

**Zalabat**

***mart***

# Change Log

Version	Date	Author	Rationale
1.0	24/11/2024	Zeyad Waleed	
1.1	27/11/2024	Mohy Saleh	
1.2	4/12/2024	Zeyad Waleed	
1.2	11/12/2024	Mohy Saleh, Alhussain	

Version	Date	Author	Rationale
2.0	19/12/2024	Mohy	
2.1	21/12/2024	Mohy	

Version	Date	Author	Rationale
3.0	23/11/2024	Zeyad waleed, Mohy saleh	
3.1	24/12/2024	Alhussain shalaby, Zeyad Waleed	

# 1. Introduction

## 1.1 Purpose of the System

Zalabat is designed to streamline operations for supermarket employees by automating core tasks such as product management, employee management, sales tracking, and inventory control. .

## 1.2 Design Goals

### **Modularity**

Ensure that the system is divided into clear, maintainable modules.

### **Scalability**

Allow for future expansion of features without significant redesign.

### **Security**

Implement robust authentication and role-based access control.

### **Usability**

Provide an intuitive and responsive user interface.

### **Performance**

Optimize system responsiveness and minimize latency in operations.

# 2. High-level Software Architecture

## 2.1 Subsystem Decomposition

The system consists of the following subsystems:

### **Authentication Subsystem**

Handles user login and role-based access.

### **Product Management Subsystem**

Manages adding, removing, searching, and sorting products.

### **Sales Subsystem**

Handles sales transactions and generates invoices.

### **Employee Management Subsystem**

Manages employee records and contracts.

### **Reporting Subsystem**

Provides sales and inventory reports.

## 2.2 Hardware/Software Mapping

hosting and deployment

## 2.3 Persistent Data Management

Database Schema:

**Users:** Stores user credentials and roles.

**Products:** Stores product details (name, ID, price, stock quantity).

**Employees:** Stores employee data (name, role, contract start/end dates).

**Sales:** Tracks sales transactions and associated invoices.

## 2.4 Access Control and Security

Role-Based Access Control:

**Admin:** Full access to all features.

**Employee:** Limited access to sales and product viewing.

**Authentication:** Login using secure password.

## 2.5 Boundary Conditions

**System Startup:** Initialize all subsystems and establish database connections.

**Error Handling:** Display an error messages for database failures, and invalid input.

**Shutdown:** Ensure all active sessions are terminated.

# 3. Low-level Design

## 3.1 Object Design Trade-offs

**Database Normalization:** Used for efficient querying but denormalized selective tables for faster reporting.

**Frontend Framework:** Chose React.js for its component-based architecture and state management.

## 3.2 Final Object Design

### Key Classes:

**User:** Handles authentication and role assignment.

**Product:** Encapsulates product details and inventory operations.

**Employee:** Manages employee records and validation.

**Sale:** Manages transaction details and invoice generation.

### 3.3 Design Patterns

**Singleton:** Used for the Catalog, since we want only one catalog object carrying all products.

**Factory Pattern:** Used for creating user objects (Admin or Employee) based on roles.

**MVC Pattern:** Adopted for separating concerns in the application.

**DTO:** Used for sending/Receiving Data between Server and Client sides

# Business Domain Rules

ID	Rule	Changeability	Source
1	All users must authenticate using their username and password before accessing the system.	Low	System Requirements
2	The system must log user activities for auditing purposes.	Medium	Security Policy
3	Each product must have a unique ID, name, category, and stock level.	Low	Product Management Policy
4	Products can only be added or removed by users with the role of "Admin."	Medium	User Role Requirements
5	The stock level must be updated automatically when sales are recorded.	Low	System Functional Design
6	Only products with a stock level greater than zero can be added to an invoice.	Medium	Sales Processing Rules
7	Employee data can only be managed by users with the role of "Admin."	Medium	Employee Management Policy
8	The system must lock user accounts after 5 unsuccessful login attempts.	Low	Security Policy
9	Reports must include total sales, best-selling products, and stock levels.	High	Reporting Requirements
10	"Medicines" category must include an expiry date for each product.	Low	Domain-Specific Standards
11	Products nearing expiry (30 days or less) must trigger an alert to the admin.	Medium	Compliance Standards
12	Discounts for groceries and detergents must follow predefined promotional rules.	High	Marketing Guidelines
13	Passwords must be stored in an encrypted format.	Low	Security Best Practices
14	Refunds can only be processed for purchases made within 30 days.	Medium	Financial Policy

## MD Code

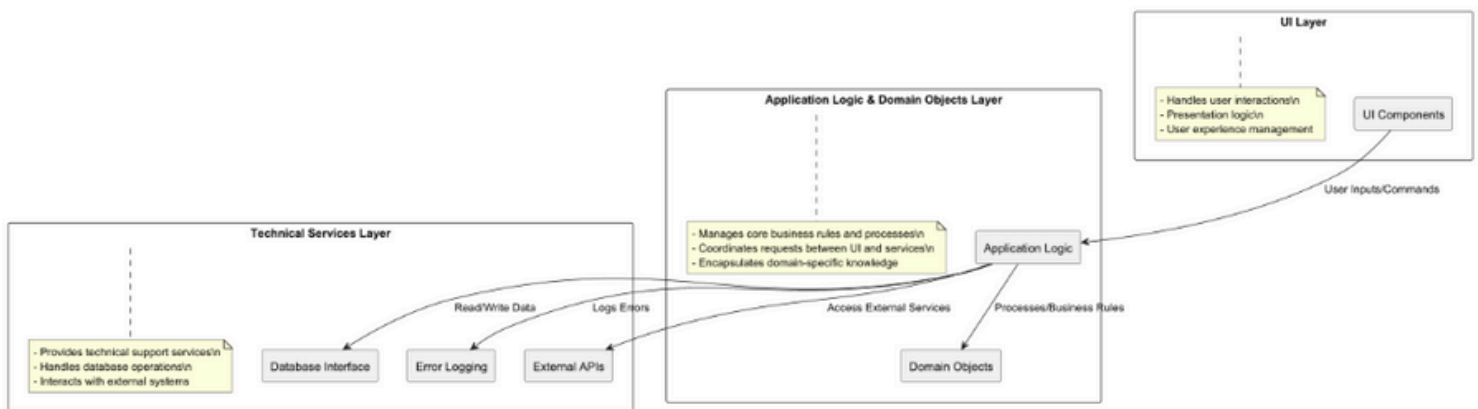
**ID**	**Rule**	**Changeability**	**Source**
1	All users must authenticate using their username and password before accessing the system.	Low	System Requirements
2	The system must log user activities for auditing purposes.	Medium	Security Policy
3	Each product must have a unique ID, name, category, and stock level.	Low	Product Management Policy
4	Products can only be added or removed by users with the role of "Admin."	Medium	User Role Requirements
5	The stock level must be updated automatically when sales are recorded.	Low	System Functional Design
6	Only products with a stock level greater than zero can be added to an invoice.	Medium	Sales Processing Rules
7	Employee data can only be managed by users with the role of "Admin."	Medium	Employee Management Policy
8	The system must lock user accounts after 5 unsuccessful login attempts.	Low	Security Policy
9	Reports must include total sales, best-selling products, and stock levels.	High	Reporting Requirements
10	"Medicines" category must include an expiry date for each product.	Low	Domain-Specific Standards
11	Products nearing expiry (30 days or less) must trigger an alert to the admin.	Medium	Compliance Standards
12	Discounts for groceries and detergents must follow predefined promotional rules.	High	Marketing Guidelines
13	Passwords must be stored in an encrypted format.	Low	Security Best Practices
14	Refunds can only be processed for purchases made within 30 days.	Medium	Financial Policy



# The Logical Architecture

## Layers include:

- **UI Layer:** Handles user interactions, presentation logic, and overall user experience.
- **Application Logic & Domain Objects Layer:** Encapsulates core business rules and processes, coordinating communication between the UI and underlying technical services.
- **Technical Services Layer:** Provides essential support services such as database management, error logging, and external system integration.



