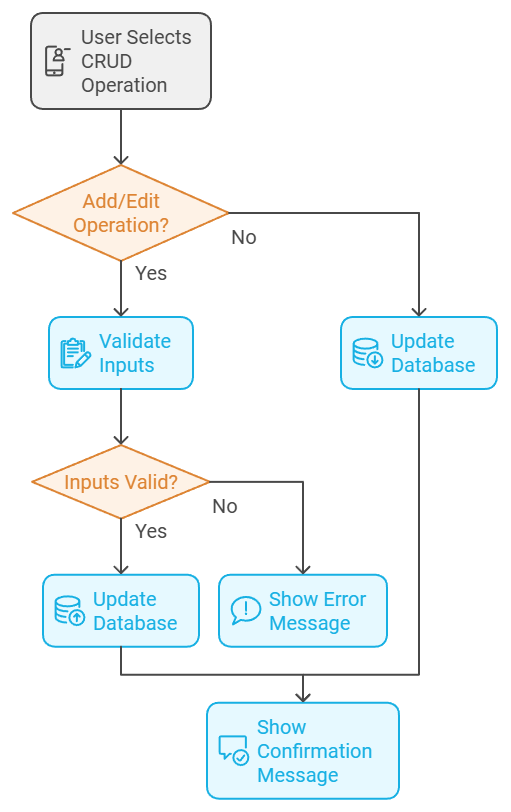
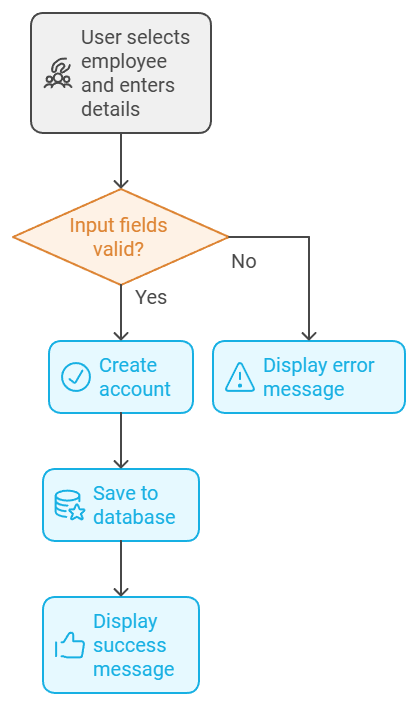
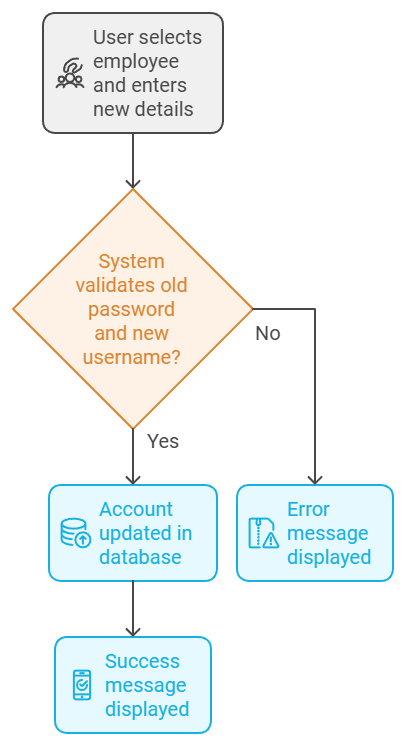
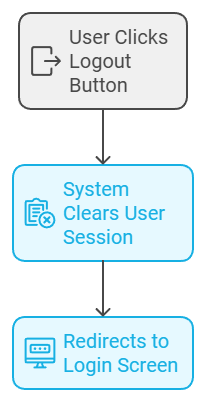
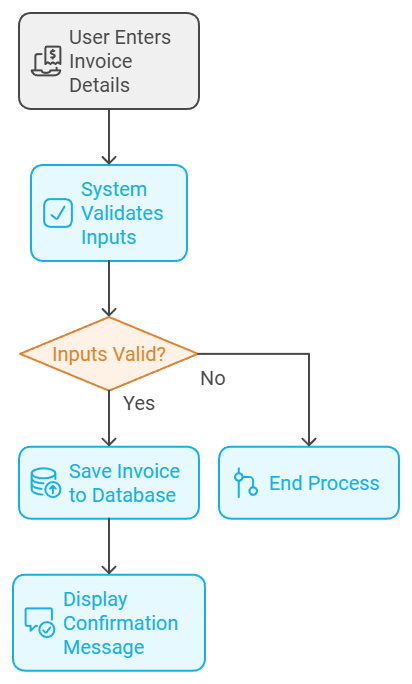
* + - * ***business\_financial***

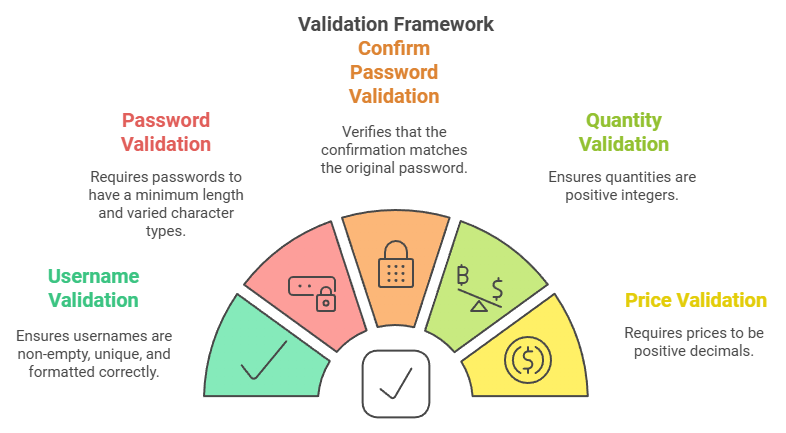
|  |  |  |
| --- | --- | --- |
| **Column** | **Data type** | **Constraint** |
| id | int | Auto increment, Primary key |
| date | date |  |
| turnover | Decimal (12, 2) |  |
| capital | Decimal (12, 2) |  |
| profit | Decimal (12, 2) |  |

## Sitemap



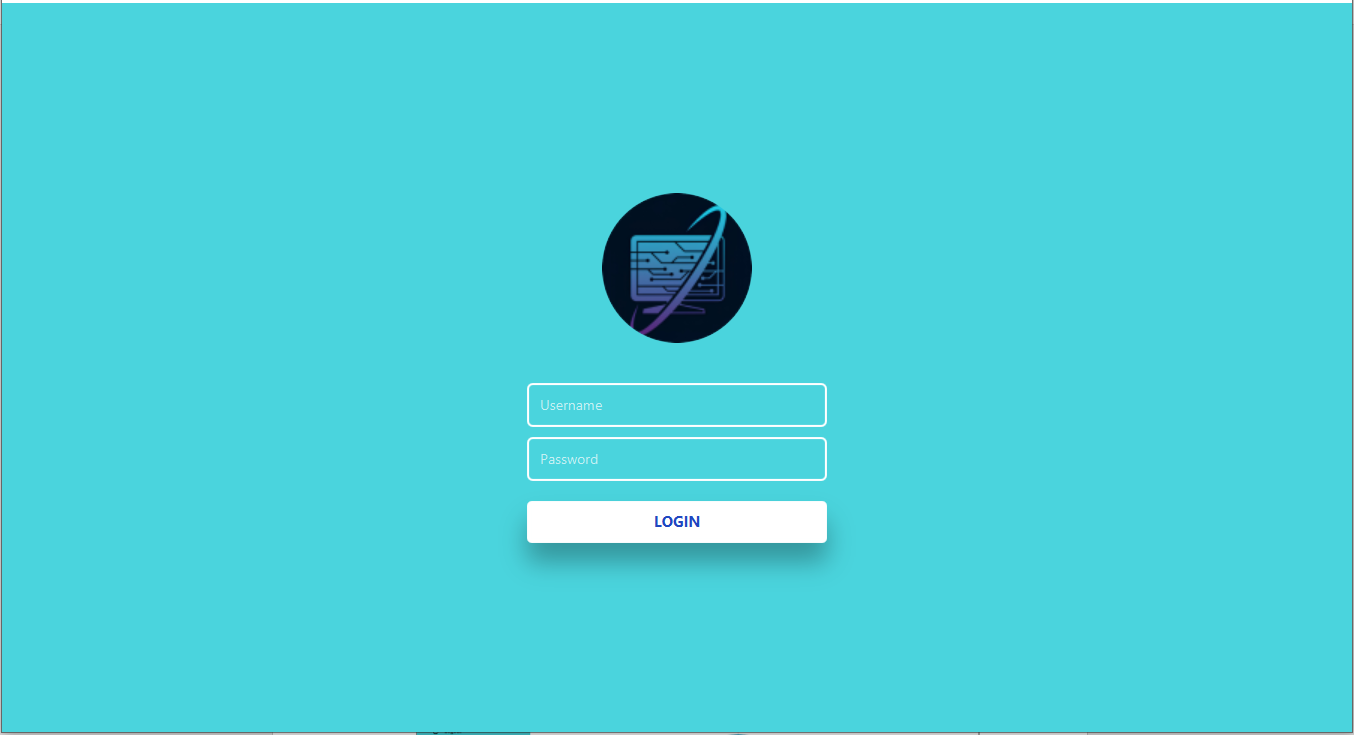
## System fucntions design

* + - **Login**
      * Input Interface:
        + Fields: Username (text field), Password (password field)
        + Buttons: Login
      * Output Interface:
        + Messages: "Login Successful," "Invalid Username or Password"
        + Redirect to Dashboard upon successful login
      * Algorithm (Text):
        + User enters their username and password.
        + System verifies credentials with the database.
        + If valid, the user is directed to the Dashboard; otherwise, an error message is displayed.
      * Flowchart:  
        
