Software Design Document (SDD) for FreshMart Inventory Management System

Hania Maher, Jana Tarek, Rawan Jalaleldin, Bahig Mahmoud, Ahmed Osama

Version	Date	Reason for Change
1.0	27-12-2024	SDD first version's description are defined.

Table 1: Document Version History

Contents

1 Introduction

1.1 Abstract

FreshMart Inventory Management System is a web-based application designed to improve inventory management in supermarkets and retail businesses. The project increases operational efficiency by providing effective solutions for tracking stock levels, managing orders and facilitating customer interaction. All control is provided through a user-friendly interface. The system uses PHP for server-side scripting, MySQL for database management and HTML/CSS for front-end development to ensure user experience. Future enhancements such as payment processing and analytics tools.

1.2 Purpose

The purpose of this Software Design Document (SDD) is to provide a comprehensive overview of the FreshMart Inventory System, detailing its design, architecture, and implementation strategies. This document serves as a guide for developers, stakeholders, and project managers involved in the development and maintenance of the system.

1.3 Scope

The FreshMart Inventory System is designed to manage inventory for supermarkets, allowing users to track products, manage orders, and handle supplier information. The system will include user authentication, product management, order processing, and reporting functionalities.

1.4 Overview

The FreshMart Inventory System will be a web-based application built using PHP and MySQL, providing a user-friendly interface for both administrators and regular users. The system will support various user roles, including Admin, Supplier, and User, each with specific permissions and functionalities.

1.5 Similar Systems

There are existing supermarket-style inventory management systems that offer similar functionalities, such as Talabat, and insta shop. These systems allow businesses to manage inventory, process sales, and track customer data. How- ever, FreshMart focuses on stock level tracking with real-time supplier integration and automatic reordering tailored for smaller-scale supermarkets.

1.6 Intended Audience

This document is intended for:

- **Software Developers:** To provide detailed technical specifications and guidelines for implementing the FreshMart Inventory Management System.
- **Project Managers:** To facilitate project planning, resource allocation, and timeline management by outlining project scope and deliverables.
- **System Architects:** To offer insights into the system's architecture, design patterns, and integration points for effective system design and scalability
- **Quality Assurance Engineers:** To establish testing criteria and quality standards, ensuring the system meets functional and non-functional requirements.
- **Stakeholders:** To communicate project objectives, benefits, and progress, ensuring alignment with business goals and user needs.

1.7 Reference Material

Software Requirement Specification Document for FreshMart Management Inventory System

2 System Overview

2.1 System Scope

The system will cover the following functionalities:

- User registration and login
- Users can view products and add them to the cart
- User management (create, read, update, delete)
- Product management (create, read, update, delete)
- Order management (create, read, update, delete)
- Supplier management
- Reporting and analytics

2.2 System Objectives

- To provide a reliable inventory management solution for supermarkets.
- To ensure user-friendly navigation and accessibility.
- To implement secure user authentication and data protection.

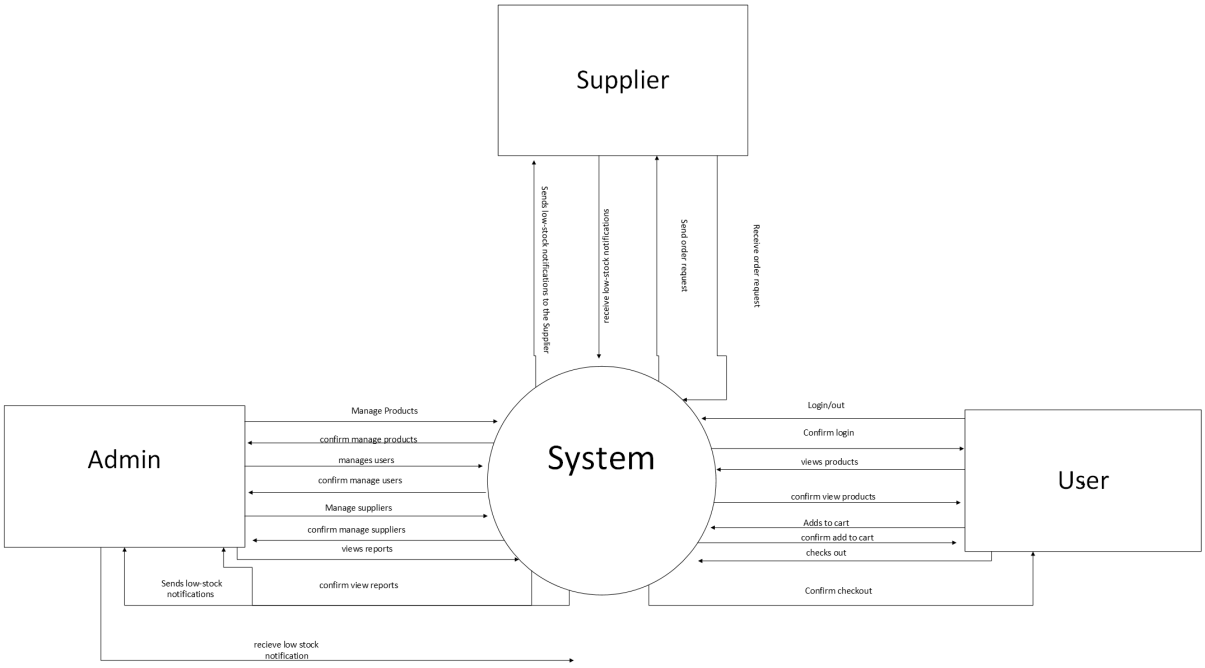
2.3 System Timeline

- Phase 1: Requirements Gathering (2 week)
- Phase 2: System Design (3 weeks)
- Phase 3: Development (5 weeks)
- Phase 4: Testing (2 weeks)