# The Logical Architecture

## PlantUML Code

```
@startuml
skinparam packageStyle rectangle
skinparam backgroundColor #FFFFFF

title Logical Architecture

package "Logical Architecture" {

    package "UI Layer" as UILayer {
        rectangle "UI Components" as UIComponents
        note bottom of UILayer
        - Handles user interactions\n
        - Presentation logic\n
        - User experience management
        end note
    }

    package "Application Logic & Domain Objects Layer" as AppLayer {
        rectangle "Application Logic" as AppLogic
        rectangle "Domain Objects" as DomainObjects
        note bottom of AppLayer
        - Manages core business rules and processes\n
        - Coordinates requests between UI and services\n
        - Encapsulates domain-specific knowledge
        end note
    }

    package "Technical Services Layer" as TechLayer {
        rectangle "Database Interface" as DBInterface
        rectangle "Error Logging" as ErrorLogging
        rectangle "External APIs" as ExternalAPIs
        note bottom of TechLayer
        - Provides technical support services\n
        - Handles database operations\n
        - Interacts with external systems
        end note
    }
}

UIComponents --> AppLogic : User Inputs/Commands
AppLogic --> DomainObjects : Processes/Business Rules
AppLogic --> DBInterface : Read/Write Data
AppLogic --> ErrorLogging : Logs Errors
AppLogic --> ExternalAPIs : Access External Services

@enduml
```

Field	Description
Use Case ID	UC-001
Use Case Name	Manage Products
Actors	Admin
Goal	Manage Products
Preconditions	admin must be authenticated.
Basic Flow	<ol style="list-style-type: none"> <li>1. Admin selects "Manage Products"</li> <li>2. Admin enters product details (name, category, price, etc.).</li> <li>3. System validates inputs</li> <li>4. System saves product information.</li> </ol>
Altirnative Flow	Product was saved before system writs to the users "products is already in the catalog"
Postconditions	Product Info Changed/Created/Deleted

Field	Description
Use Case ID	UC-002
Use Case Name	Record Sales Invoice
Actors	Admin, Employee
Goal	Employee able to choose products requested by user, and proceed to a successful sale that is then recorded.
Preconditions	User must be authenticated. User must be Employee. Cashier should have been maintained
Basic Flow	<ol style="list-style-type: none"> <li>1.Employee selects "Record Sales Invoice.</li> <li>2.Employee selects products and enters quantities.</li> <li>3.System calculates total cost.</li> <li>4.System updates product inventory.</li> <li>5.System generates and saves the invoice.</li> </ol>
Altirnative Flow	If product stock is insufficient, the system displays an error message. Employee adjusts quantities or selects other products if possible.
Postconditions	Sales invoice is saved, product stock is updated, and customer purchase information is recorded.

Field	Description
Use Case ID	UC-003
Use Case Name	Add Employee
Actors	Admin, Employee
Goal	Ability to add Employees so that they can use System
Preconditions	Admin is user, Store Devices available.
Basic Flow	<ol style="list-style-type: none"> <li>1.Admin Select "Add Employee"</li> <li>2.Admin sets up Employee Details (Name, Email, join date, etc.).</li> <li>3.System checks details validity (email not entered before, etc.)</li> </ol>
Altirnative Flow	If email is already present, System doesn't create new record, redirect again to the add employee form.
Postconditions	Employee details stored, info like Date of joining, and number purchases initialised.

Field	Description
Use Case ID	UC-004
Use Case Name	View sales details
Actors	Admin
Goal	Admin views metrics such as total sales, profit summaries, and recent transactions.
Preconditions	Admin is authenticated
Basic Flow	<ol style="list-style-type: none"> <li>1.Admin selects "View Profit Dashboard.</li> <li>2.System fetches relevant data on sales, profits, and transaction history.</li> <li>3.System displays summarized metrics and visual reports.</li> </ol>
Altirnative Flow	Data Retrieval Failure: If the system cannot retrieve data (e.g., due to network or database issues), it displays an error message. Admin can retry after a delay or investigate the issue.
Postconditions	Dashboard data is displayed, reflecting the latest sales and profit statistics.

# Non Functional Requirements

## Usability

Simple, role-based user interface for admins and employees.

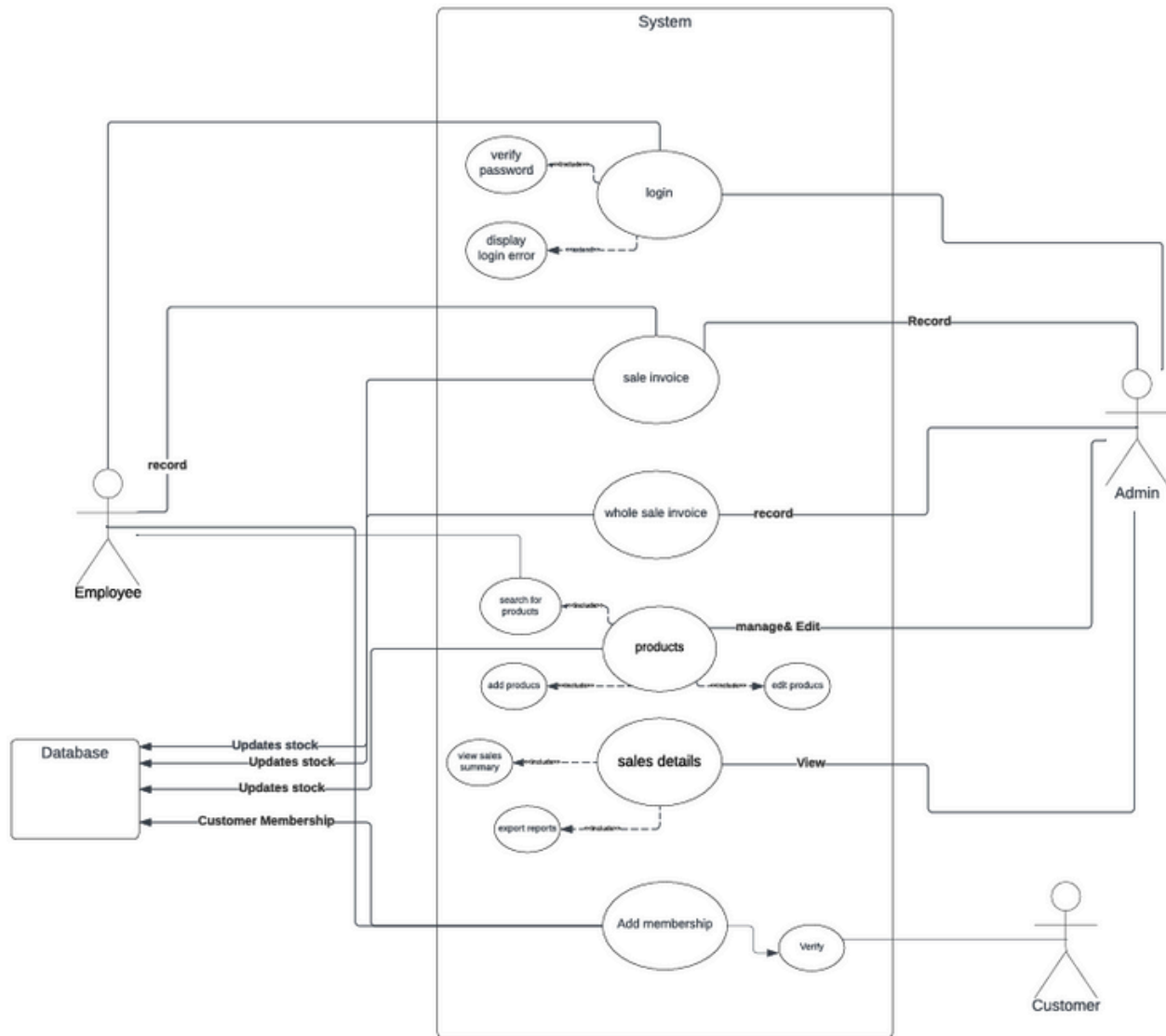
## Data Persistence

All user, product, and transaction data must be retained for future sessions, ensuring consistency.