    - **Search**
      * Input Interface:
        + Field: Search (text field for keyword input)
        + Button: Search
      * Output Interface:
        + Table/List of matching records (e.g., products, employees)
      * Algorithm (Text):
        + User enters a search term.
        + System retrieves matching records from the database.
        + Display results in a table.
      * Flowchart:  
        
    - **Filter product**
      * Input Interface:
        + checkboxes for filtering criteria
      * Output Interface:
        + Filtered list of products
      * Algorithm (Text):
        + User selects filter criteria.
        + System applies filters and retrieves matching records from the database.
        + Display filtered results.
      * Flowchart  
        
    - **Save Customer, Invoice**
      * Input Interface:
        + Fields for Customer (name, contact info) and Invoice (product, quantity, price)
      * Output Interface:
        + Confirmation message for successful saving of customer and invoice
      * Algorithm (Text):
        + User enters customer and invoice details.
        + System validates inputs.
        + If valid, save data to database.
      * Flowchart  
        
    - **Dashboard**
      * Input Interface:
        + None (dashboard data loads automatically)
      * Output Interface:
        + Display of various metrics (e.g., revenue, best-selling products)
      * Algorithm (Text):
        + System retrieves statistical data from the database.
        + Data is displayed on the dashboard.
      * Flowchart:  
        
    - **CRUD for Employee, Product, Stores, Warehouse, Account**
      * Input Interface:
        + Input fields for each entity’s details
        + Buttons: Add, Edit, Delete, View
      * Output Interface:
        + Confirmation messages for each operation
      * Algorithm (Text):
        + User selects a CRUD operation.
        + System validates inputs (for add/edit).
        + Database is updated based on the operation.
      * Flowchart:  
        
    - **Create Account**
      * Input Interface:
        + Fields: Employee selection (ComboBox), Username, Password, Confirm Password
      * Output Interface:
        + Success or error messages based on validation
      * Algorithm (Text):
        + User selects employee and enters account details.
        + System validates input fields (unique username, password confirmation).
        + If valid, the account is created and saved to the database.
      * Flowchart:  
        
    - **Edit Account**
      * Input Interface:
        + Fields: Employee selection (ComboBox), Username, Old Password, New Password
      * Output Interface:
        + Success or error messages
      * Algorithm (Text):
        + User selects employee and enters new account details.
        + System validates old password and new username.
        + If valid, the account is updated in the database.
      * Flowchart:  
        
    - **Logout**
      * Input Interface:
        + Button: Logout
      * Output Interface:
        + Redirect to the login screen
      * Algorithm (Text):
        + User clicks Logout.
        + System clears user session and redirects to the login screen.
      * Flowchart:  
        
    - **Create Purchase Invoice**
      * Input Interface:
        + Fields for warehouse/store selection, product, quantity, price, date
      * Output Interface:
        + Confirmation message for invoice creation
      * Algorithm (Text):
        + User enters purchase invoice details.
        + System validates inputs (e.g., positive quantity, valid date).
        + If valid, the invoice is saved to the database.
      * Flowchart:  
        
    - **Validation Considerations**
      * Username Validation:
        + Non-empty, unique, and follows the required format (e.g., alphanumeric)
      * Password Validation:
        + Minimum length, combination of character types
      * Confirm Password Validation:
        + Must match the initial password entry
      * Quantity and Price Validation (for Invoices):
        + Positive integer for quantity, positive decimal for price