3 Design Viewpoints

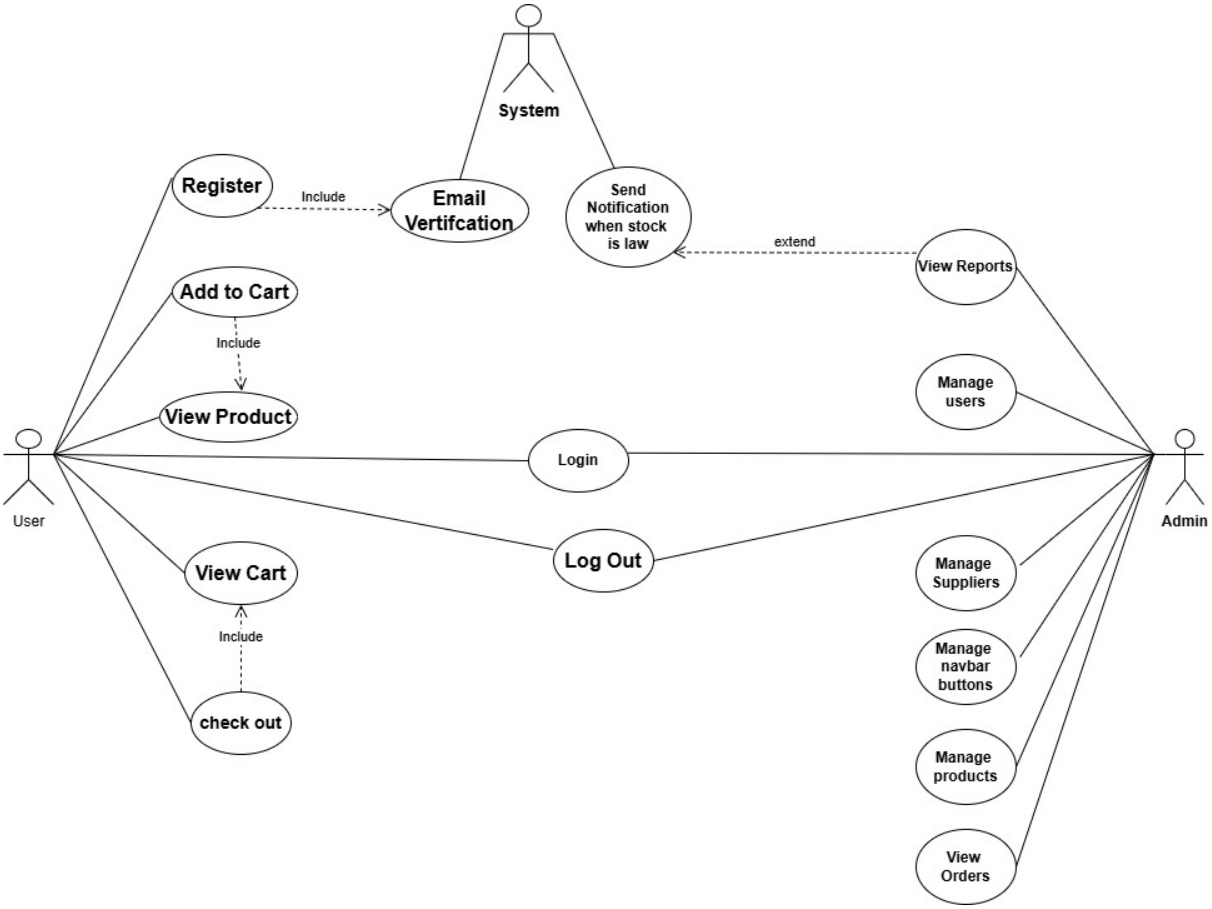
3.1 Context Diagram

Description: The context diagram shows the system's interactions with external entities.



3.2 Use Case Diagram

Description: The use case diagram illustrates the interactions between users and the system functionalities.



3.3 Tabular Description for Use Case

Register

Actor	User
Description	Allow a new user to create an account.
Data	User details (name, email, password)
Stimulus	User submits registration details.
Response	Account is created.
Comments	Includes email verification to ensure that it's written correctly.

Email Verification

Actor	System
Description	Email verification to ensure that it's written correctly.
Data	User email
Stimulus	Triggered by user registration.
Response	Account is created.
Comments	Mandatory step for account registration.

Add to Cart

Actor	User
Description	Enables a user to add a product to their shopping cart.
Data	Product details
Stimulus	User clicks "Add to Cart" for a specific product.
Response	Product added to the user's cart.
Comments	The user can add, remove, or update the quantity of products.

View Product

Actor	User
Description	Allows users to browse and view product details.
Data	Product information
Stimulus	User wants to view details.
Response	Product details displayed.
Comments	Provides detailed descriptions and images of products.

View Cart

Actor	User
Description	Enables a user to review items added to their shopping cart.
Data	Cart items
Stimulus	User clicks on the "View Cart" option.
Response	Displays all items currently in the cart.
Comments	Includes item quantity and price details.

Check Out

Actor	User
Description	Facilitates the completion of a purchase of items in the cart.
Data	Cart items, payment details
Stimulus	User proceeds to checkout.
Response	Order is confirmed, and payment processed.
Comments	Includes validating payment.

Login

Actor	User/Admin
Description	Allows users or admins to access their accounts.
Data	Username and password
Stimulus	User enters their data.
Response	User is logged in and directed to the dashboard.
Comments	Required for accessing the website.

Log Out

Actor	User/Admin
Description	Logs the user or admin out of their account.
Data	Session details
Stimulus	User clicks "Log Out."
Response	Session is terminated.
Comments	Required for logging out from the website.

Send Notification When Stock is Low

Actor	System
Description	Sends notifications to the admin when stock levels are low.
Data	Stock levels, product details
Stimulus	Stock reaches a certain number.
Response	Notification sent to the admin.
Comments	Extends the functionality of "View Reports."

View Reports

Actor	Admin
Description	Allows the admin to view detailed reports.
Data	Sales data and stock of products
Stimulus	Admin selects "View Reports."
Response	Report is displayed.
Comments	Includes sales data and stock of products.

Manage Users

Actor	Admin
Description	Enables the admin to manage user accounts.
Data	User account details
Stimulus	Admin selects "Manage Users."
Response	User accounts are added, updated, or deleted.
Comments	Includes user details.

Manage Suppliers

Actor	Admin
Description	Allows the admin to manage supplier information.
Data	Supplier details
Stimulus	Admin selects "Manage Suppliers."
Response	Supplier details are added, updated, or deleted.
Comments	Maintains updated supplier information.

Manage Navbar Buttons

Actor	Admin
Description	Enables the admin to customize navigation buttons on the website.
Data	Navigation details
Stimulus	Admin selects "Manage Navbar Buttons."
Response	Navbar is updated.
Comments	Facilitates better navigation.

Manage Products

Actor	Admin
Description	Allows the admin to add, update, or delete product details.
Data	Product information
Stimulus	Admin selects "Manage Products."
Response	Product information is updated.
Comments	When the admin adds a product, they can include an image, description, and price.

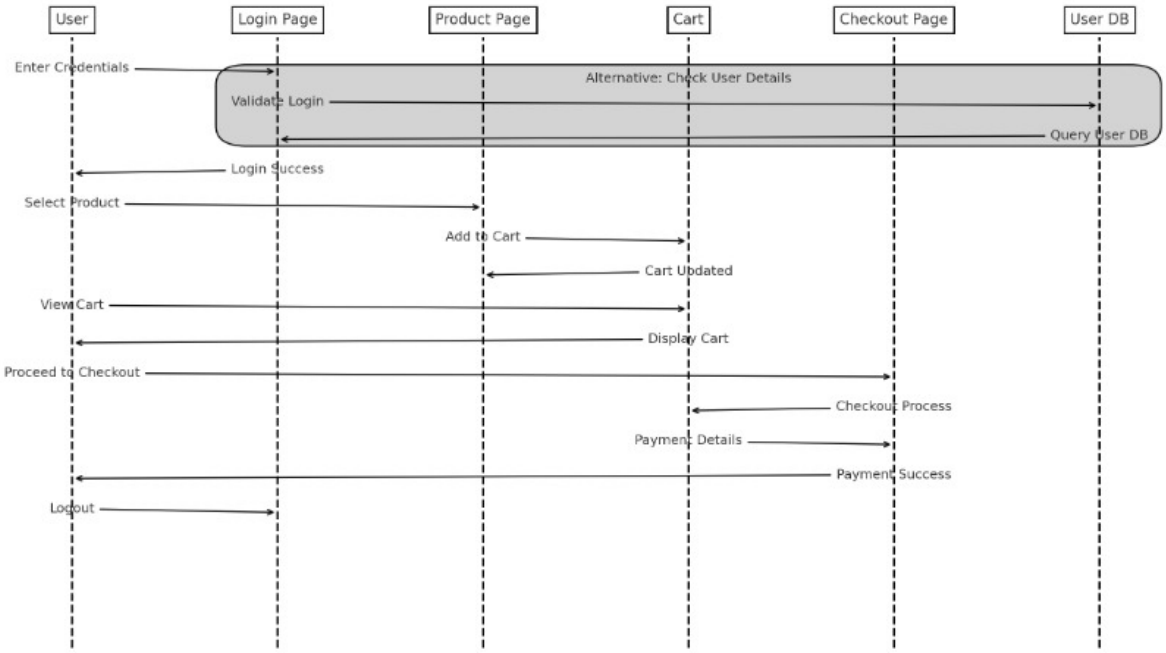
View Orders

Actor	Admin
Description	Enables the admin to view and manage all orders placed by users.
Data	Order details
Stimulus	Admin selects "View Orders."
Response	Displays all user orders.
Comments	Includes order details.