## Maintainability

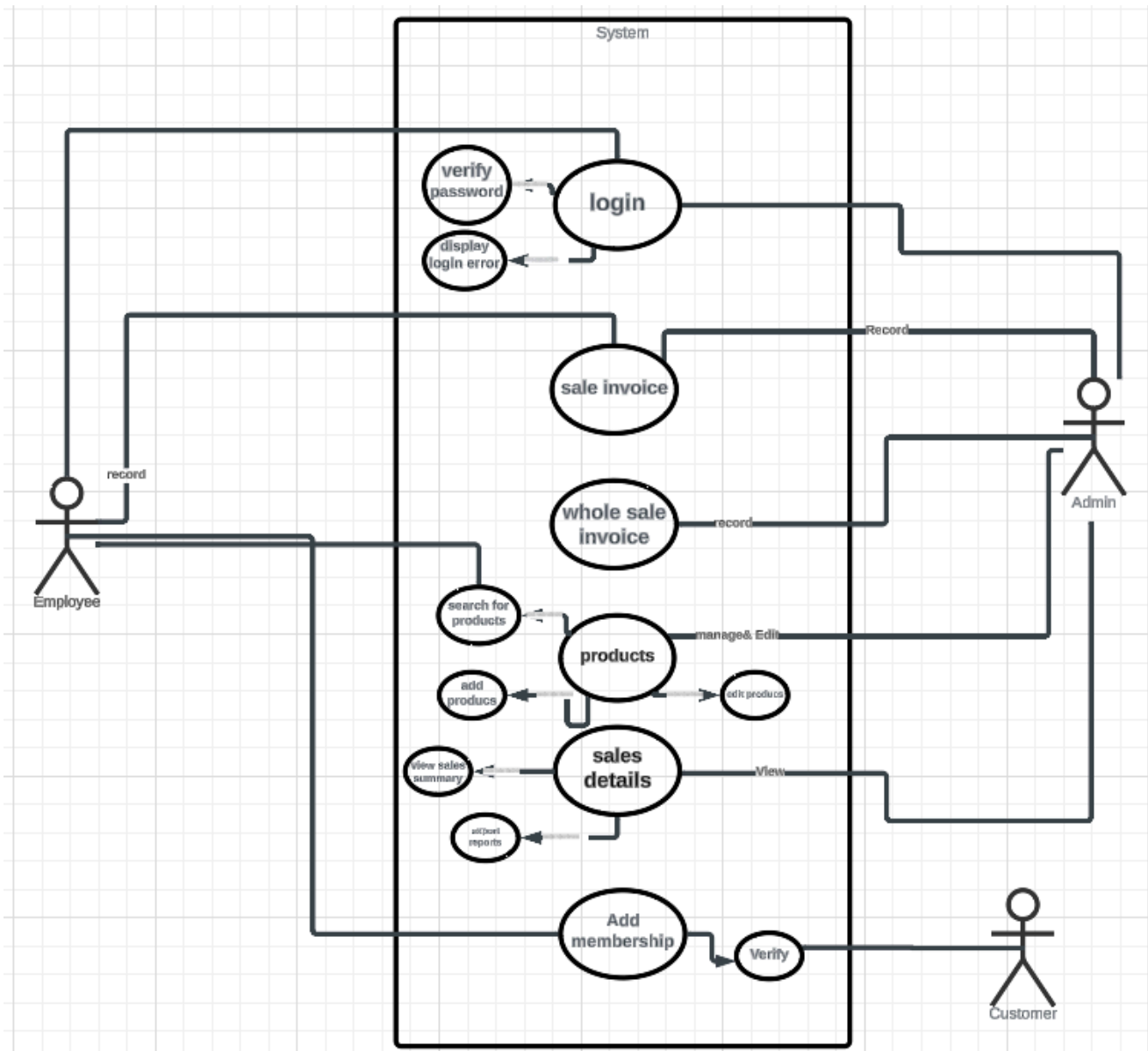
Easy to Change and maintain code, allowing for easy integration of new features.