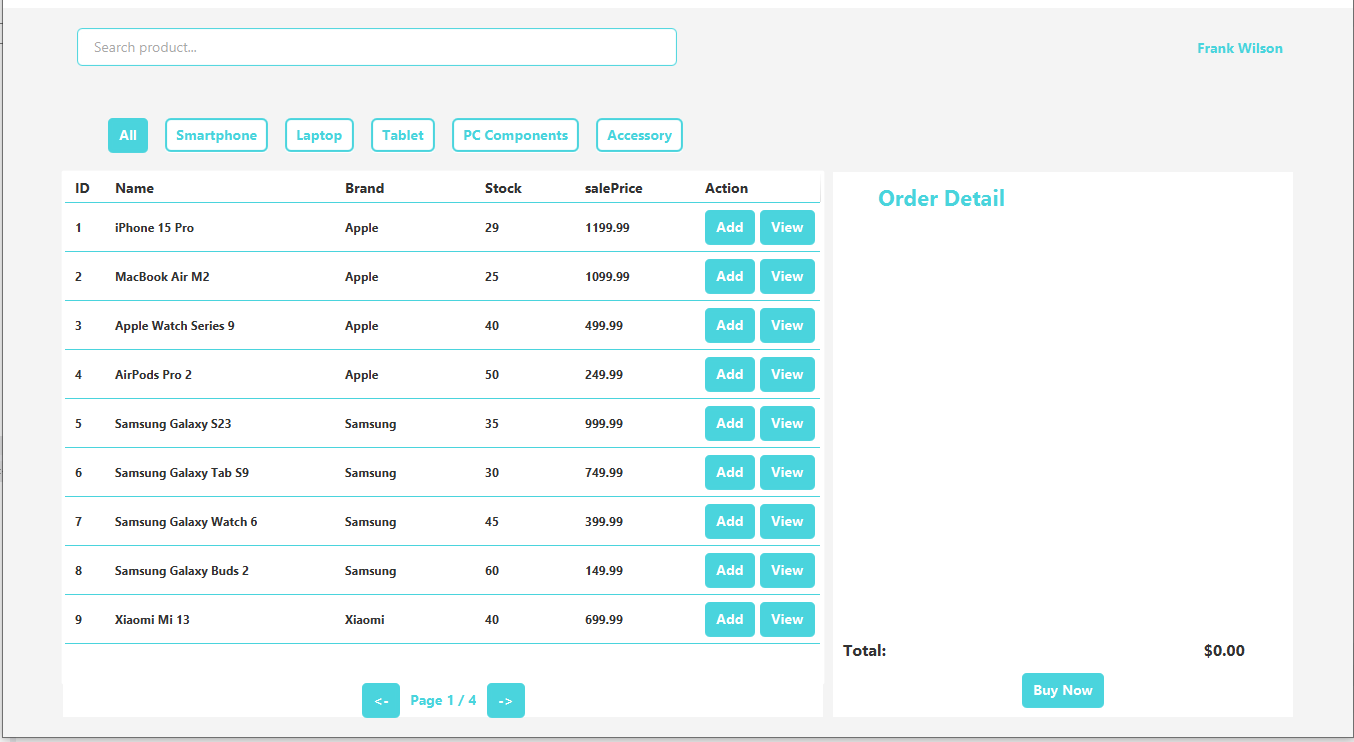


## Screen

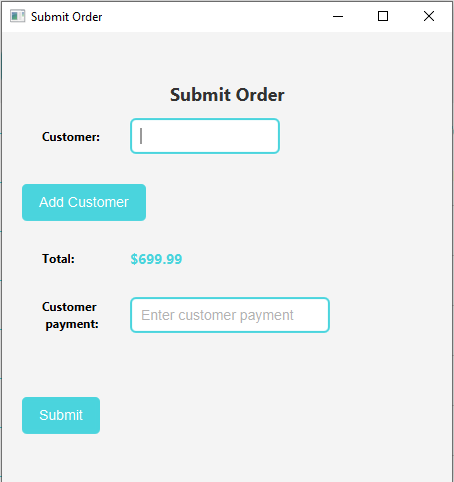
* + - **Login**

****

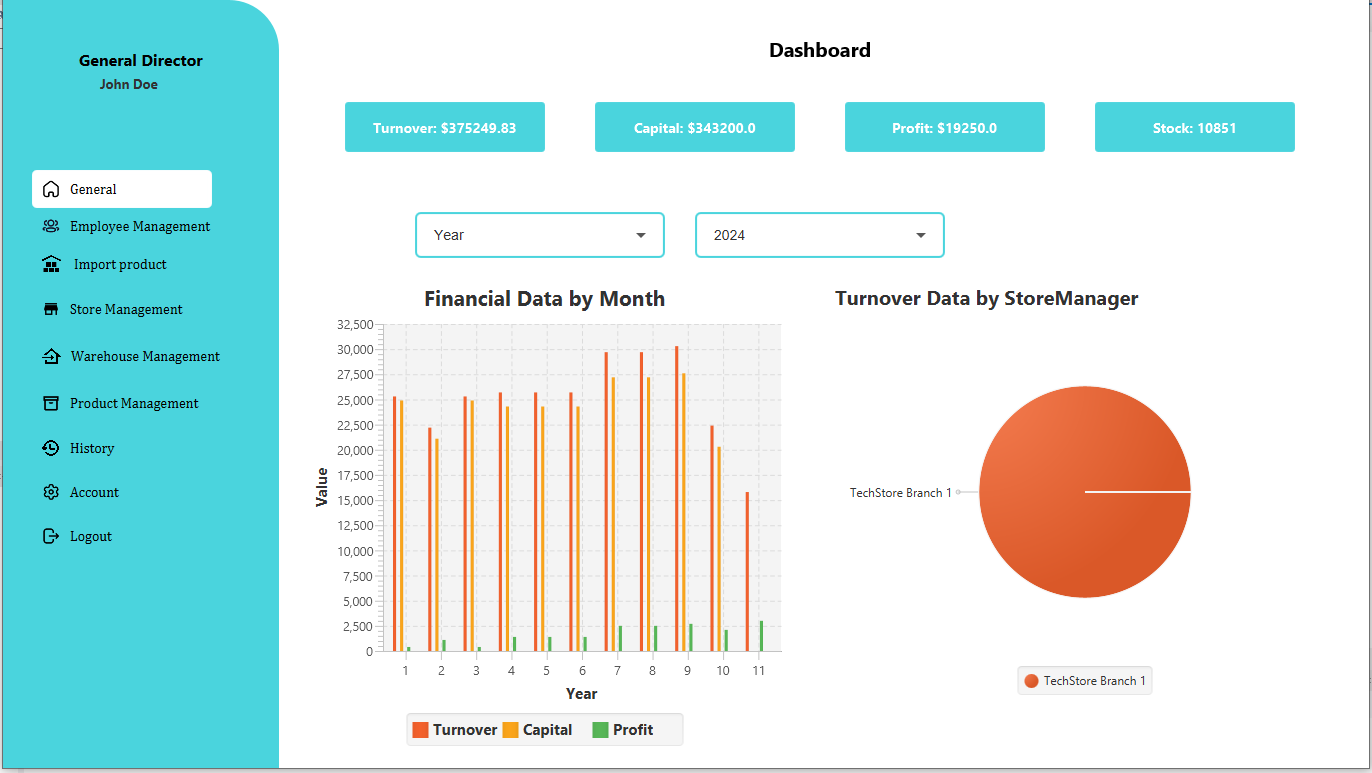
* + - **Cashier**
      * **Main**

****

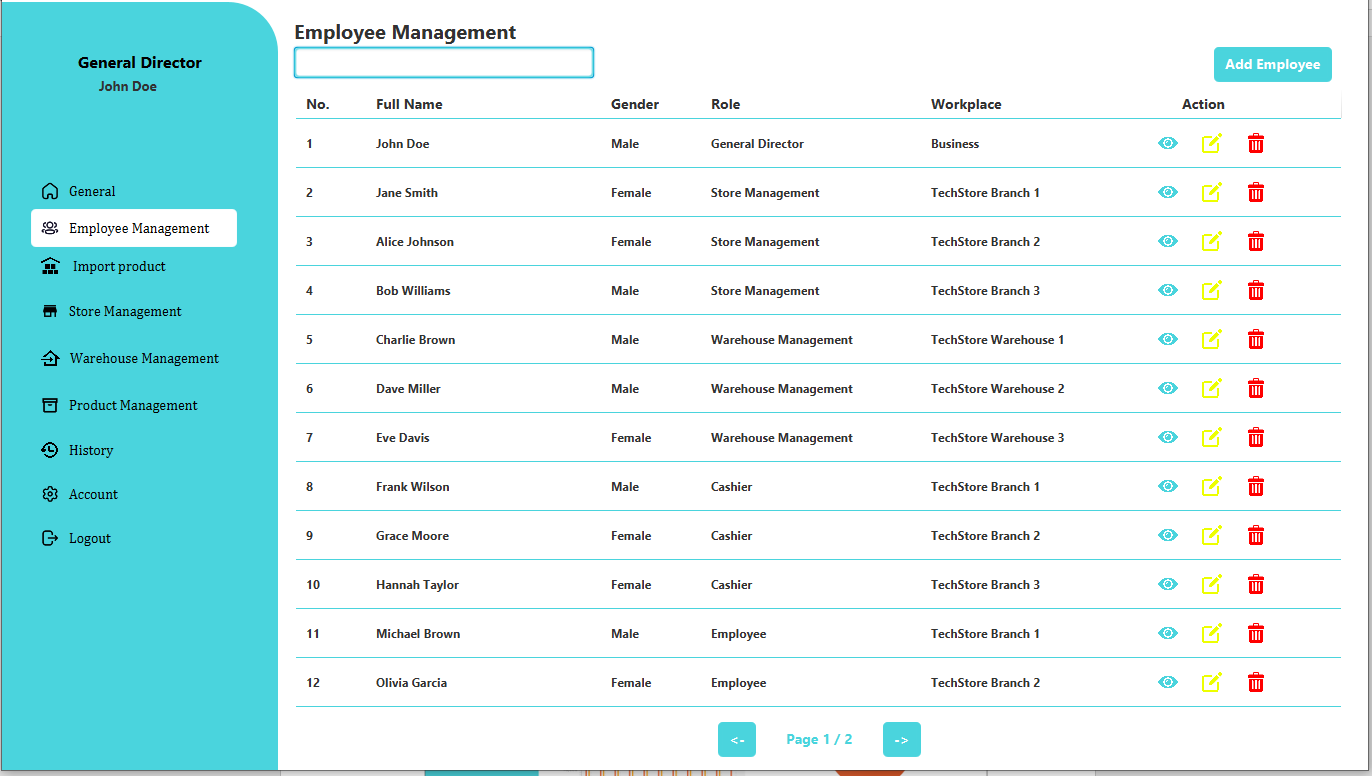
* + - * **Confirm payment**

****

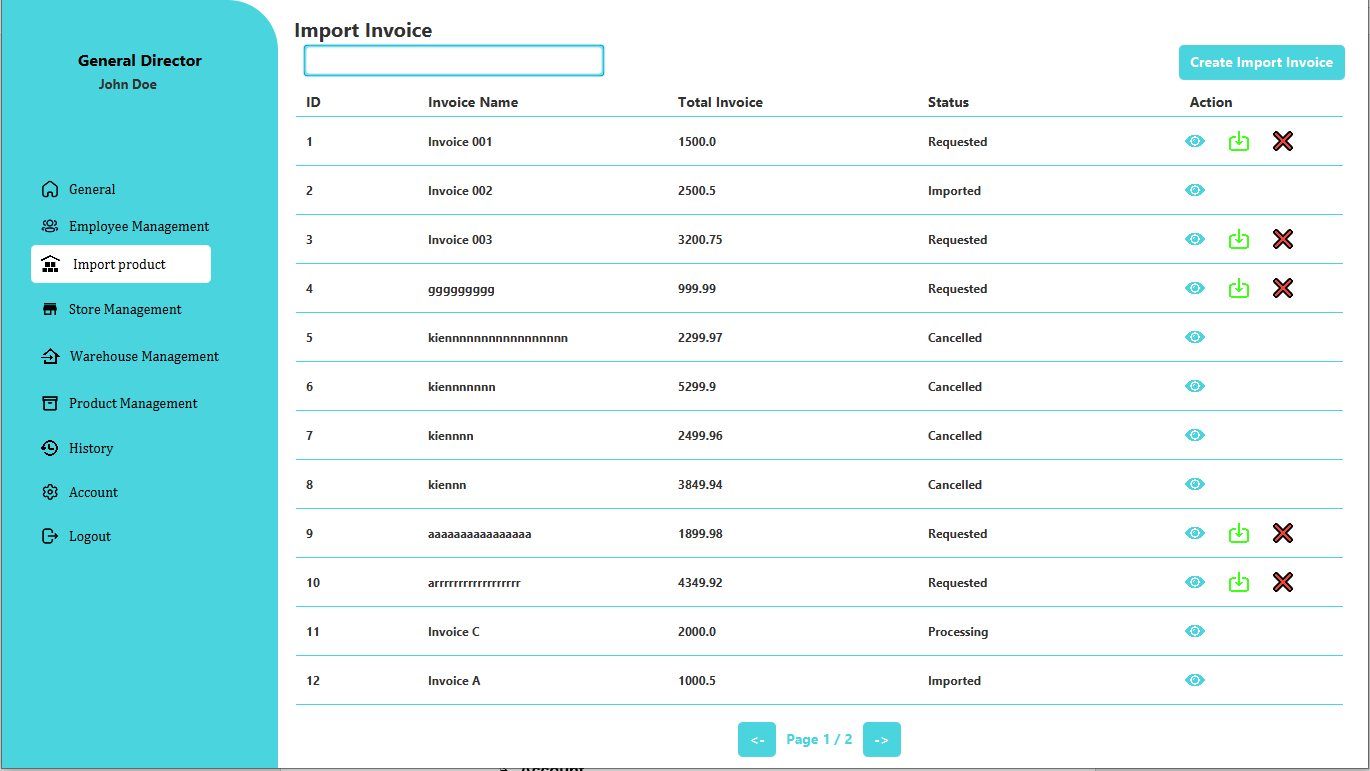
* + - **General Director**
      * **General**