3.4 Sequence Diagram

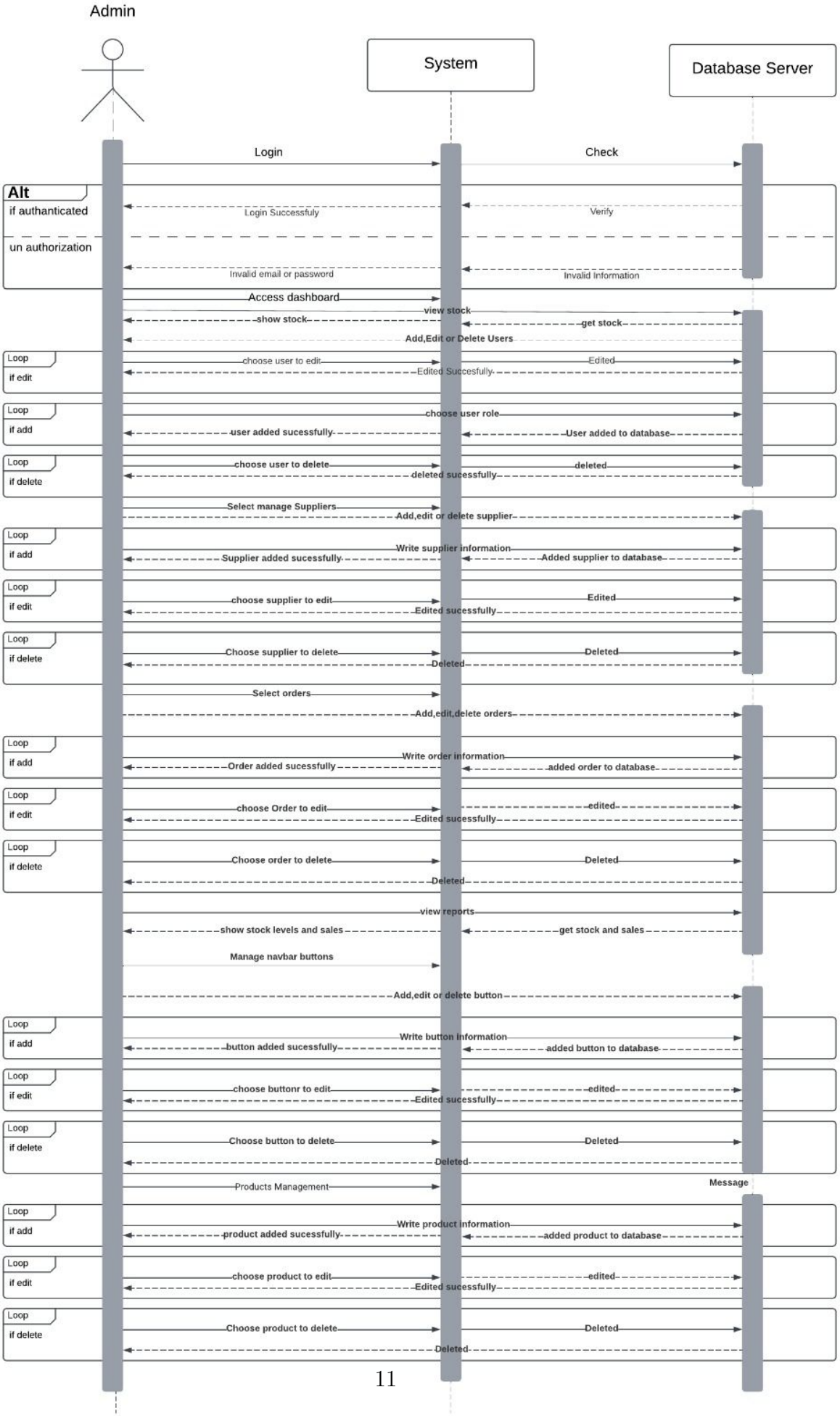
3.4.1 User Sequence Diagram

Description: The user sequence diagram illustrates the sequence of interactions between the user and the system during a specific use case.



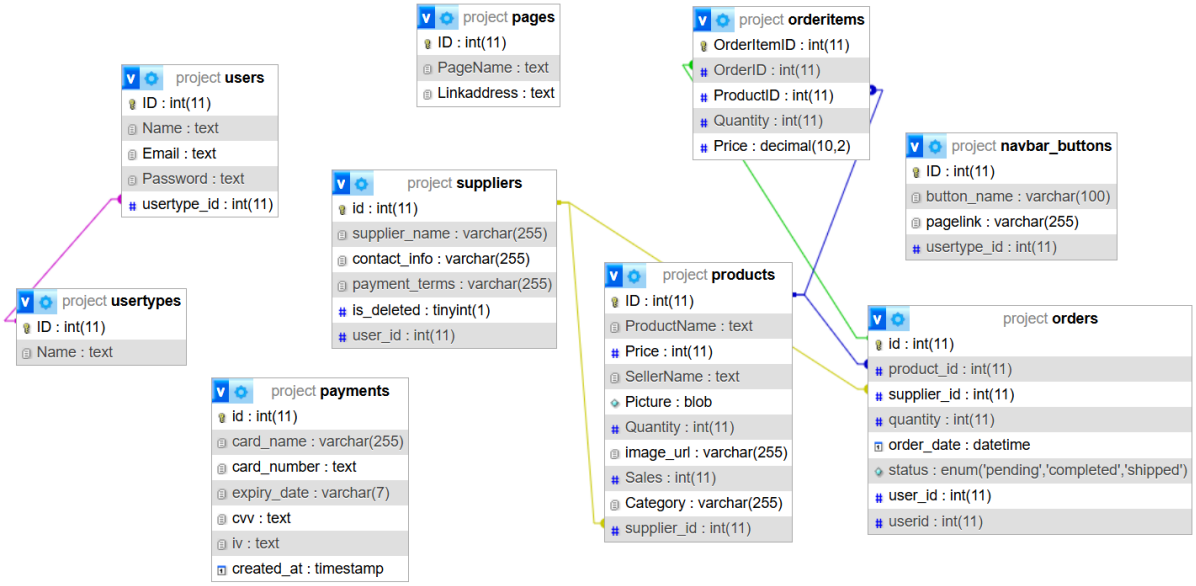
3.4.2 Admin Sequence Diagram

Description: The admin sequence diagram illustrates the sequence of interactions between the admin and the system during administrative tasks such as user management and product management.