# UC Diagram V1

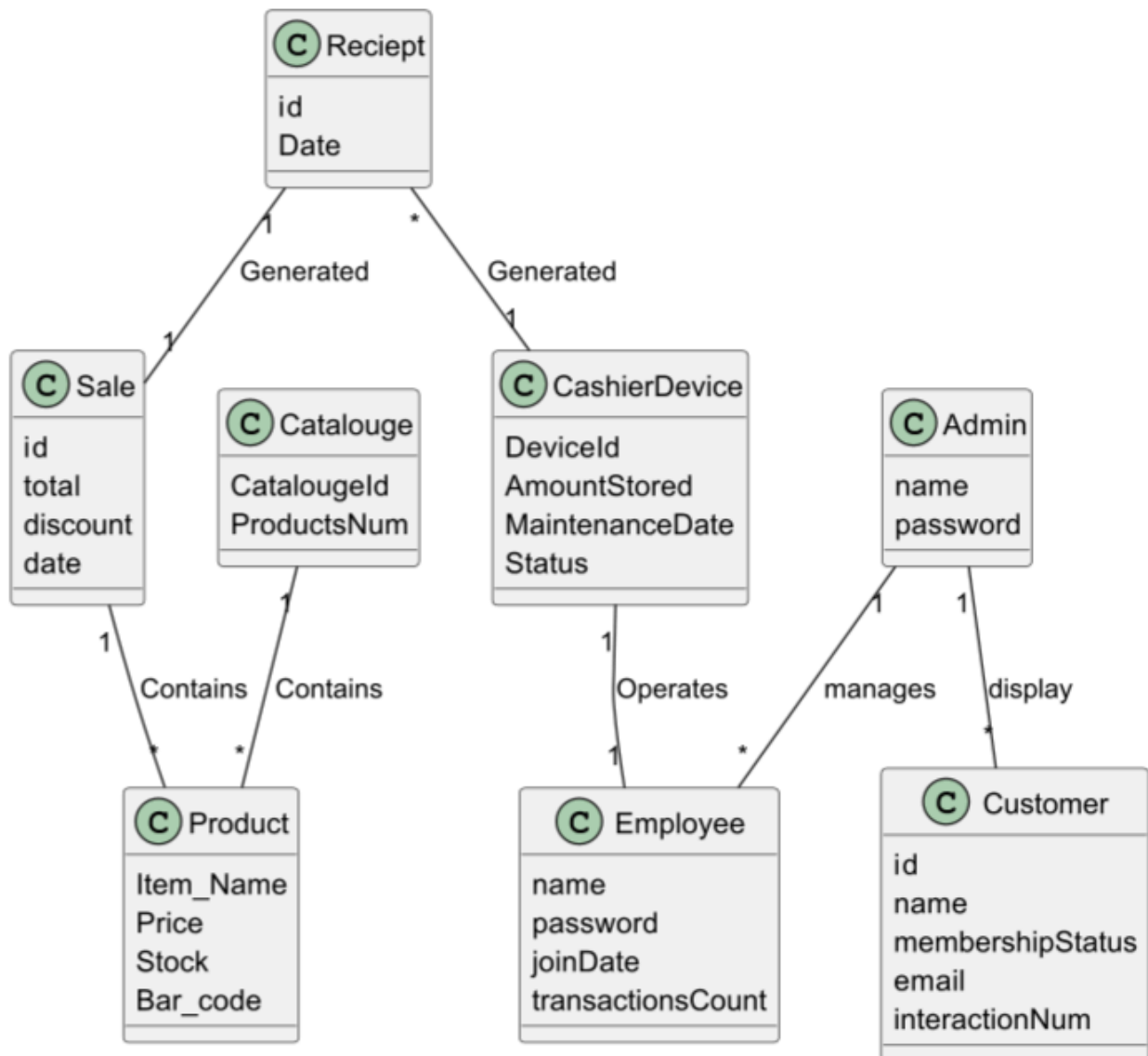




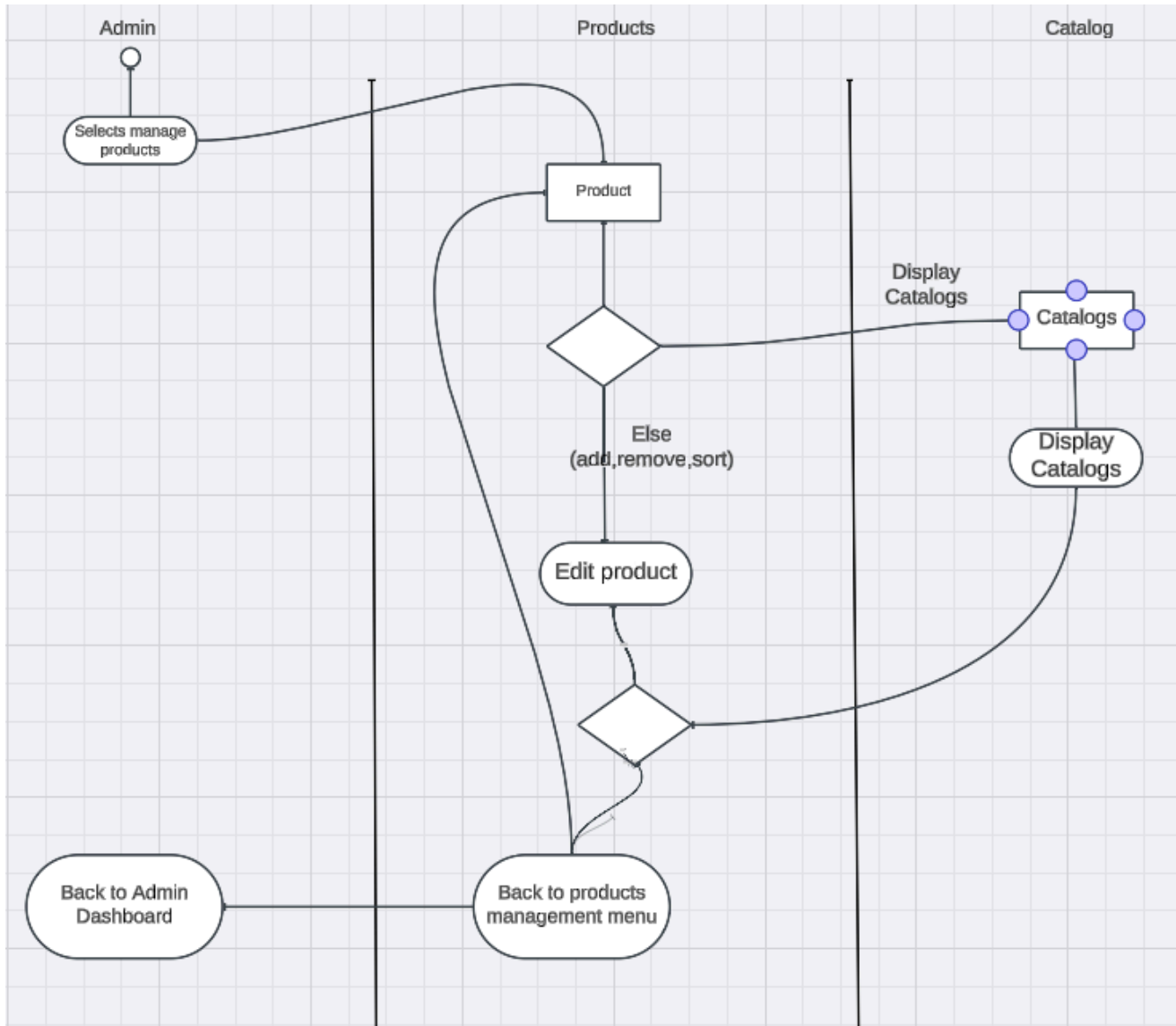
# UC Diagram V2



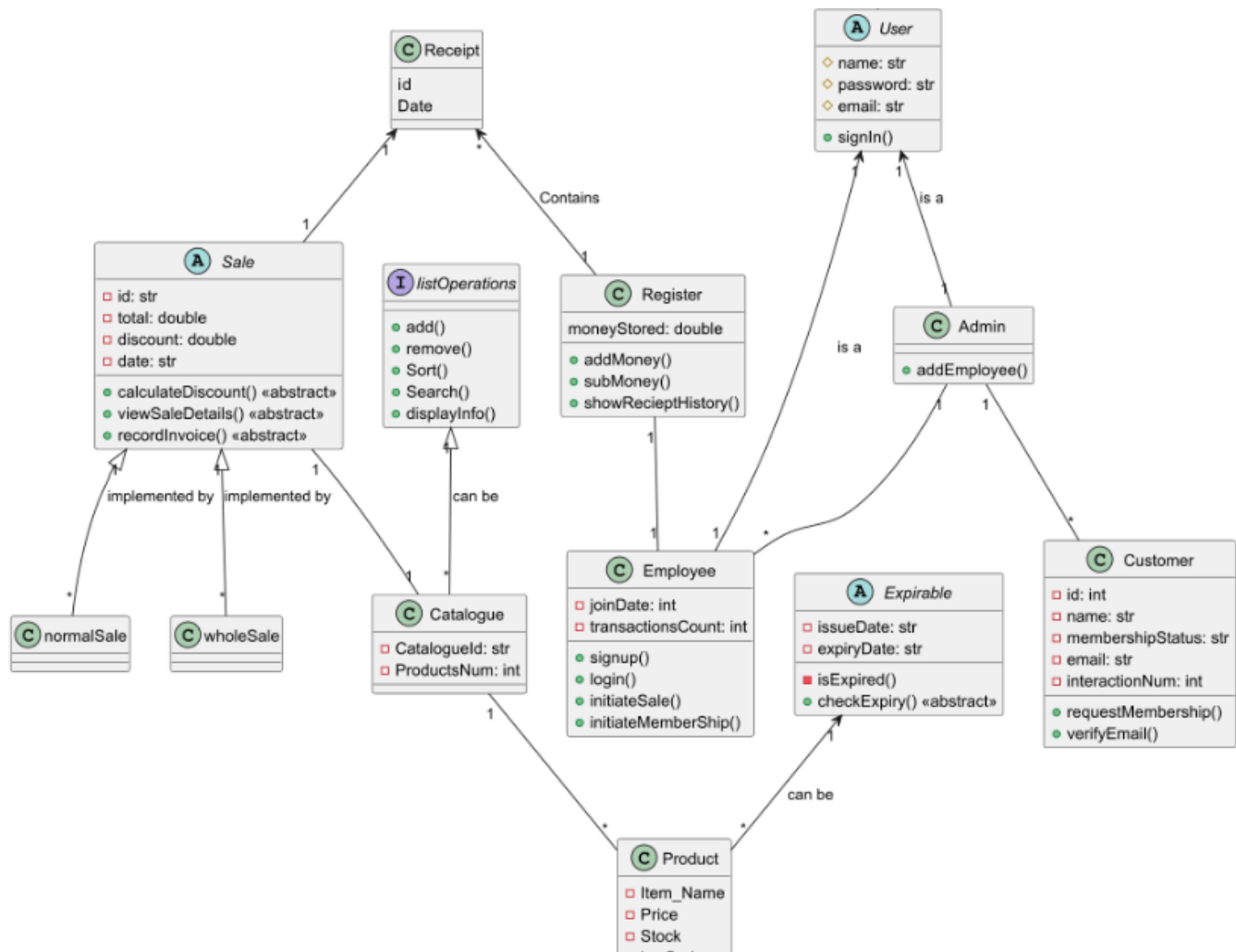