****

* + - * **Employee Management**

****

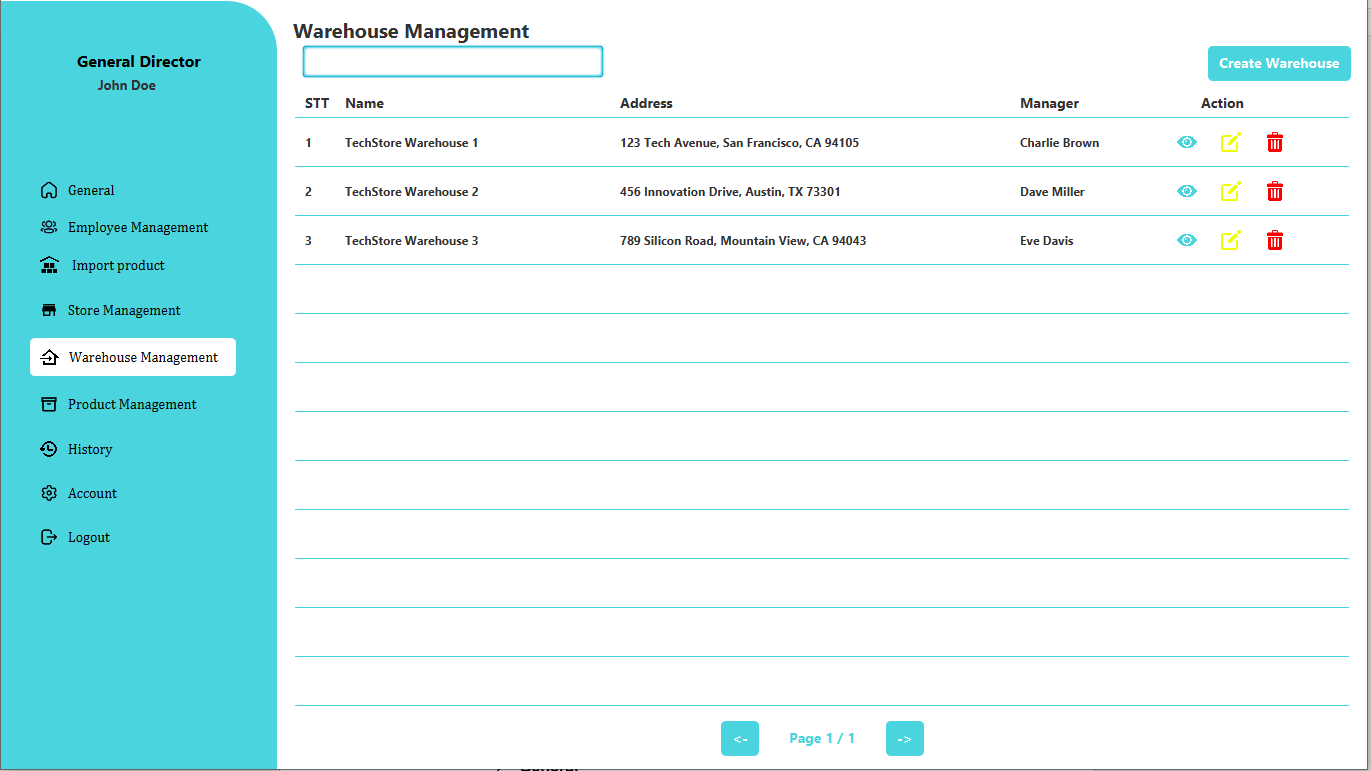
* + - * **Import Product**

****

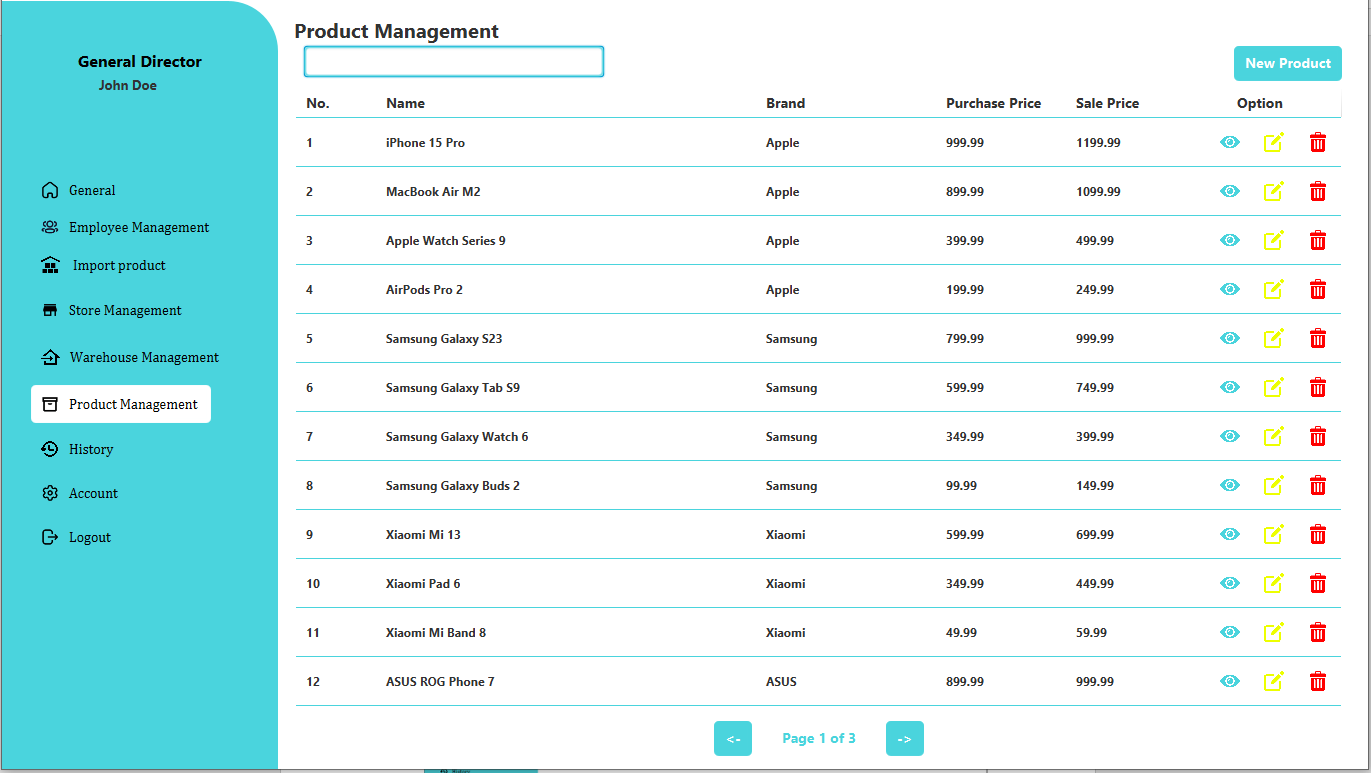
* + - * **Store Management**

****

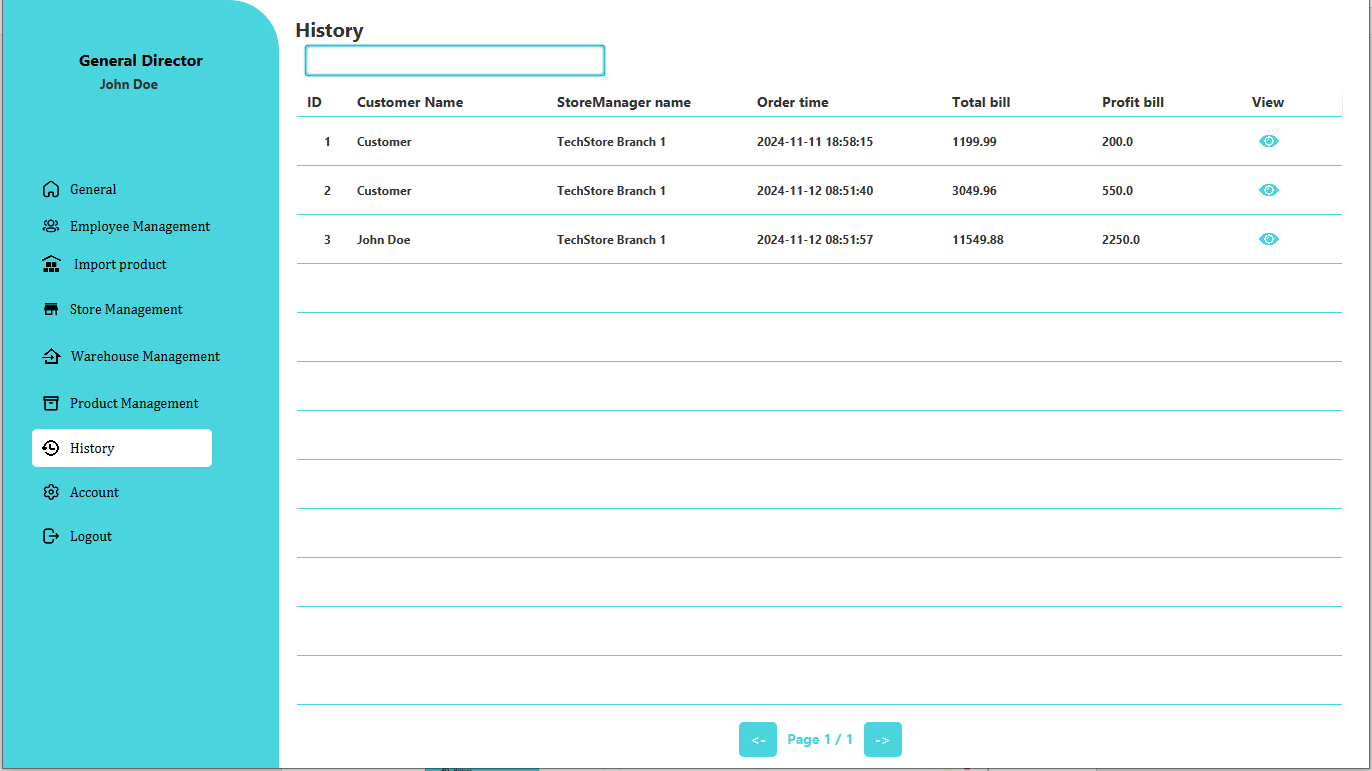
* + - * **Warehouse Management**