Admin
System
Database Server
Login
Check
Alt
if authanticated
Login Successfuly
Verify
un authorization
Invalid email or password
Invalid Information
Access dashboard
view stock
show stock
get stock
Add,Edit or Delete Users
Loop
if edit
choose user to edit
Edited
Edited Succesfully
Loop
if add
choose user role
user added sucessfully
User added to database
Loop
if delete
choose user to delete
deleted
deleted sucessfully
Select manage Suppliers
Add,edit or delete supplier
Loop
if add
Write supplier information
Supplier added sucessfully
Added supplier to database
Loop
if edit
choose supplier to edit
Edited
Edited sucessfully
Loop
if delete
Choose supplier to delete
Deleted
Deleted
Select orders
Add,edit,delete orders
Loop
if add
Write order information
Order added sucessfully
added order to database
Loop
if edit
choose Order to edit
edited
Edited sucessfully
Loop
if delete
Choose order to delete
Deleted
Deleted
view reports
show stock levels and sales
get stock and sales
Manage navbar buttons
Add,edit or delete button
Loop
if add
Write button information
button added sucessfully
added button to database
Loop
if edit
choose buttonr to edit
edited
Edited sucessfully
Loop
if delete
Choose button to delete
Deleted
Deleted
Products Management
Message
Loop
if add
Write product information
product added sucessfully
added product to database
Loop
if edit
choose product to edit
edited
Edited sucessfully
Loop
if delete
Choose product to delete
Deleted
Deleted

3.5 Database Diagram

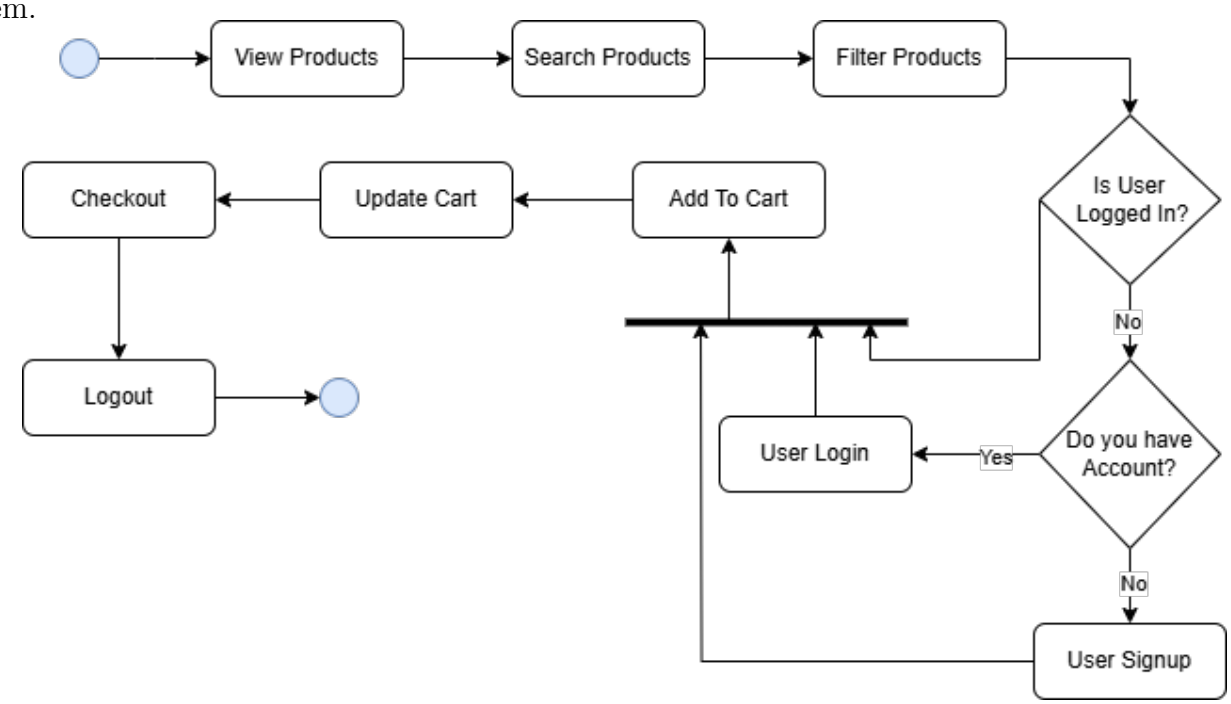
Description: The database diagram provides a visual representation of the database schema, including tables, relationships, and key attributes.



3.6 Activity Diagram

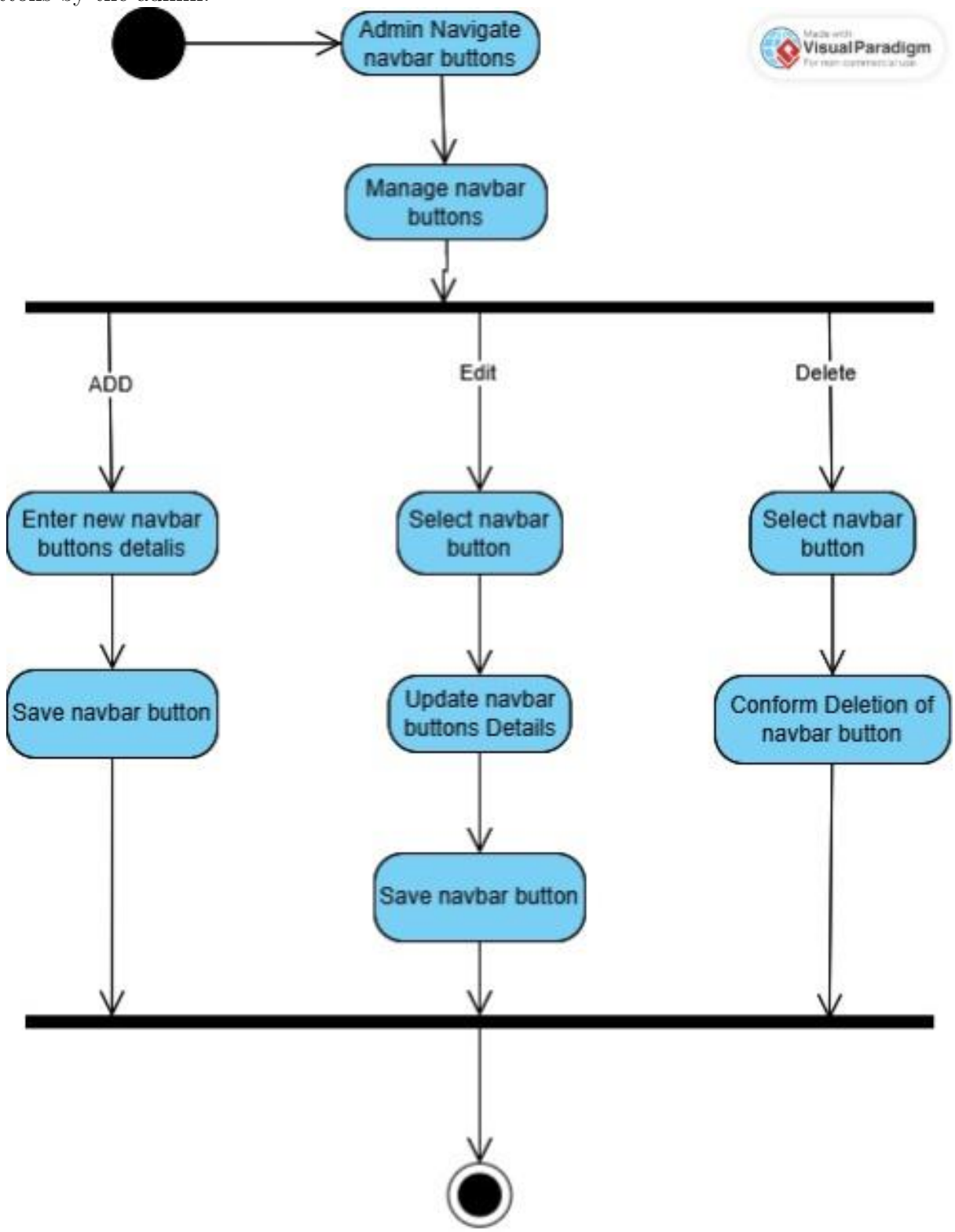
3.6.1 User Activity Diagram

Description: This diagram represents the activities performed by users within the system.