# Domain model



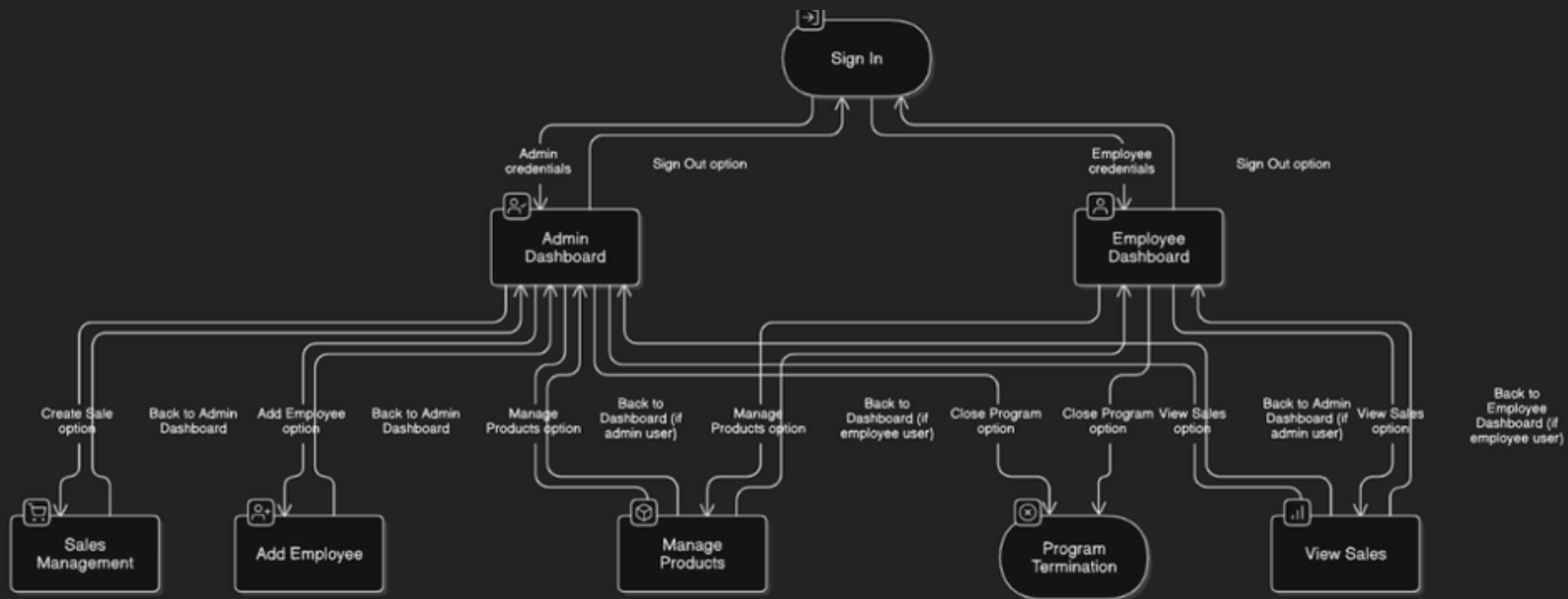
# Activity diagram



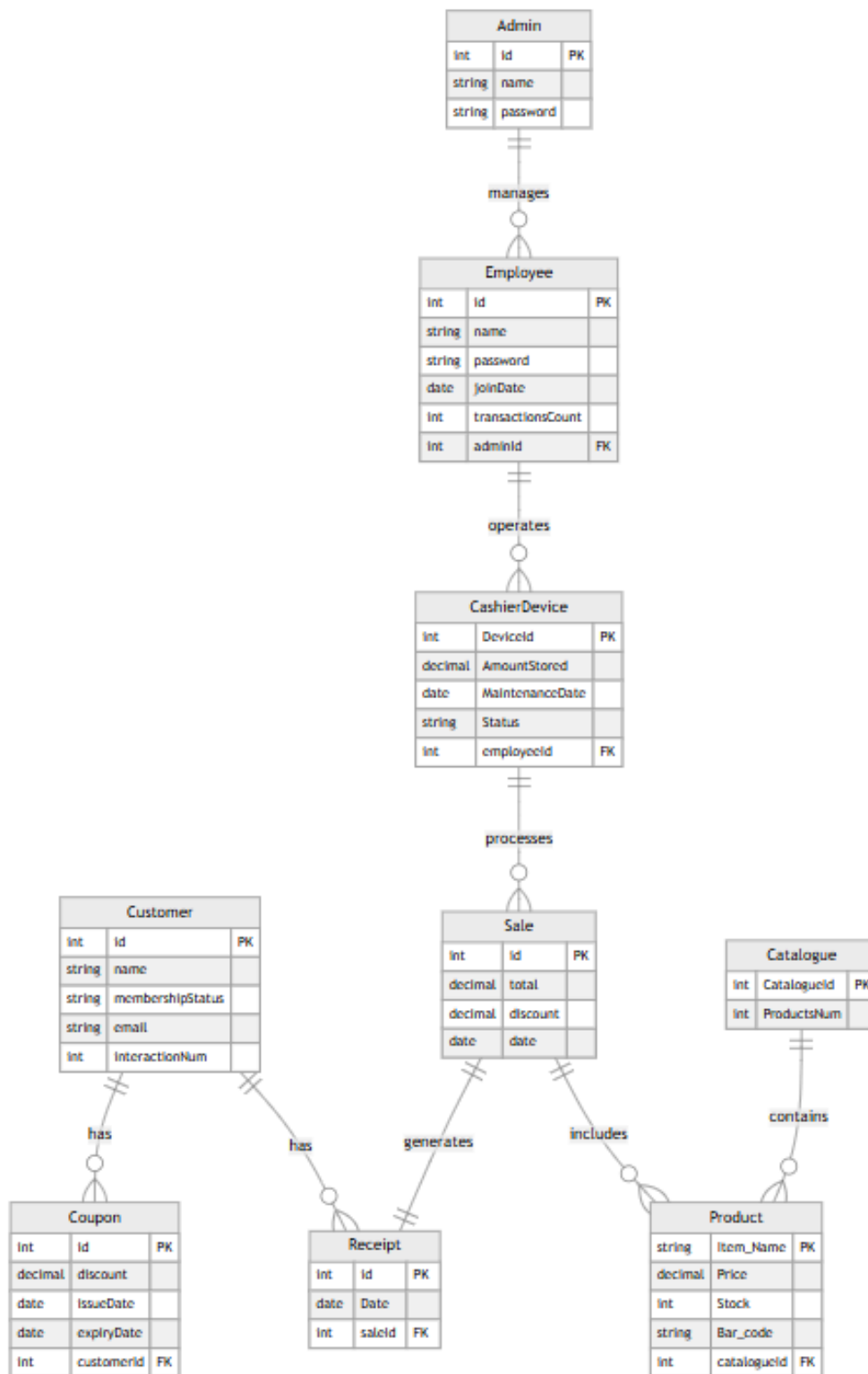
# Class diagram



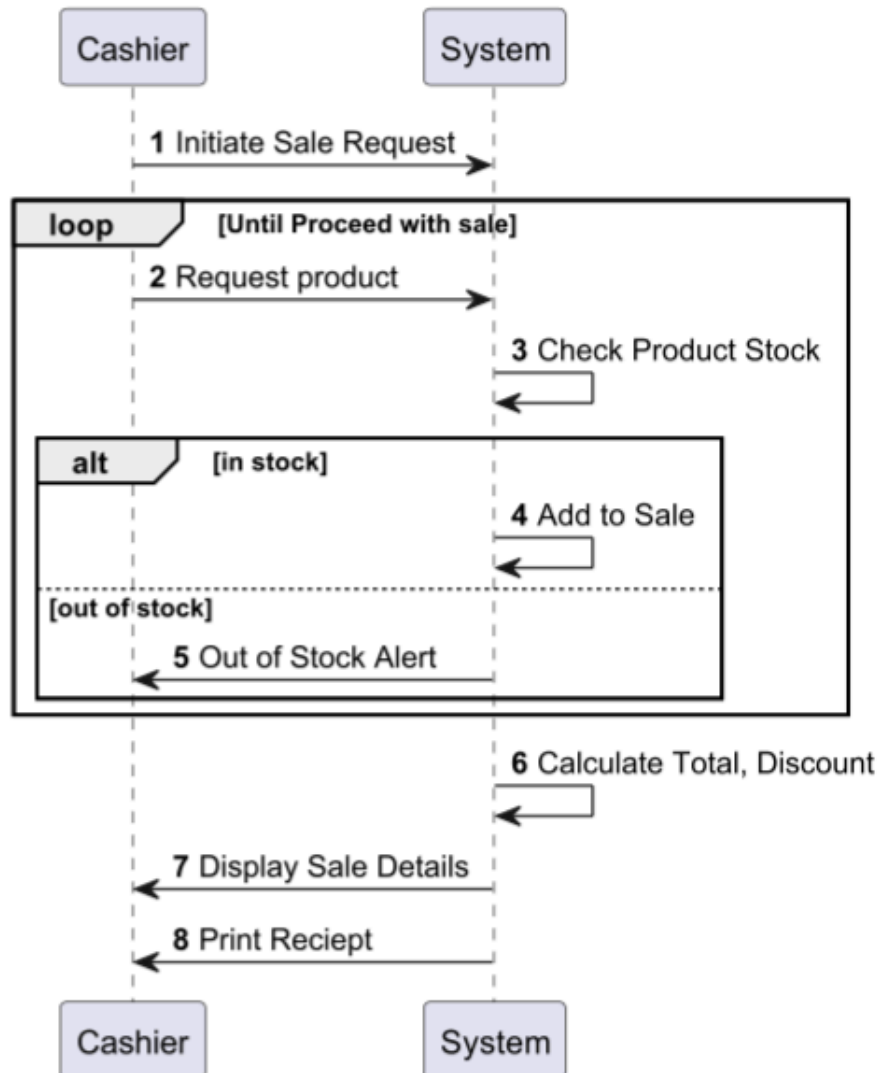