****

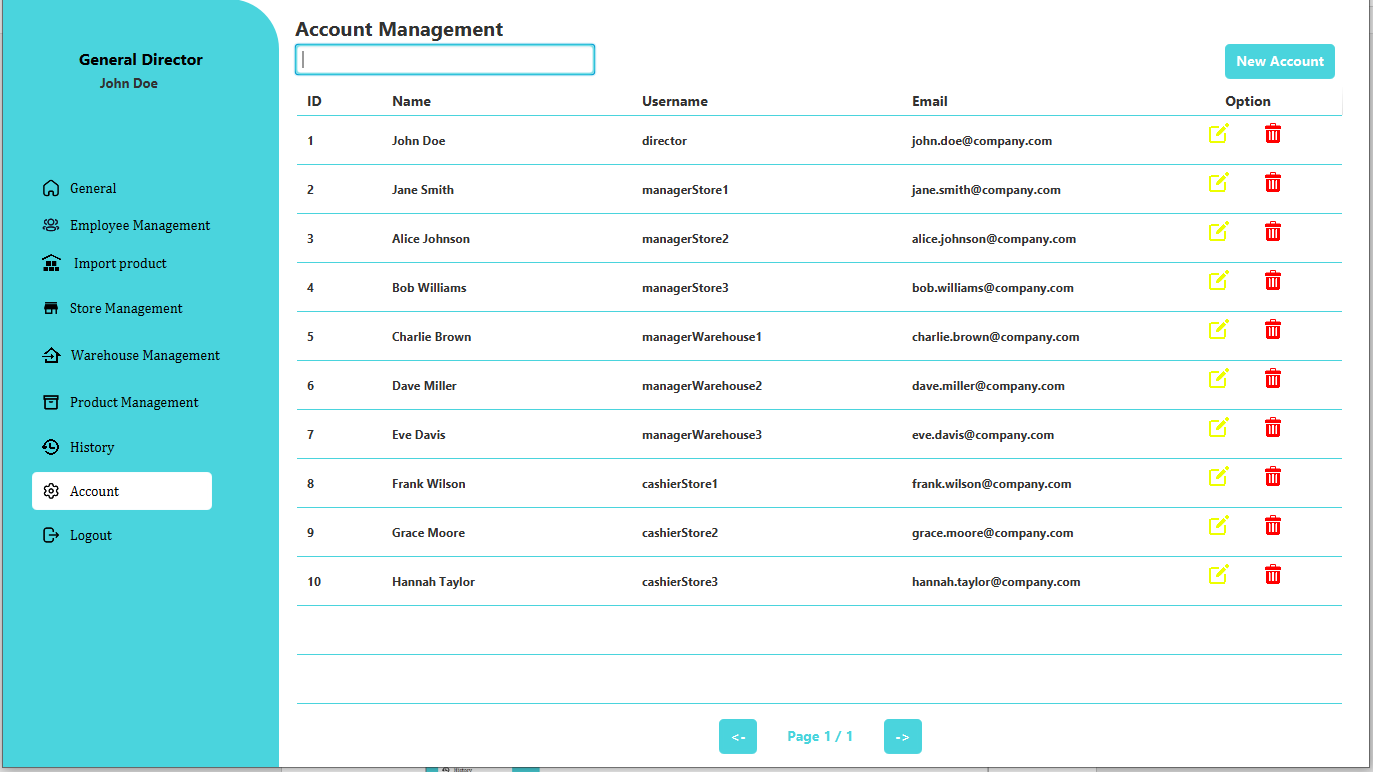
* + - * **Product Management**

****

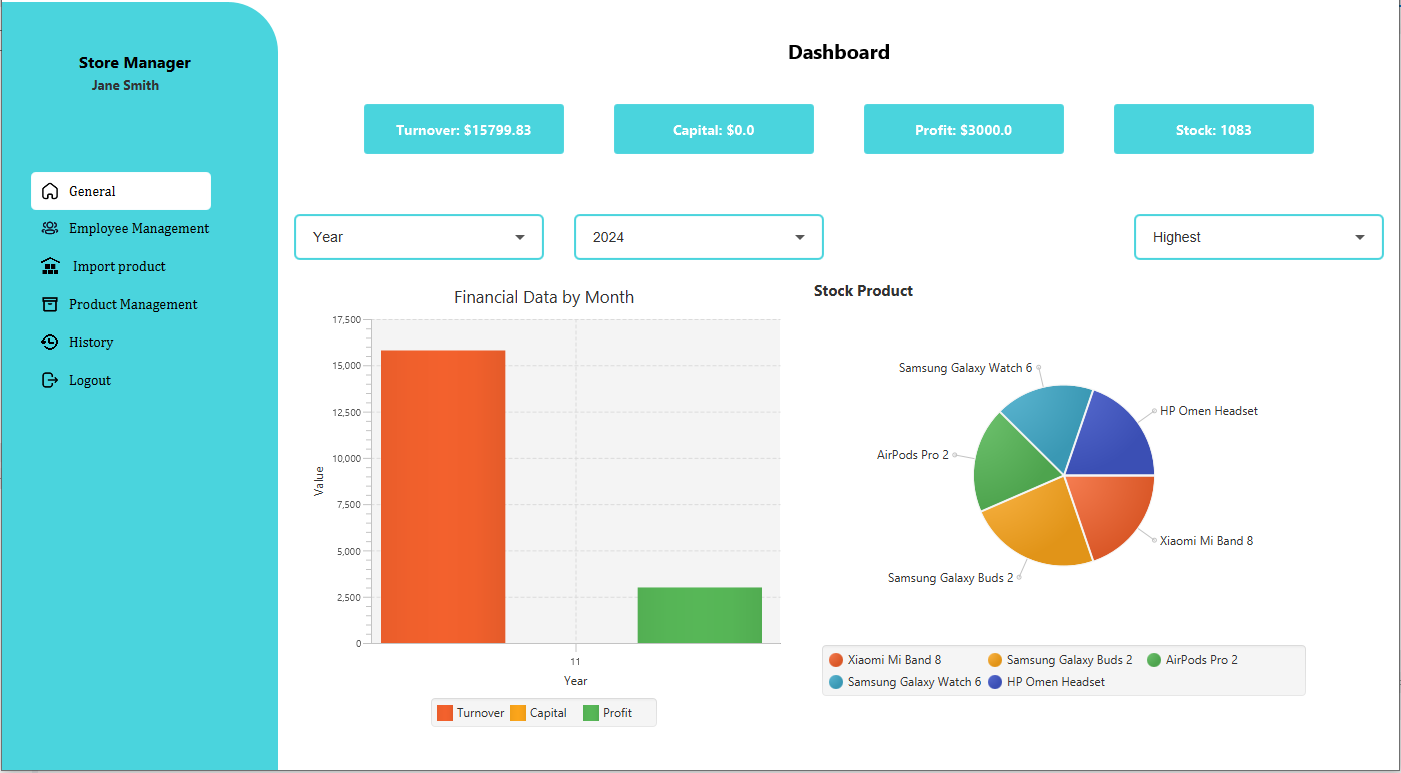
* + - * **History**

****

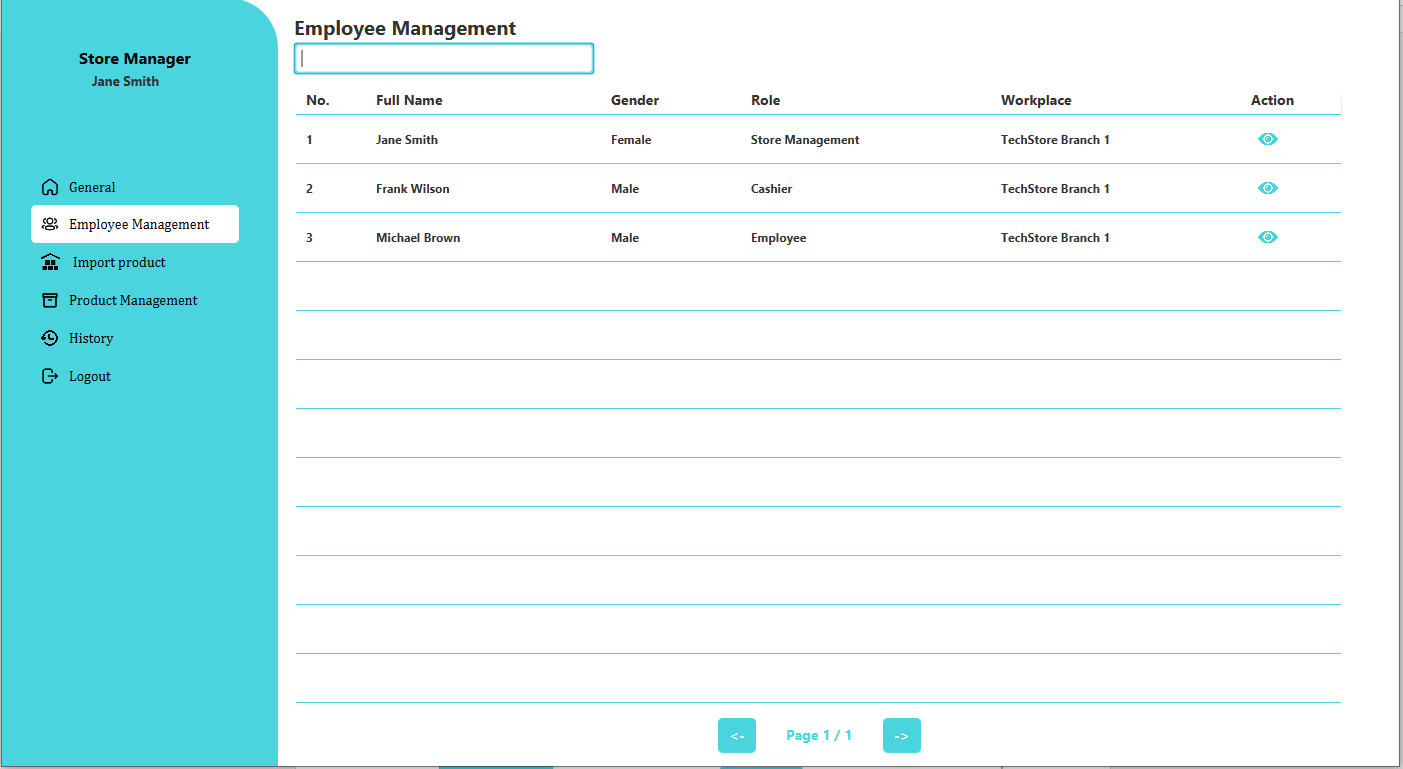
* + - * **Account**

****

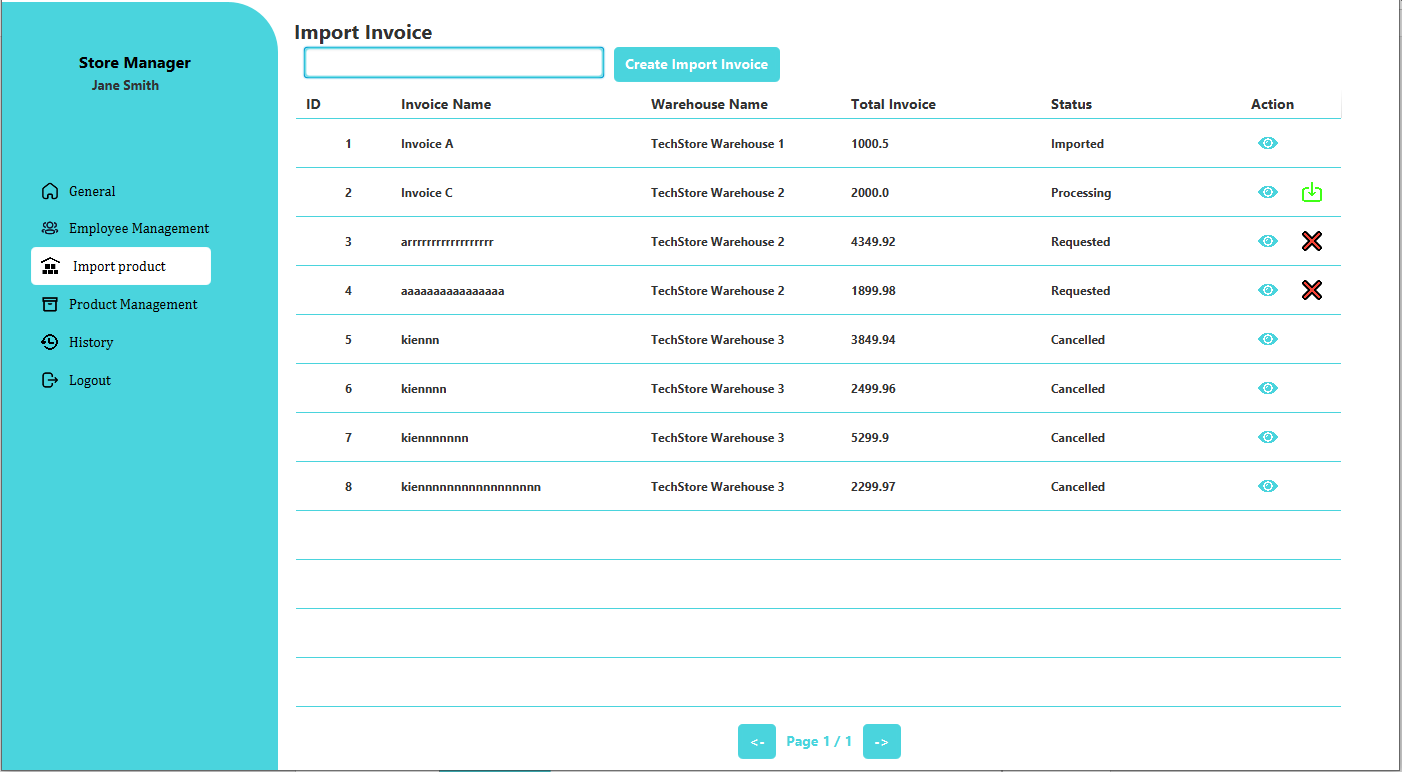
* + - **Store Management**
      * **General**