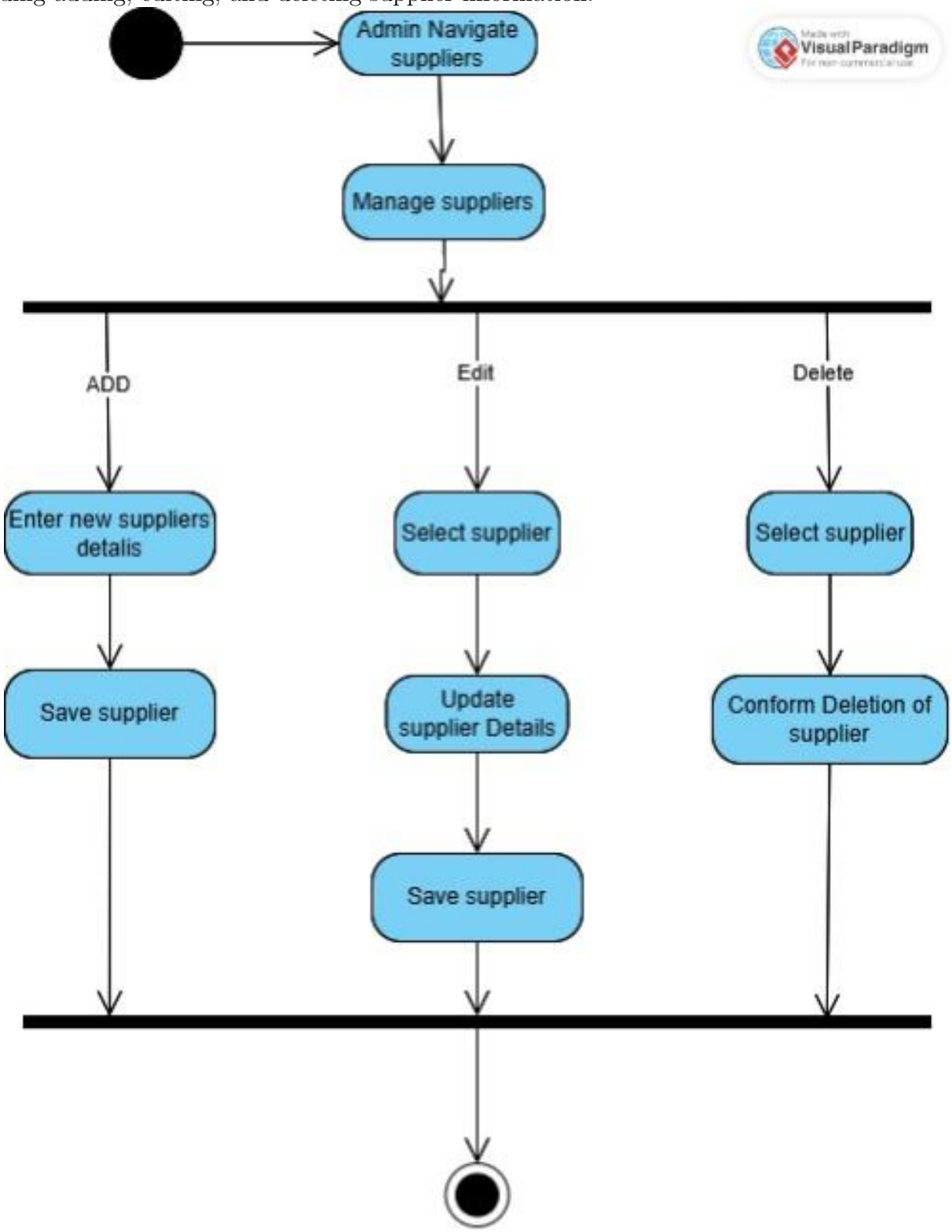
3.6.2 Admin NavBar Buttons Management Activity Diagram

Description: This activity diagram illustrates the workflow for managing navigation bar buttons by the admin.



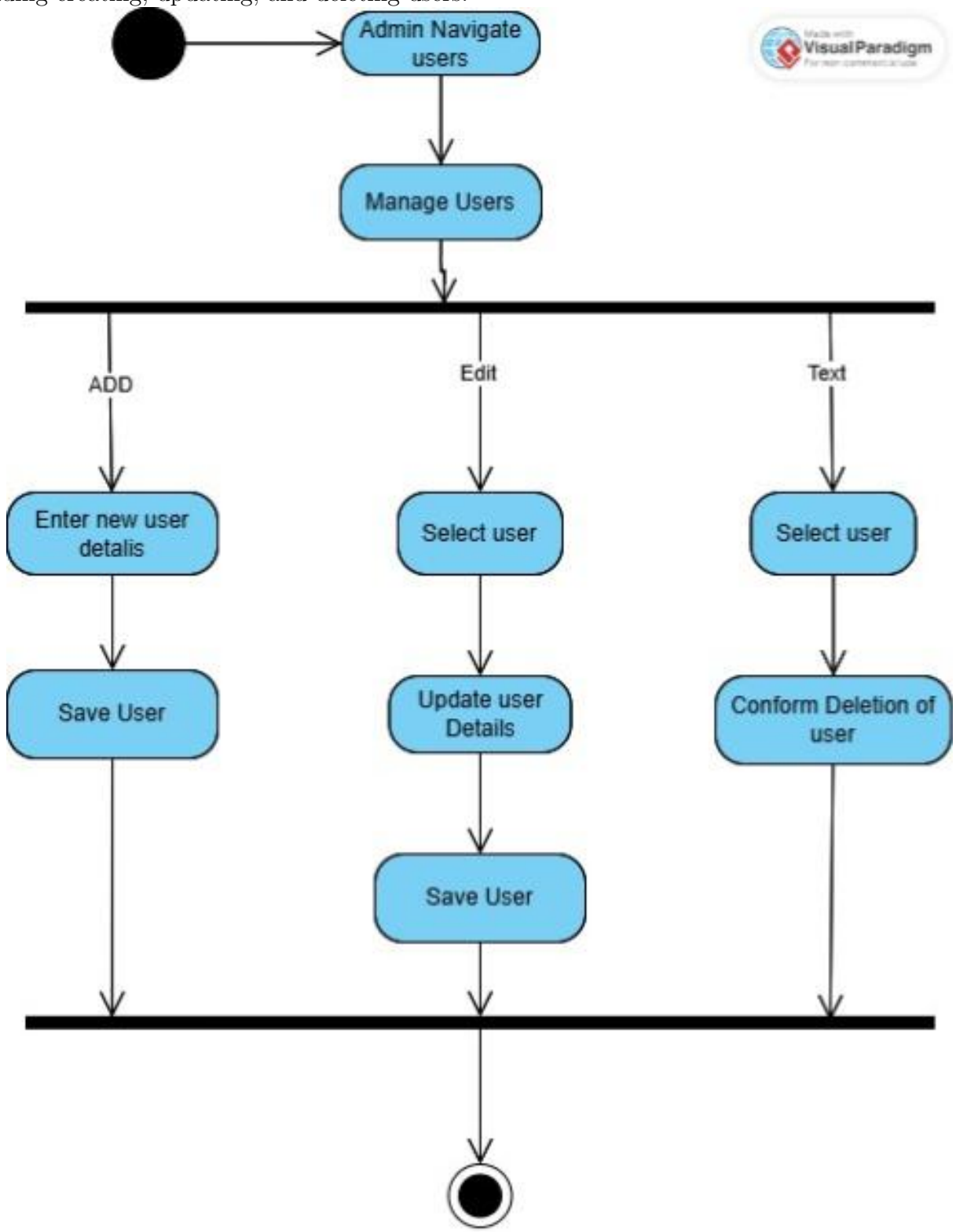
3.6.3 Admin Supplier Management Activity Diagram

Description: This diagram represents the activities involved in managing suppliers, including adding, editing, and deleting supplier information.



