# Report 2: System Analysis & Architectural Design Report (Design Phase)

## System Overview

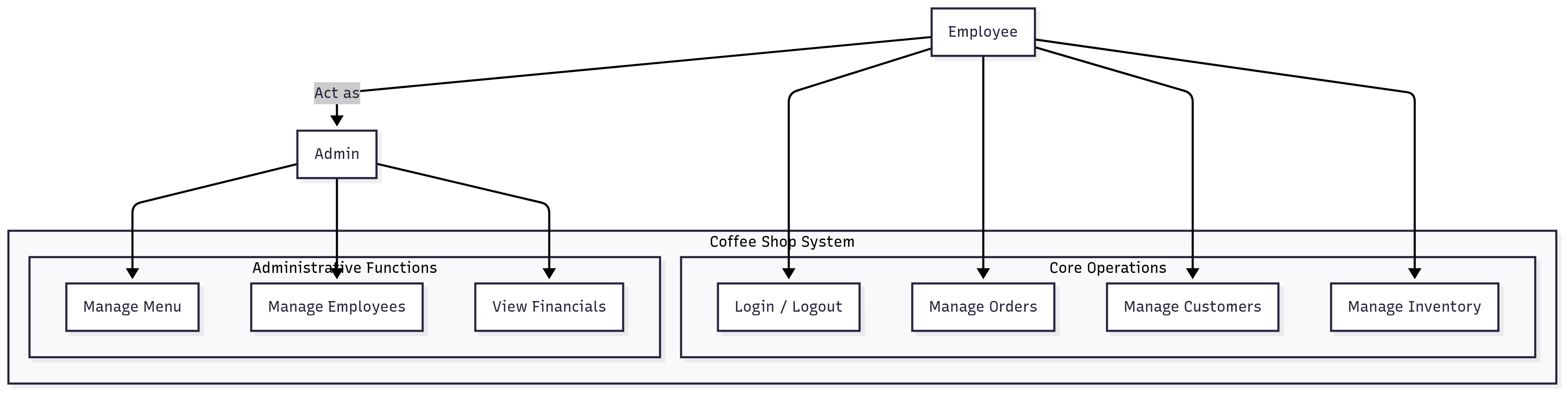
The Coffee Shop Chain Management System is a desktop application developed using C# and .NET 8. It aims to streamline and automate the daily operations of a coffee shop. The system features a user-friendly Windows Forms (WinForms) interface for staff and administrators, and a Console App for backend tasks like data imports or maintenance.

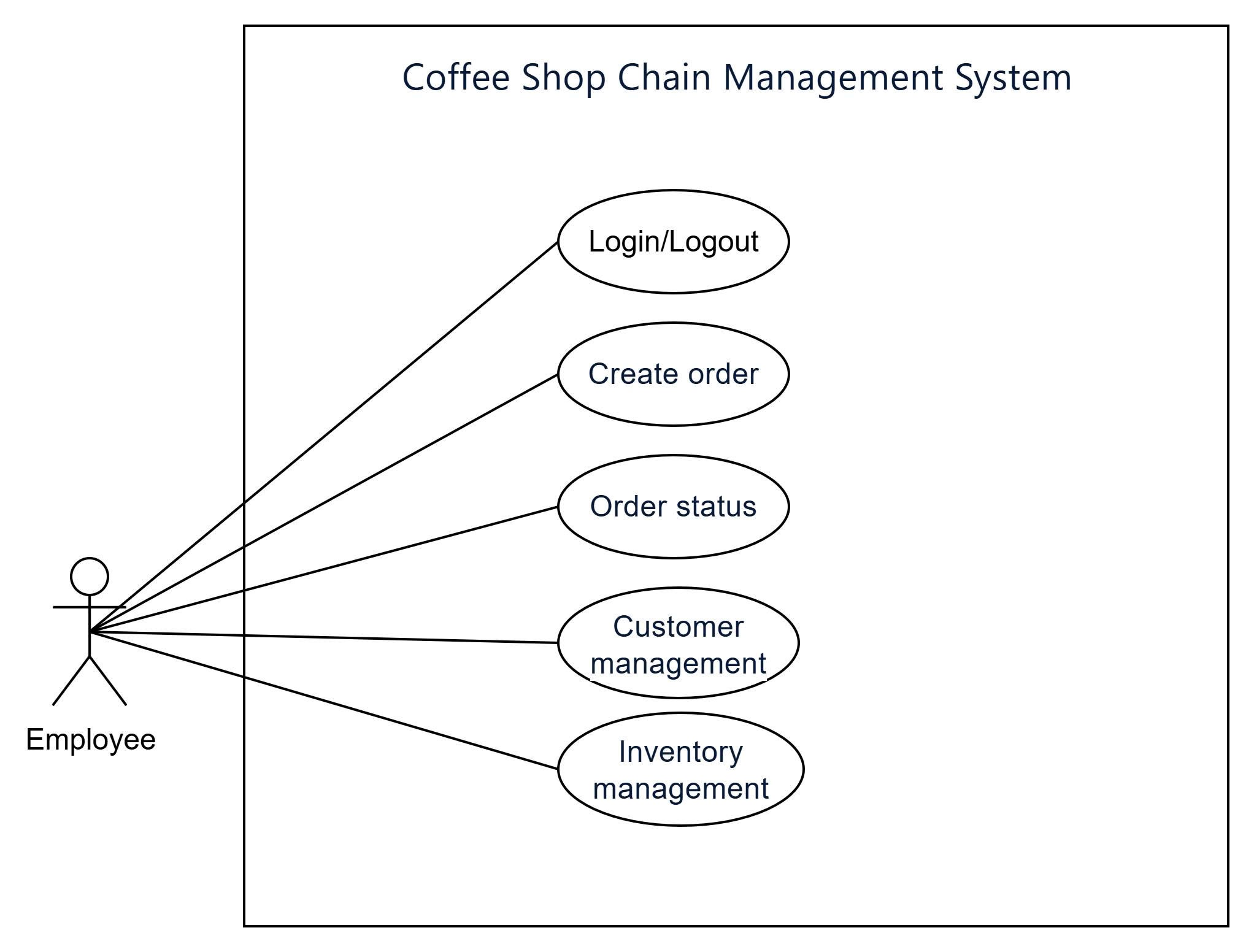
* Intended Users: The system is designed for two main roles:
  + Admin/Staff: Manage all operational aspects, including inventory, menu items, staff, customers, and sales reporting.
  + Customers: View the menu, place orders, and manage their loyalty program status.
* Target Platform: The application is built for the Windows operating system.