# State digram



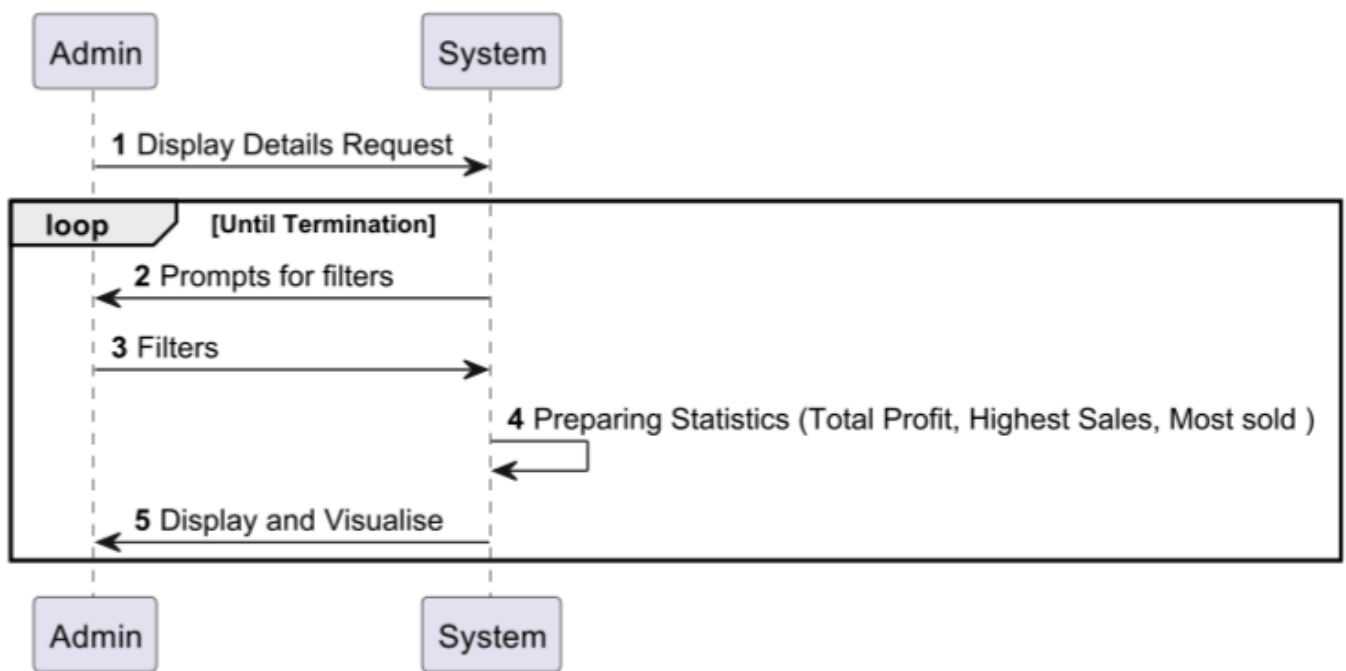
# ERD



# System Sequence Diagram

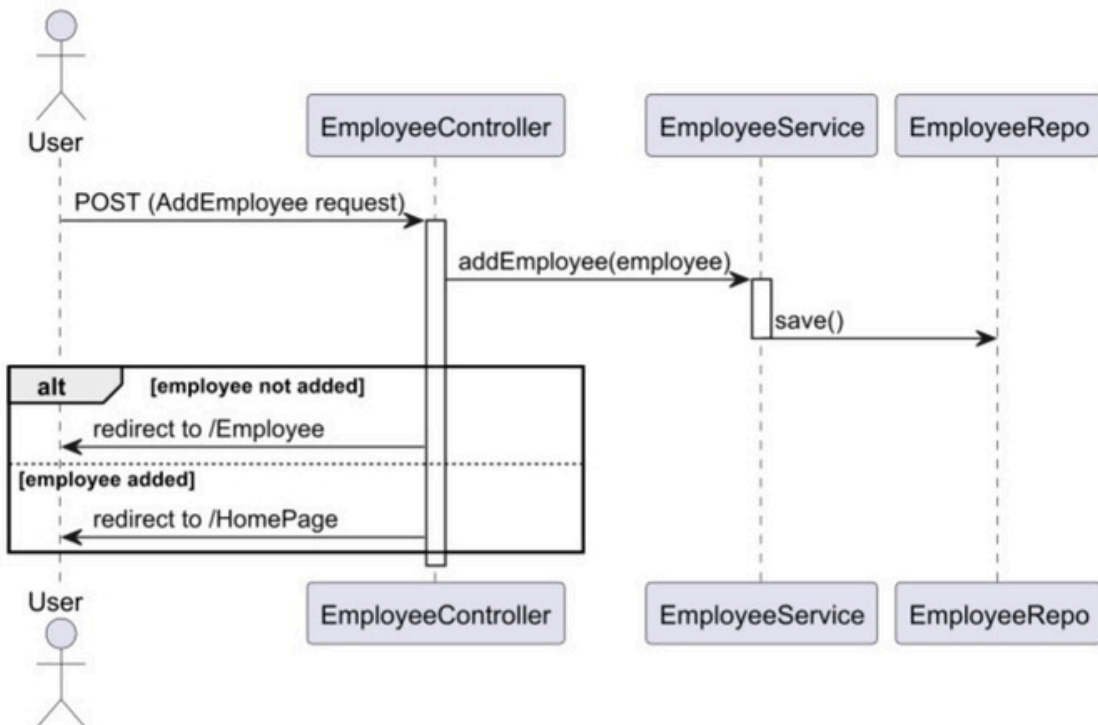
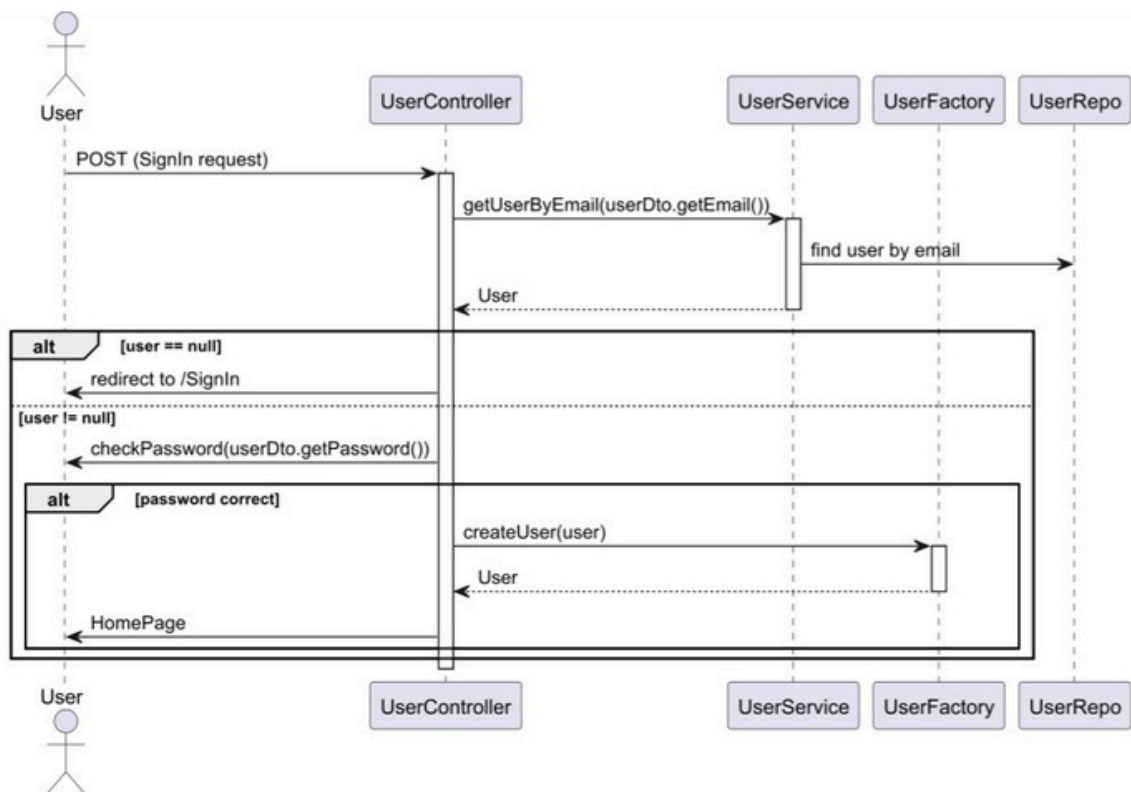