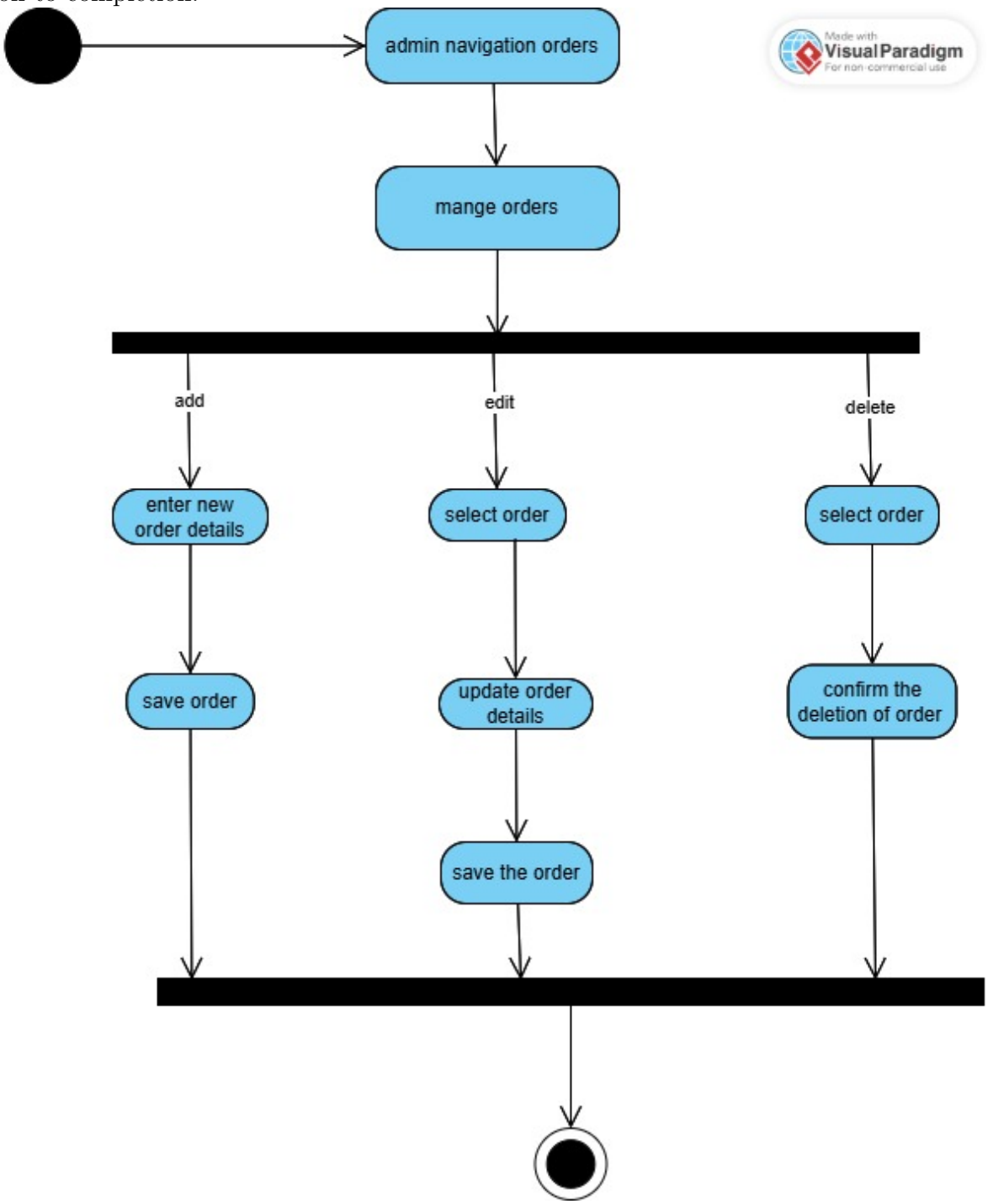
3.6.4 Admin Users Management Activity Diagram

Description: This activity diagram shows the workflow for managing user accounts, including creating, updating, and deleting users.



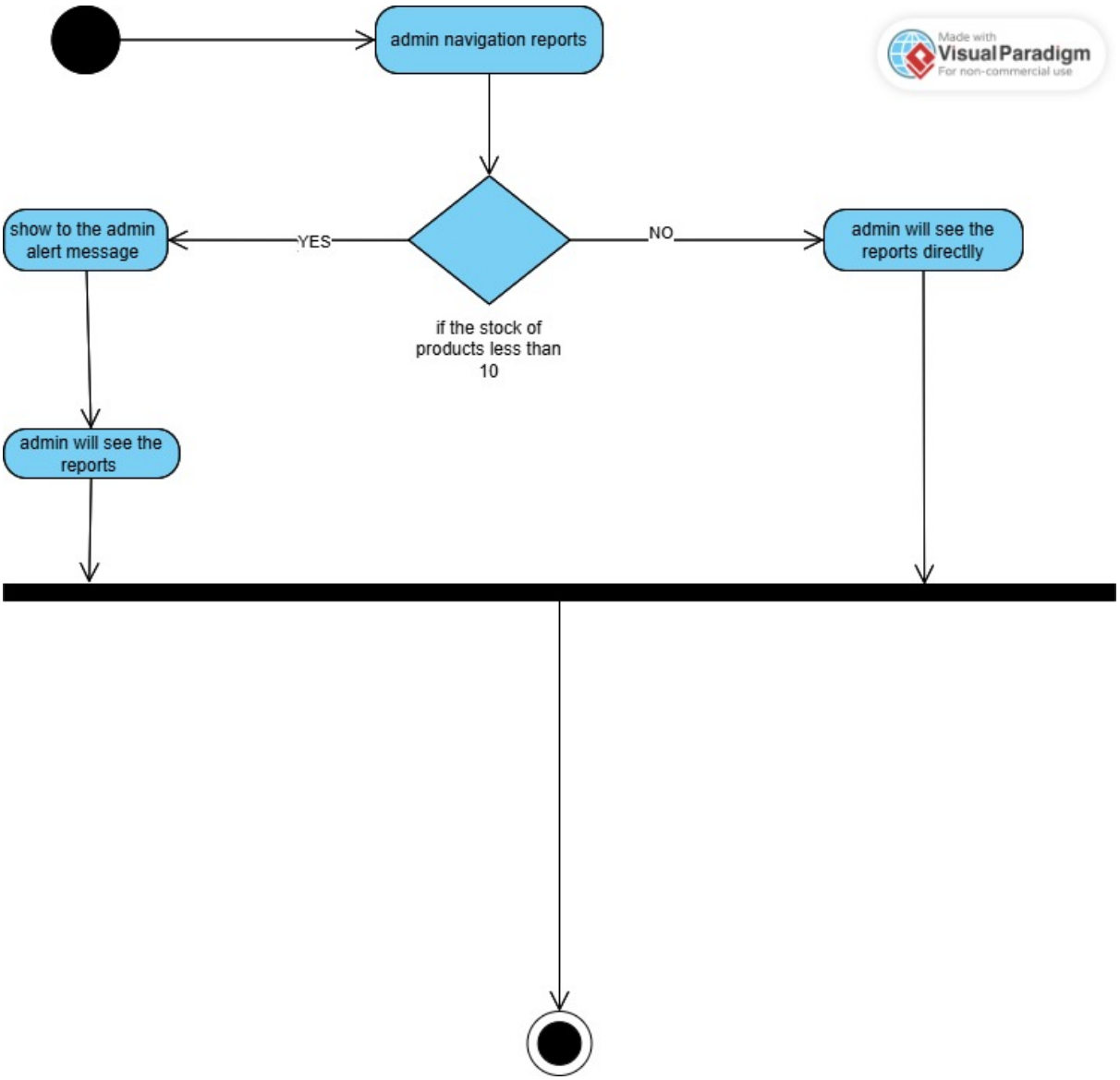
3.6.5 Admin Order Management Activity Diagram