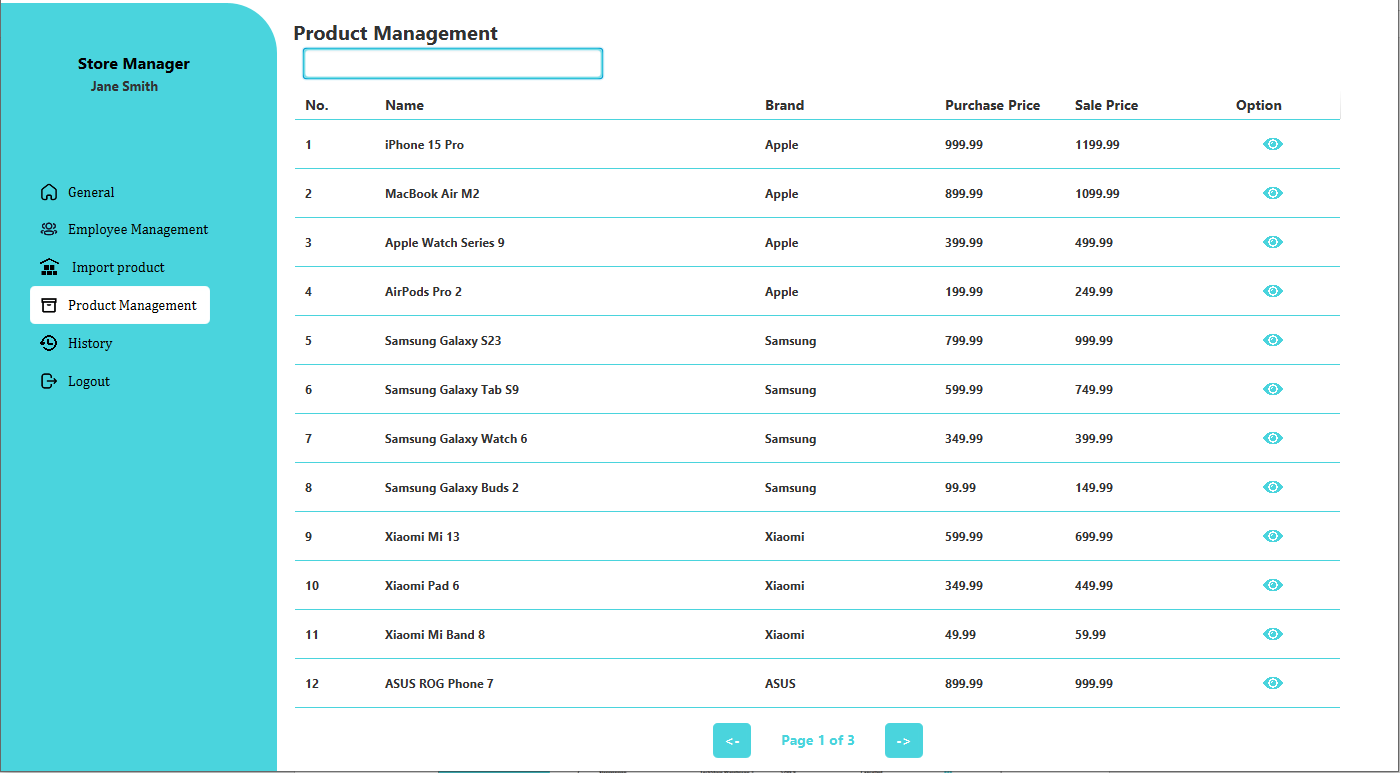
* + - * **Employee Management**



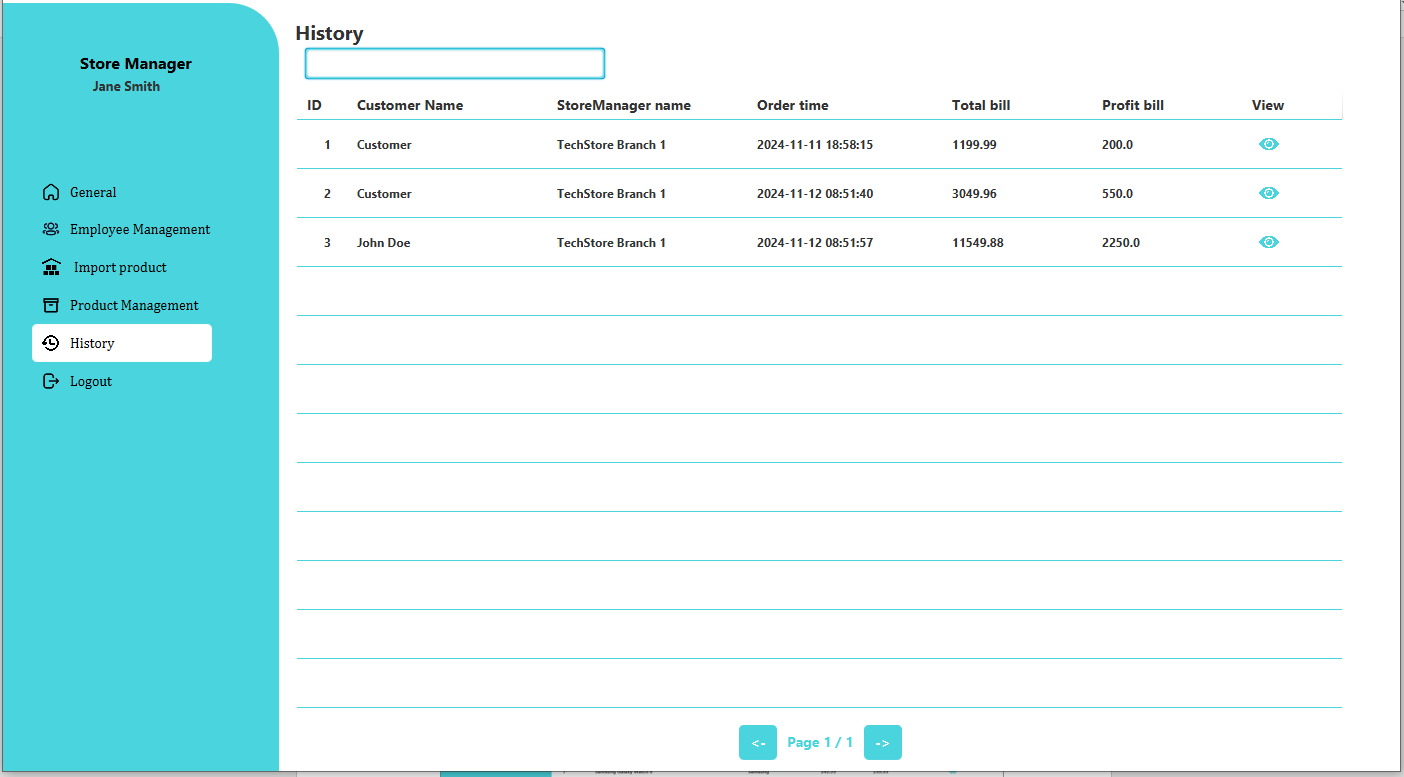
* + - * **Import Product**



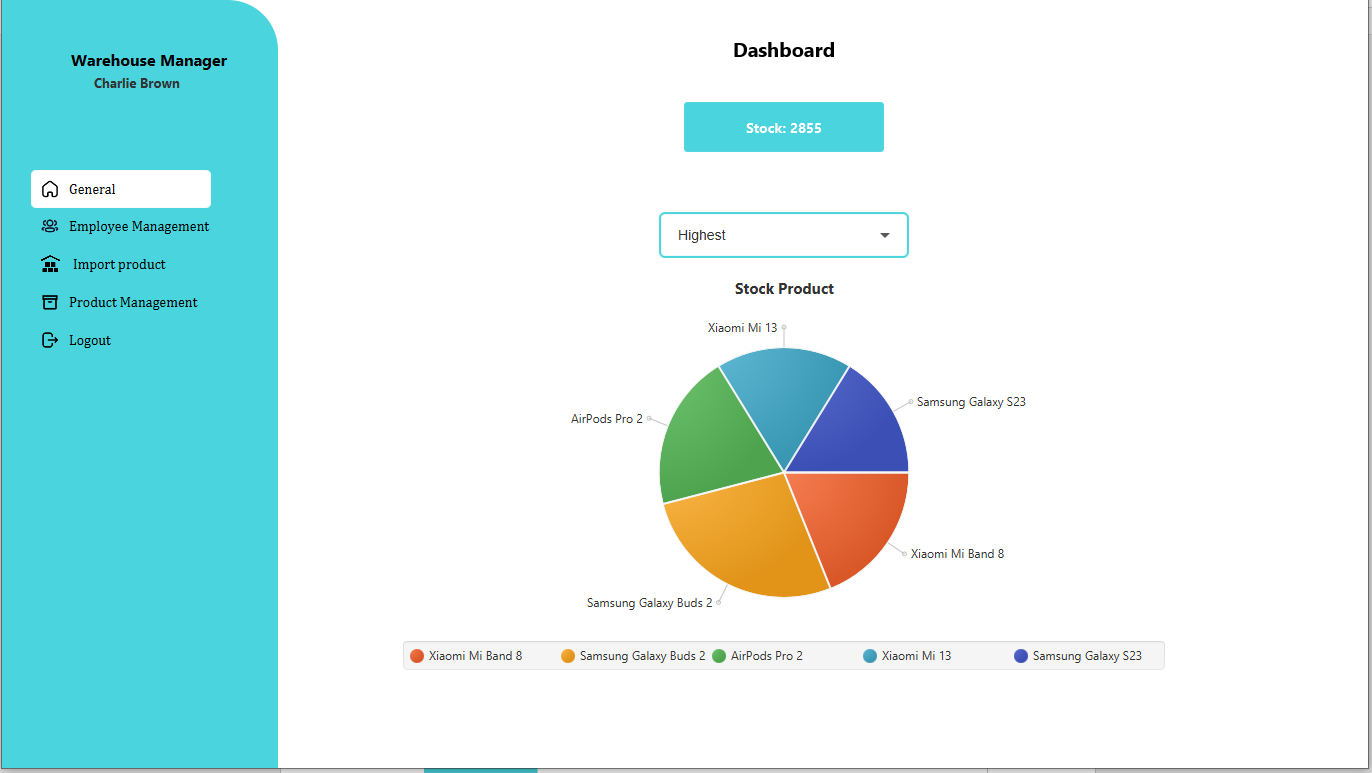