# System Sequence Diagram

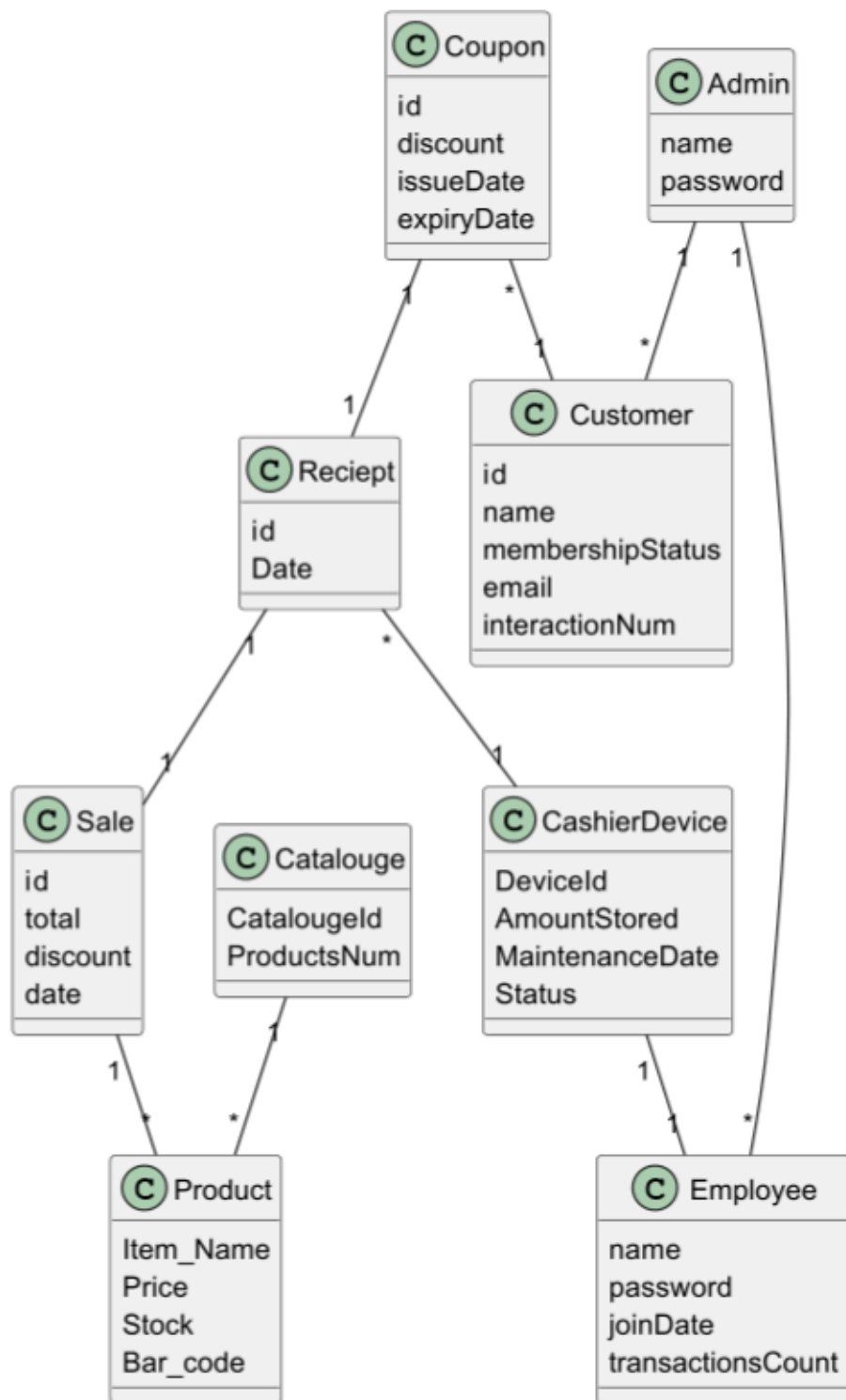




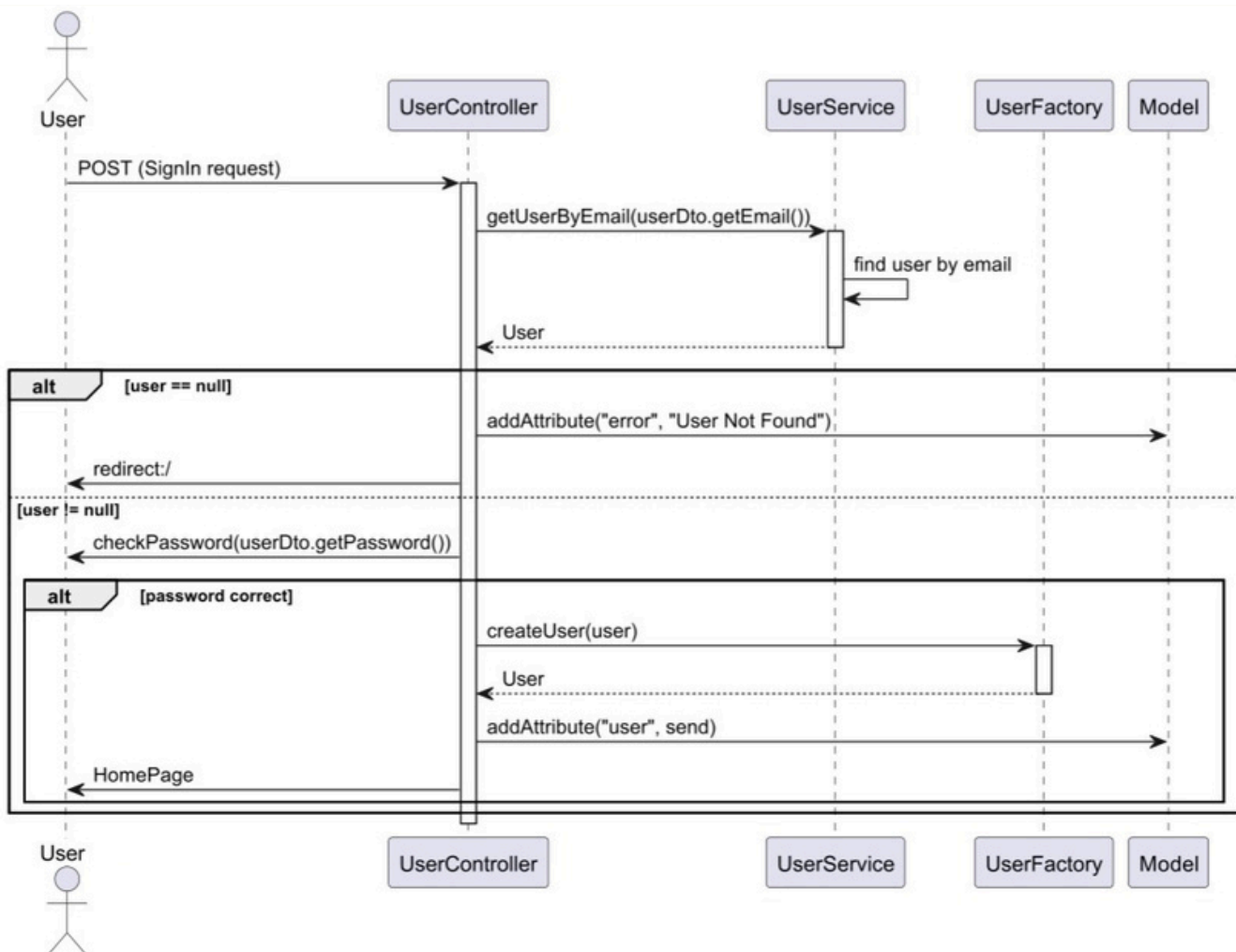
# Sequence Diagram



# Domain Model



# Domain Model



# GUI Snapshots

Zalabat **mart**



Create Sale



Add employee



View Sales Details

© 2024 Zalabat Mart. All rights reserved.