Description: This activity diagram outlines the steps involved in managing orders, from creation to completion.



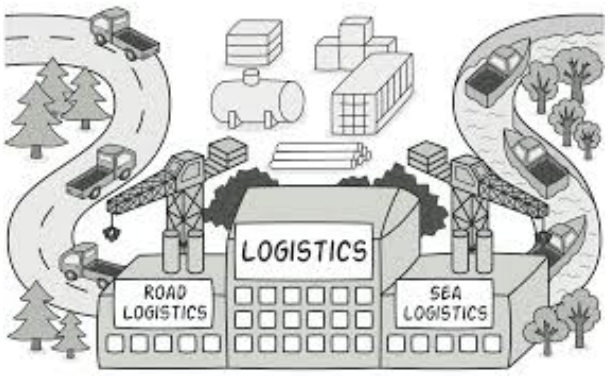
3.6.6 Admin View Reports Activity Diagram

Description: This diagram illustrates the process of generating and viewing reports within the system.



3.7 Pattern Use View Point

There are existing supermarket-style inventory management systems that offer similar functionalities, such as Talabat, and insta shop. These systems allow businesses to manage inventory, process sales, and track customer data. How- ever, FreshMart focuses on stock level tracking with real-time supplier integra- tion and automatic reordering tailored for smaller-scale supermarkets. Factory pattern: is a design pattern used to create objects without specifying the exact class of the object being created. Instead, it defines a method in a parent class (or interface) that is overridden by subclasses to decide which specific object to create.



- Factory :

The Factory class contains a static method, create, which is responsible for creating instances of different object types (User, Product, etc.) based on the provided parameters. The type parameter determines the type of object to be created, while params contains the specific details required for instantiation. User Class: Represents the users of the system with properties like name, email, and role. Contains methods for retrieving user details and saving them to the database. Factory Workflow: A client (e.g., admin or user) interacts with the factory through a simple interface. The create method abstracts away the complexity of object creation, ensuring that the right object is instantiated based on user input.

- The Observer:

Design pattern is a behavioral design pattern that facilitates a subscription mechanism, enabling multiple dependent components (observers) to receive real-time notifications about changes or significant events in a central object (subject). This design is especially effective in systems where multiple actions must be coordinated in response to changes in state. In our implementation, the Product class acts as the subject, monitoring stock levels and triggering notifications when specific thresholds are reached. Observers, such as notification services or dashboard components, are dynamically attached to the Product class. When the stock level falls below a defined threshold, these observers are automatically notified.

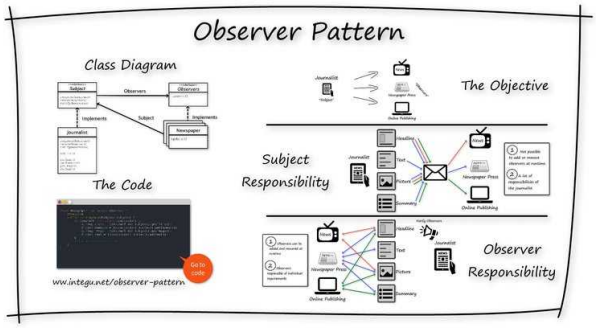


Figure 1: Observer Pattern

4 Data Design

4.1 UML Diagram

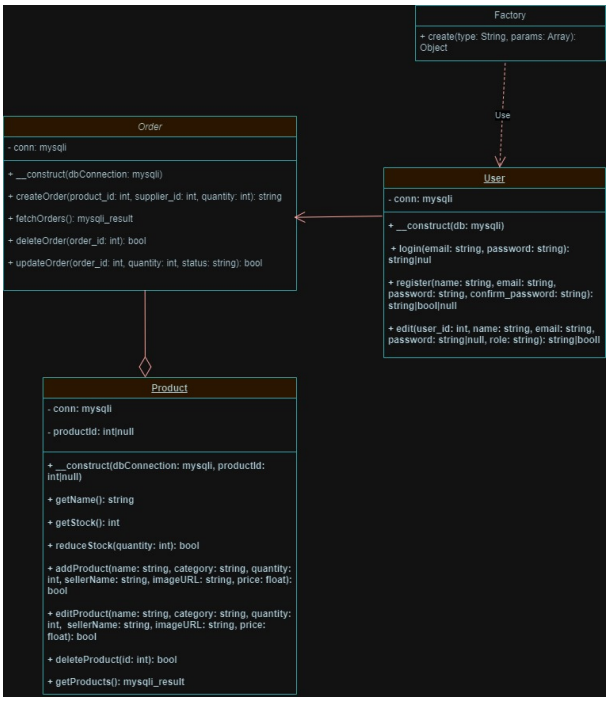


Figure 2: UML Diagram

Description: This UML diagram provides a clear representation of a basic FreshMart Inventory Management system's class structure, highlighting the interaction between orders, users, and products.

4.2 Relationships

- The Order Class: One or more products are "has" in an order. Every product may show up in a variety of orders. This indicates that an order (as part of its information) aggregates the products in the order

- The User Class: manages user authentication and registration, ensuring that only authorized users can place orders.

4.3 Data Description

The system will manage the following data entities:

- Users: ID, Name, Email, Password, UserType
- Products: ID, Name, Price, Quantity, Category
- Orders: ID, ProductID, UserID, Quantity, OrderDate, Status
- Suppliers: ID, Name, ContactInfo, PaymentTerms

4.4 Dataset Description

Each entity will have specific attributes:

- User: `ID (int, primary key), Name (varchar), Email (varchar, unique), Password (varchar), UserType (int)`
- Product: `ID (int, primary key), Name (varchar), Price (decimal), Quantity (int), Category (varchar)`
- Order: `ID (int, primary key), ProductID (int, foreign key), UserID (int, foreign key), Quantity (int), OrderDate (datetime), Status (varchar)`
- Supplier: `ID (int, primary key), Name (varchar), ContactInfo (varchar), PaymentTerms (varchar)`

4.5 Database Design Description

The database will be structured with the following tables:

- `users`
- `products`
- `orders`
- `suppliers`
- cart
- payments
- pages
- orderitems
- usertypes
- usertypes-pages
- navbar$_{b}uttons$