* + - * **Product Management**



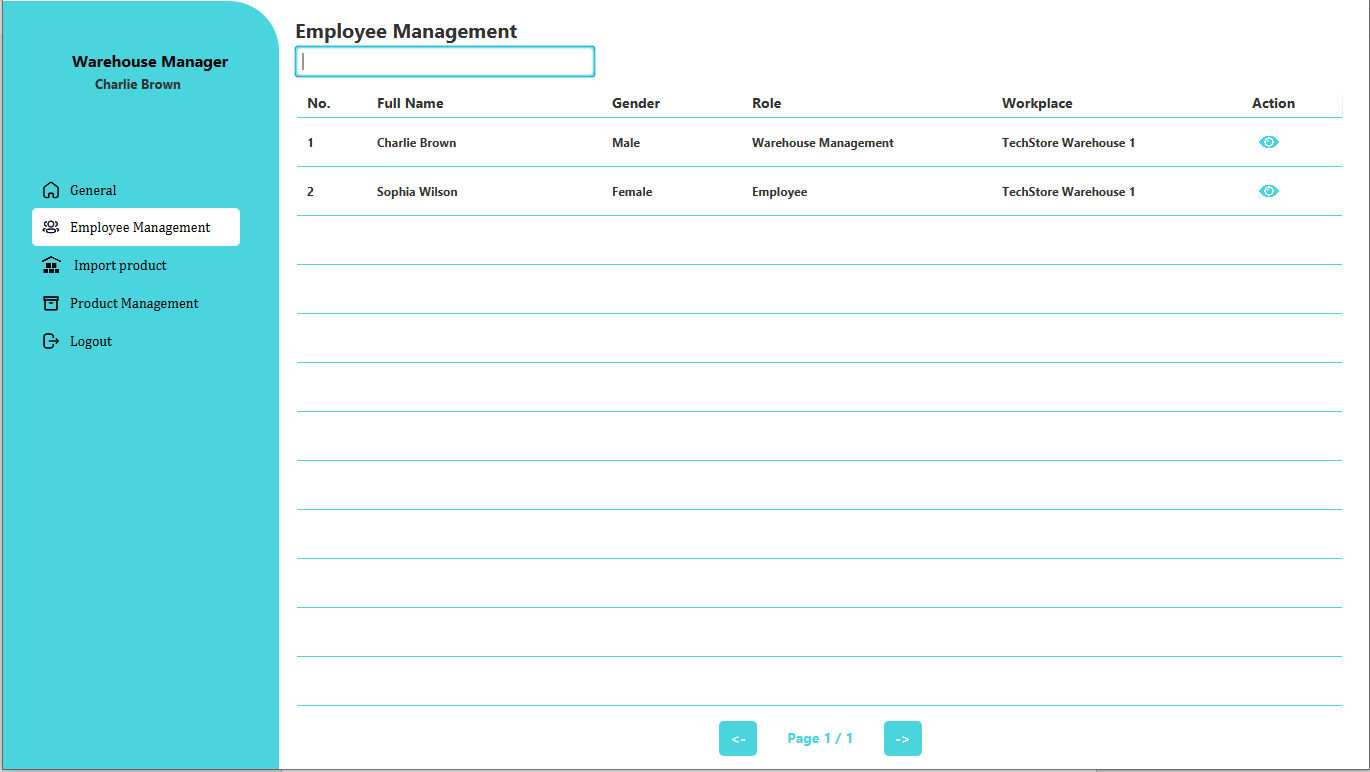
* + - * **History**



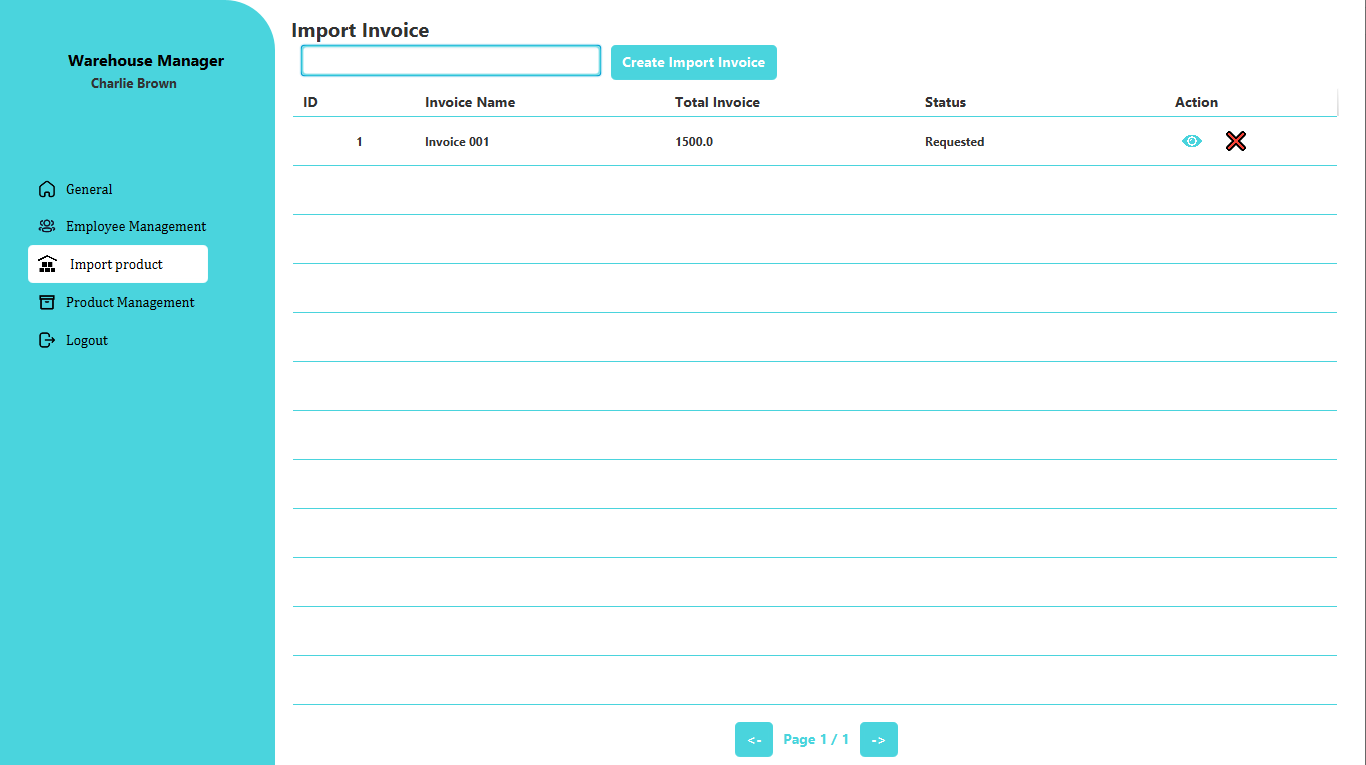
* + - **Warehouse Management**
      * **General**