Zalabat **mart**

## Welcome Back 🙌

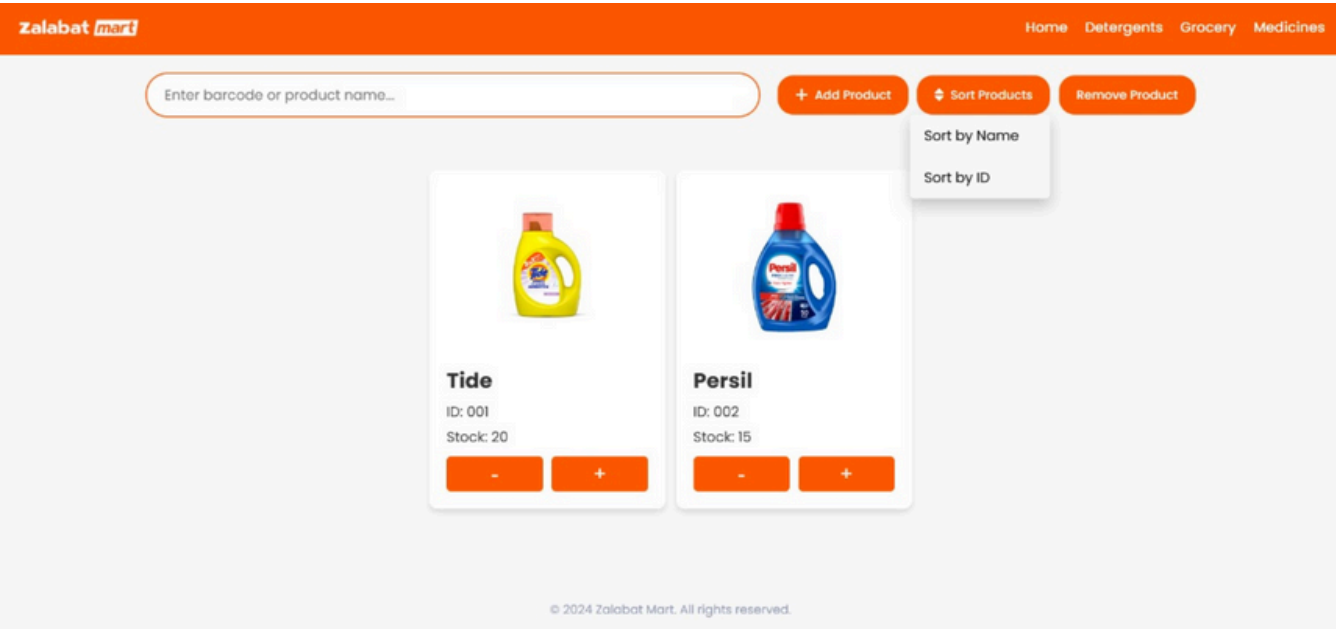
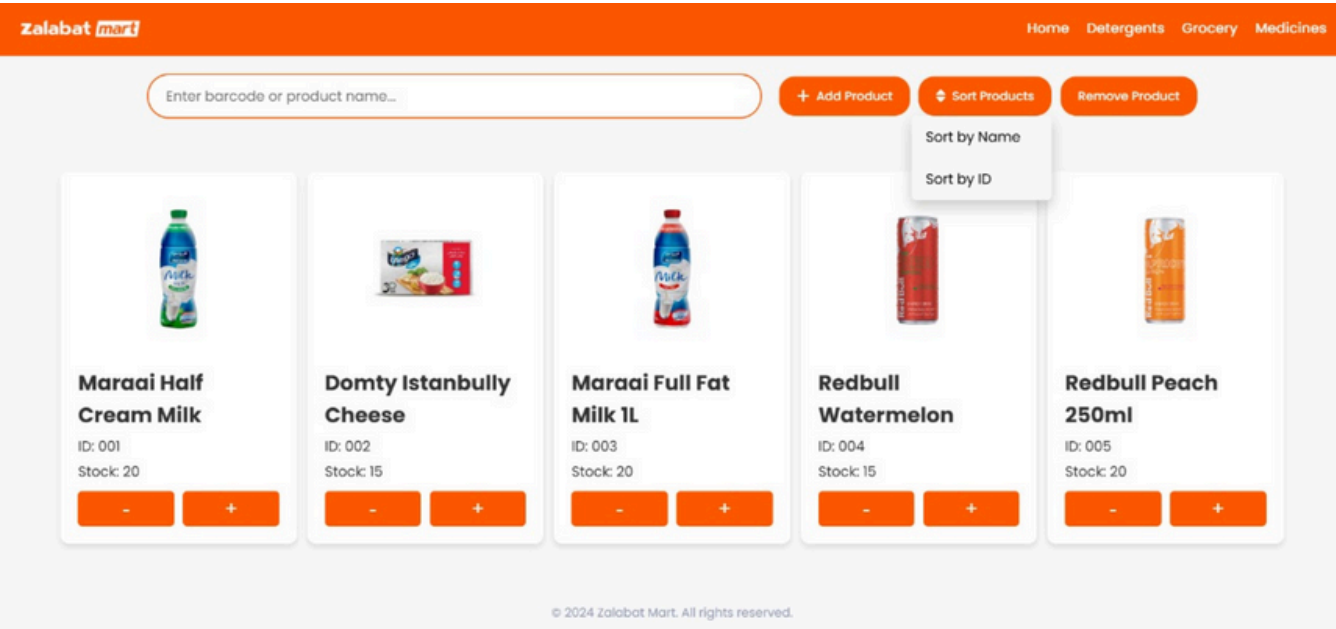
Today is a new day. It's your day. You shape it. login to start managing your projects.

Login

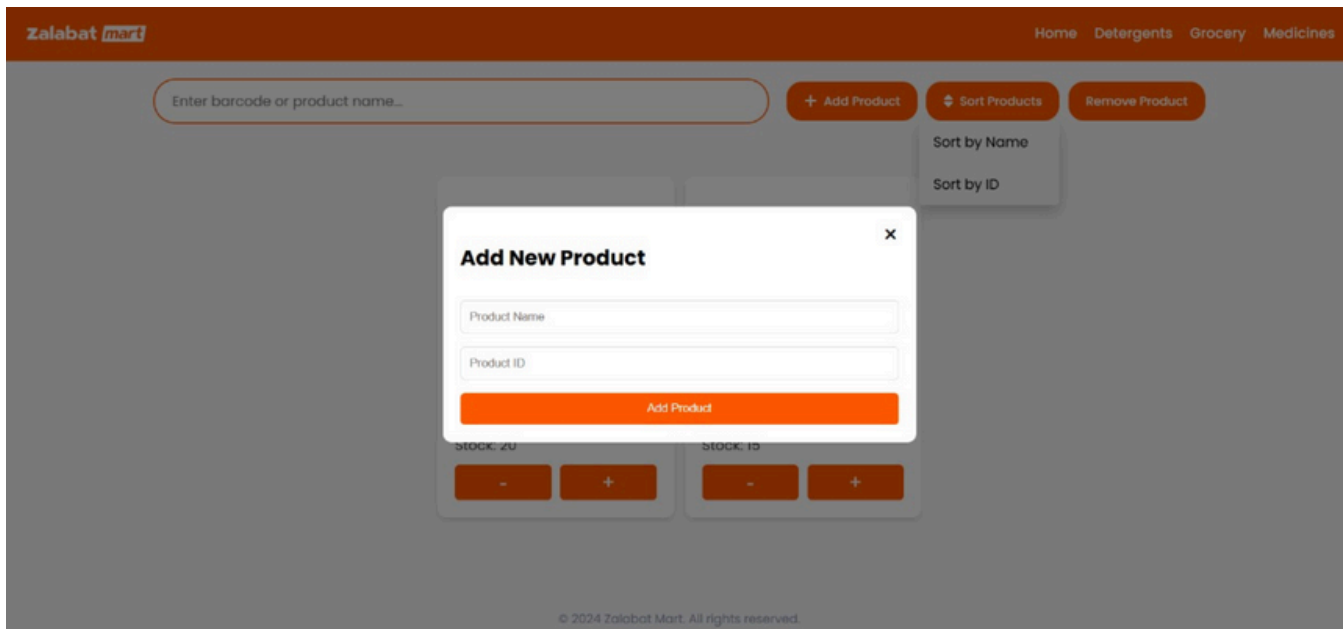


© 2024 Zalabat Mart. All rights reserved.

# GUI Snapshots



# GUI Snapshots



# Glossary

## Revision History

Version	Date	Description	Author
1.0	24/11/2024	Revised	Alhussain Shalaby
1.1	27/11/2024	Revised	Mohy Saleh
1.2	4/12/2024	Revised	Zeyad Waleed

## Terms and Definitions

Term	Defention	Format	Validation Rules
1.0			Zeyad Waleed
1.1			Mohy Saleh
1.2			Alhussain Shalaby

**Zalabat** ***mart***