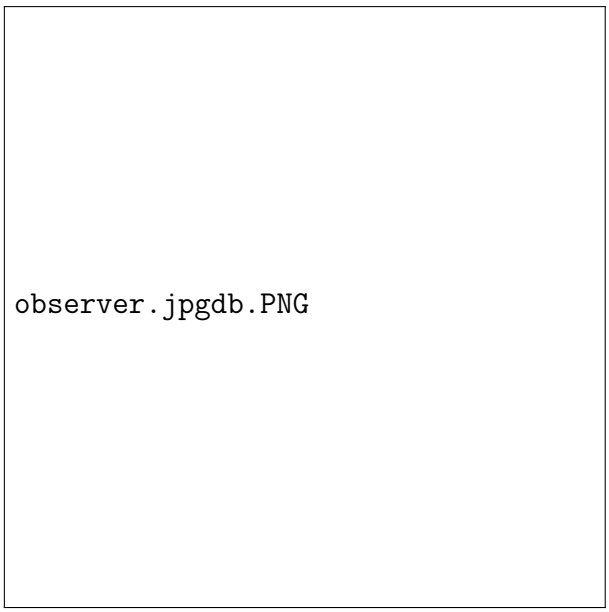


Figure 3: DataBase Diagram

5 Human Interface Design

5.1 User Interface

The User Interface (UI) allows regular users to easily browse products, place orders, and manage their accounts. It is designed for simplicity and ease of navigation, ensuring a smooth experience for customers and suppliers. Key components will include:

- Login Page
- Registration Page
- Homepage
- Product Listing Page
- Cart Page

5.2 Admin Interface

The Admin Interface provides administrators with tools to manage users, products, and orders. It offers powerful features for oversight and reporting, ensuring efficient control of the inventory system. Key components will include:

- Login Page
- Dashboard for Admins
- Navigation Bar Management Pages
- Product Management Pages
- Order Management Pages

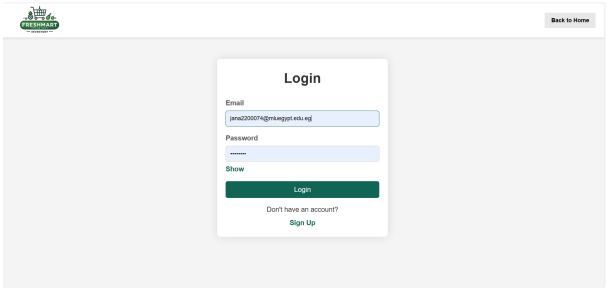


Figure 5: Login Page

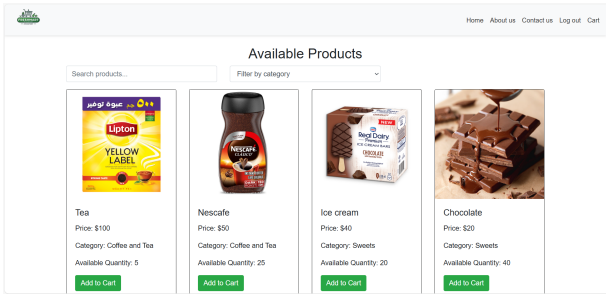


Figure 6: Products Page

- Supplier Management Pages
- Reports Page

5.3 Screen Images

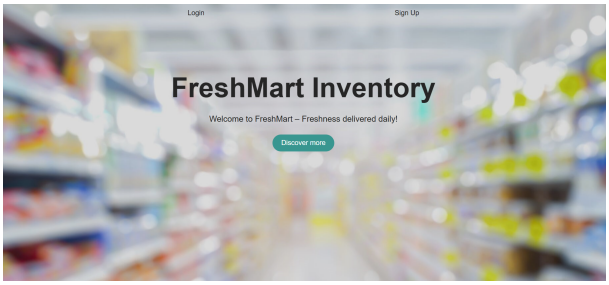


Figure 4: User Homepage

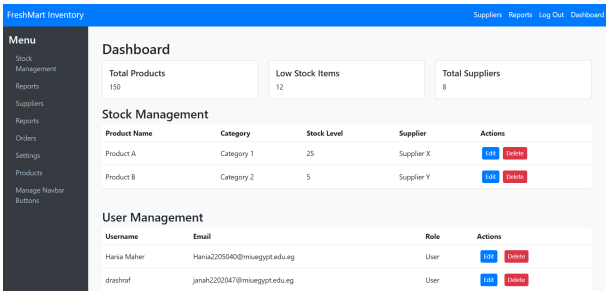


Figure 7: Admin Dashboard

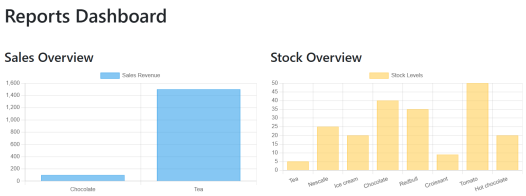


Figure 8: Reports

6 Requirements Matrix

Req. ID	Req. Desc	Class	Test Cases ID	Status
A01	Admin adds, edits, deletes, and views products	Product	TC01	In Progress
A02	Admin adds, edits, deletes, and views suppliers	Product	TC02	In Progress
A03	Admin manages user roles (Admin, Supplier, User)	User	TC03	In Progress
U01	User registers and logs in	User	TC04	Done
U02	User can view available products and add them to cart	Product	TC05	Done
U03	User can view scheduled orders in profile	Order	TC06	Done
U04	User can edit personal details in profile	User	TC07	Done
A04	Admin can log into account and log out	User	TC08	Done
A05	Admin can view sales and stock reports	Product	TC09	Done
A06	Admin can manage navbar items based on user roles	User	TC10	Done
A07	Admin can manage automatic reordering notifications	Product	TC11	Done
A08	Admin can manage user accounts (add, edit, delete)	User	TC12	Done
A09	Admin can view and manage supplier information	Product	TC13	Done
A10	Admin can manage product categories	Product	TC14	Done
A11	Admin can manage order statuses	Order	TC15	Done

Table 2: Requirements and Test Cases

6.1 Test Case: User Login

Test Name: Login Functionality Test **TC04**

Description: Validate that a user with valid credentials can successfully log in and navigate to the dashboard.

Preconditions:

- The user "Hania" is registered in the system with a username and password.
- The database contains valid login credentials for the user.

Steps:

1. Navigate to the login page of the inventory management system.
2. Enter the username **Hania**.
3. Enter the correct password.
4. Click the "Login" button.

Expected Result:

- The user is successfully authenticated.
- The system redirects the user to the dashboard.
- The dashboard displays relevant user details and product information.

```
PS C:\Users\smsm2\OneDrive\Desktop\xampp\htdocs\trial-project\software-project\software-project> ./vendor/bin/phpunit --bootstr
ap vendor/autoload.php tests/UserTest.php
PHPUnit 10.5.40 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.2.12

                                                                    1 / 1 (100%)

Time: 00:01.015, Memory: 6.00 MB

OK (1 test, 1 assertion)
PS C:\Users\smsm2\OneDrive\Desktop\xampp\htdocs\trial-project\software-project\software-project>
```

Figure 9: Unit Test Result for User Login

7 Appendices

7.1 GitHub Repository

`https://github.com/janamaklad/Management-Inventory`

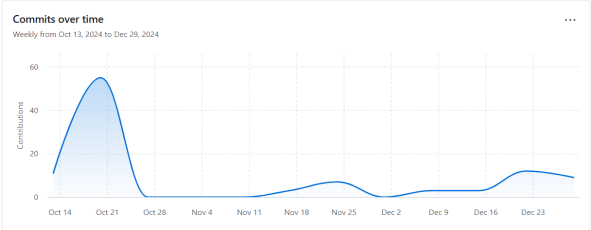


Figure 10: Commits