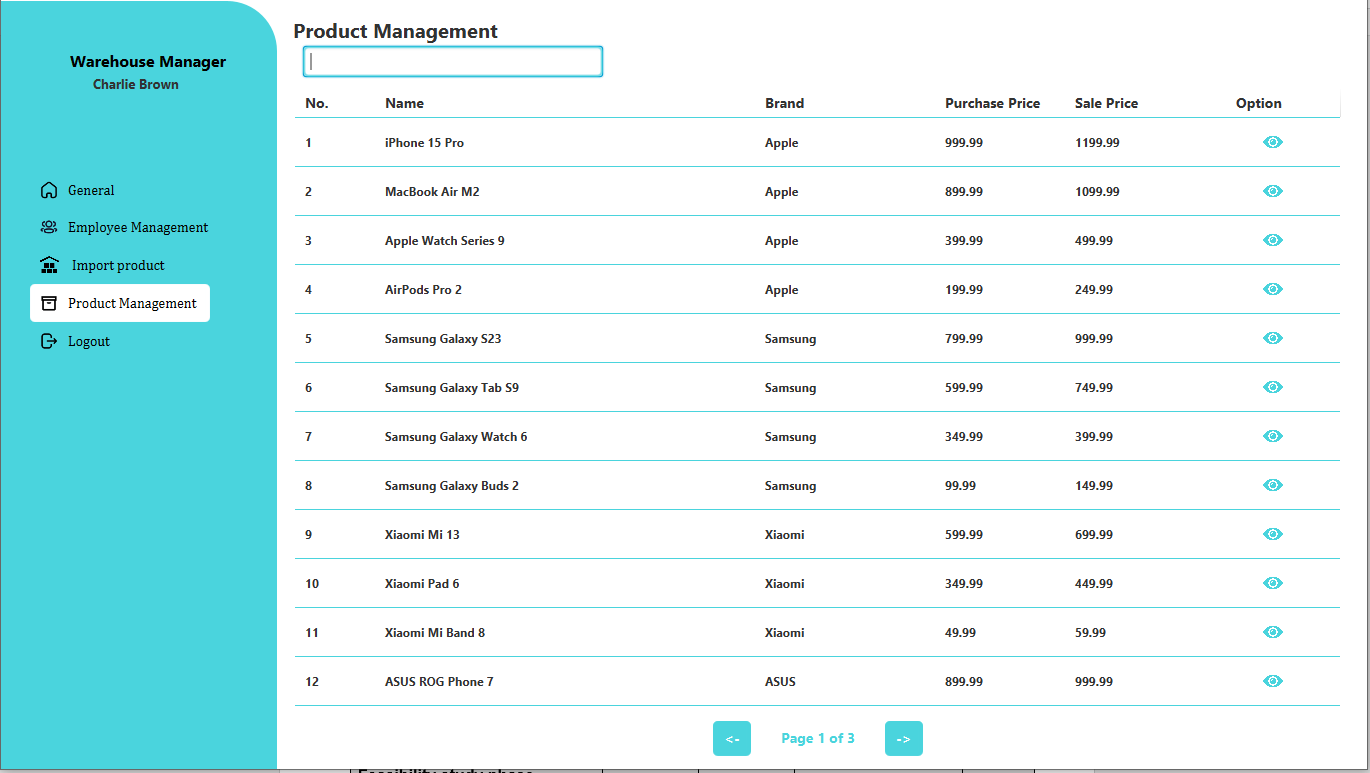
* + - * **Employee Management**



* + - * **Import Product**



* + - * **Product Management**



# Task sheet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Start time** | **Finish time** | **Implemented by** | **Evaluation(%)** |
| **Feasibility study phase** |  |  |  |  |
| Write the document | 30/09/2024 | 3/10/2024 | Le Trung Kien | 100 |
| Comments | 3/10/2024 | 3/10/2024 | Le Trung Kien | 100 |
| Review and edit content | 28/10/2024 | 30/10/2024 | Le Trung Kien | 100 |
| **Requirement Analysis phase** |  |  |  |  |
| Write the document | 30/09/2024 | 3/10/2024 | Le Trung Kien  Nguyen Thi Yen | 20  80 |
| Comments | 3/10/2024 | 3/10/2024 | Le Trung Kien | 100 |
| Review and edit content | 28/10/2024 | 30/10/2024 | Le Trung Kien | 80  20 |
| **Design phase** |  |  |  |  |
| Entity Relationship Diagram | 30/09/2024 | 3/10/2024 | Le Trung Kien  Nguyen Thi Yen | 80  20 |
| Comments | 3/10/2024 | 3/10/2024 | Le Trung Kien | 100 |
| Database Relationshp  Diagram | 30/10/2024 | 3/10/2024 | Le Trung Kien  Nguyen Thi Yen | 80  20 |
| Comments | 3/10/2024 | 5/10/2024 | Le Trung Kien  Nguyen Minh Quang | 70  30 |
| Function Requirement | 3/10/2024 | 10/10/2024 | Le Trung Kien | 100 |
| Comments | 10/10/2024 | 10/10/2024 | Le Trung Kien | 100 |
| System Requirement | 9/10/2024 | 10/10/2024 | Le Trung Kien | 100 |
| Commets and Edit | 10/10/2024 | 15/10/2024 | Le Trung Kien | 100 |
| **Development phase** |  |  |  |  |
| Build Project | 30/09/2024 | 1/10/2024 | Le Trung Kien | 100 |
| UI Login | 1/10/2024 | 5/10/2024 | Tran Bao Viet  Nguyen Dinh Chien | 50  50 |
| Server-side Login | 1/10/2024 | 5/10/2024 | Tran Bao Viet  Nguyen Dinh Chien | 50  50 |
| UI Cashier | 5/10/2024 | 10/10/2024 | Tran Bao Viet  Nguyen Dinh Chien  Nguyen Minh Quang | 70  15  15 |
| Server-side Cashier | 5/10/2024 | 14/10/2024 | Tran Bao Viet  Nguyen Minh Quang  Nguyen Dinh Chien  Le Trung Kien | 10  40  40  10 |
| UI General Director | 15/10/2024 | 28/10/2024 | Tran Bao Viet  Nguyen Minh Quang  Nguyen Dinh Chien | 70  10  20 |
| Server-side General Director | 15/10/2024 | 4/11/2024 | Tran Bao Viet  Nguyen Minh Quang  Nguyen Dinh Chien  Le Trung Kien | 10  10  10  70 |
| UI Store Management | 4/11/2024 | 8/11/2024 | Tran Bao Viet | 100 |
| Server-side Store Management | 4/11/2024 | 8/11/224 | Le Trung Kien  Nguyen Minh Quang  Nguyen Dinh Chien | 70  15  15 |
| UI Warehouse Management | 4/11/2024 | 8/11/2024 | Tran Bao Viet | 100 |
| Server-side Warehouse Management | 4/11/2024 | 8/11/2024 | Le Trung Kien  Nguyen Minh Quang  Nguyen Dinh Chien | 70  15  15 |
| **Testing phase** |  |  |  |  |
| Test and Fix Bug | 6/11/2024 | 8/11/2024 | Le Trung Kien  Nguyen Dinh Chien  Nguyen Minh Quang  Tran Bao Viet | 60  15  15  15 |
| **Deployment phase** |  |  |  |  |
| Write The  Document | 30/09/2024 | 3/10/2024 | Nguyen Thi Yen  Le Trung Kien | 80  20 |
| Comment and Edit | 3/10/2024 | 3/10/2024 | Le Trung Kien | 100 |

# Validation Checklists

* 1. **Login**
     + Verify that the login account has a non-empty username.
     + Ensure the password meets the required format (minimum length, includes special characters).
     +  Cross-check login credentials with the database (DB).
     +  Redirect to the dashboard upon successful authentication; display an error message if authentication fails.
  2. **Customer & Invoice Management**
     + Verify customer information (name, phone number) is valid and non-empty.
     + Products, quantities, and prices in the invoice must be positive numbers.
     +  Confirm that customer information is saved correctly in the customers table and invoices in the receipts table.
     +  Display a confirmation message upon successful saving.
  3. **Product Inventory Management**
     + Ensure added products have valid names, purchase prices, and sale prices.
     + Inventory quantity is always updated automatically with every sale or stock receipt.
     +  The system should alert users when inventory drops below a minimum threshold.
     +  Verify accurate display of stock quantity for each product.
  4. **Employee Management**
     + Employee information (name, birth date, phone number, address) must be in the correct format.
     + Only allow deletion of employees who do not have an account.
     +  Check access rights based on roles (Cashier, Store Manager, Warehouse Manager, General Director).
     +  Edits and updates should immediately reflect in the employee list.
  5. **Sales and Revenue Reporting**
     + Ensure accurate daily, weekly, and monthly revenue and profit statistics.
     + Sales analysis reports for each store must align with data.
     +  Ensure all reports update in real time.
  6. **Supplier and Purchase Management**
     + Supplier order information must be accurate (supplier name, product, quantity).
     + Purchase price and quantity must be positive numbers.
     +  Verify that orders match the stock received in the warehouse.
     +  Store detailed order and supplier information for reporting.
  7. **System Security**
     + Encrypt passwords before storing them in the DB.
     + Check multi-factor authentication (if implemented) during the login process.
     +  Ensure appropriate permissions for each user role.
     +  Record and save user actions for activity logs.
  8. **Data Integration and Extensibility**
     +  Test integration capabilities with external platforms, such as shipping systems or CRM.
     +  Ensure smooth system performance during high transaction volumes.