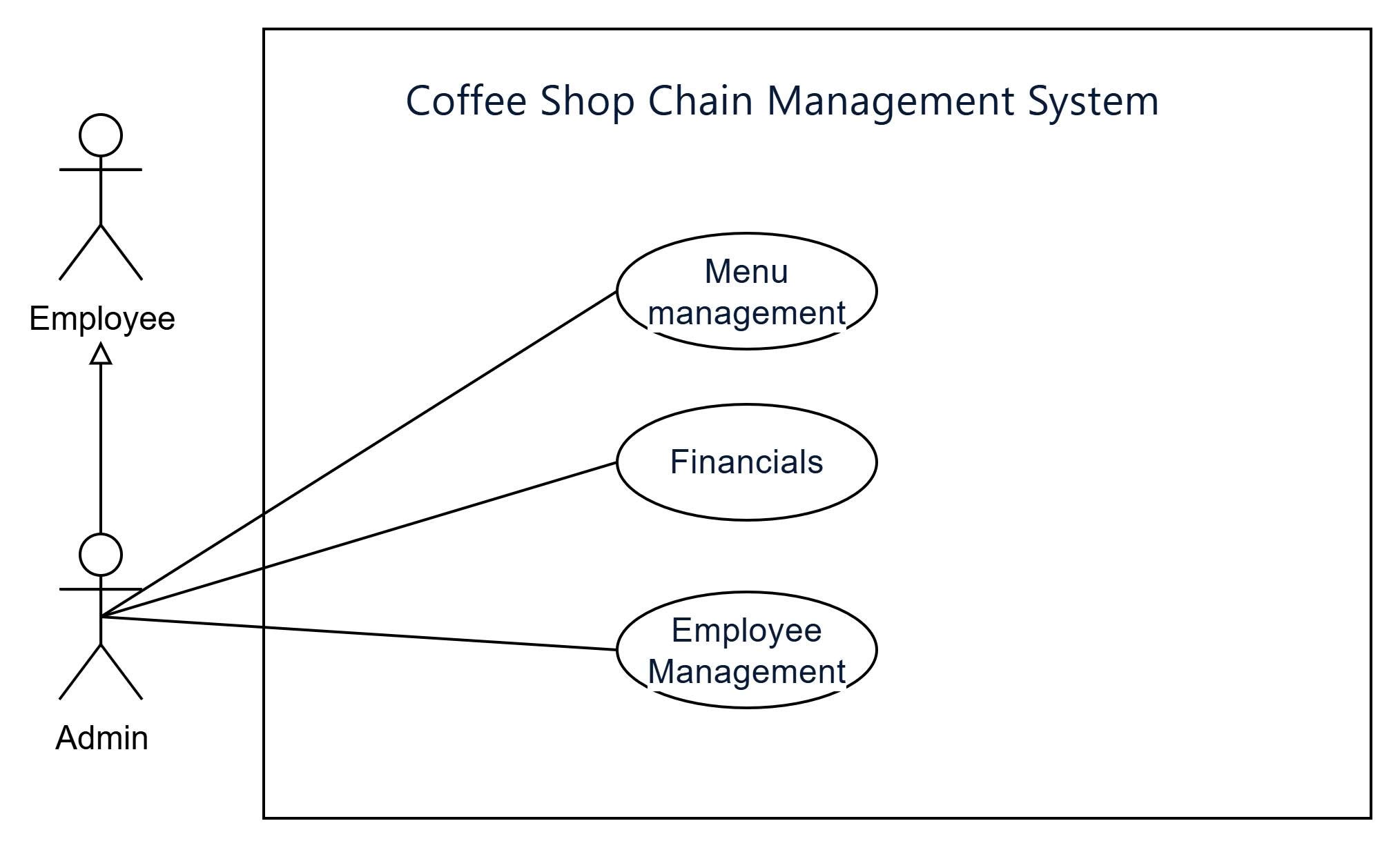
## UML Diagrams

Include relevant UML diagrams:

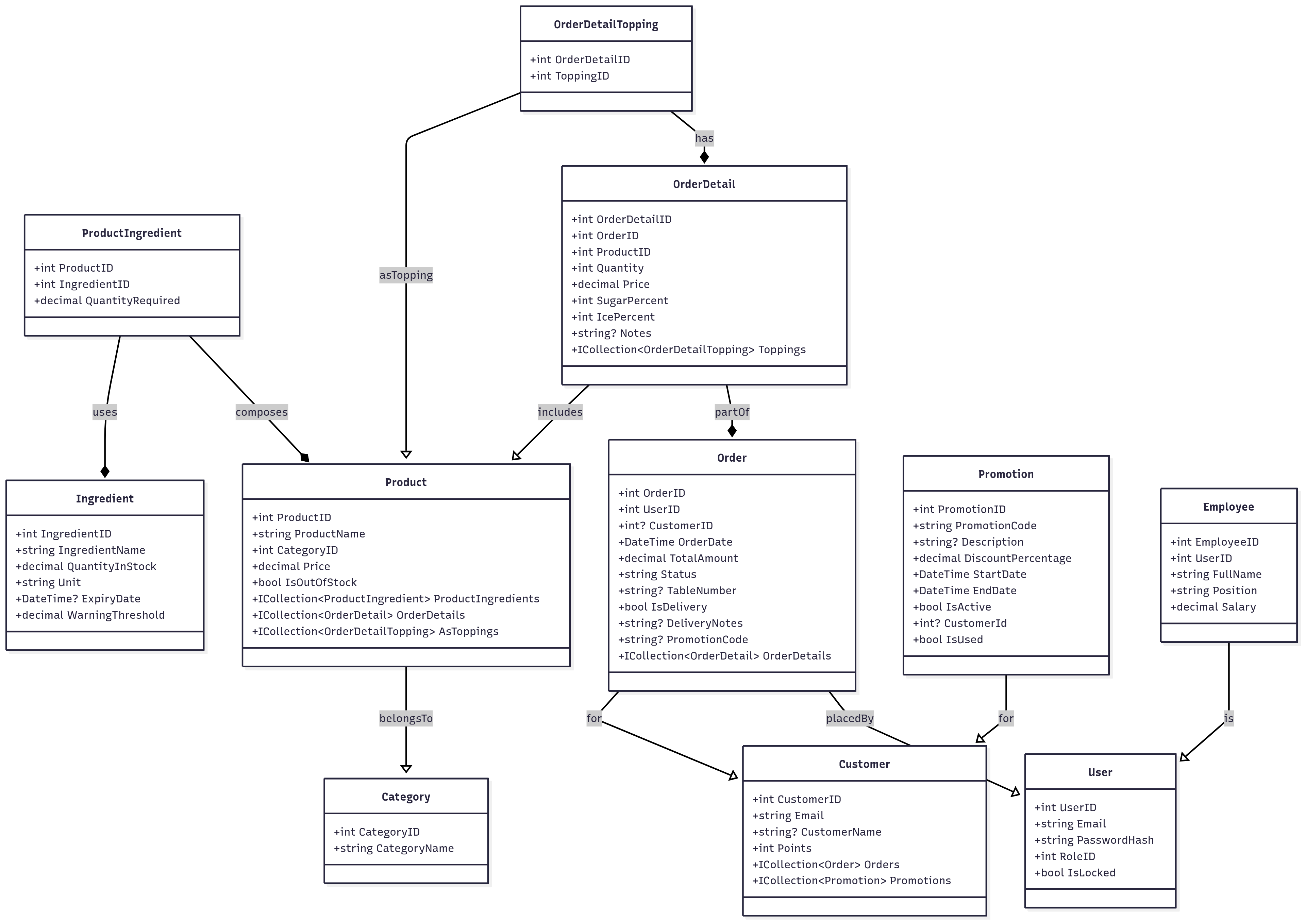
**Use Case Diagram:**

****

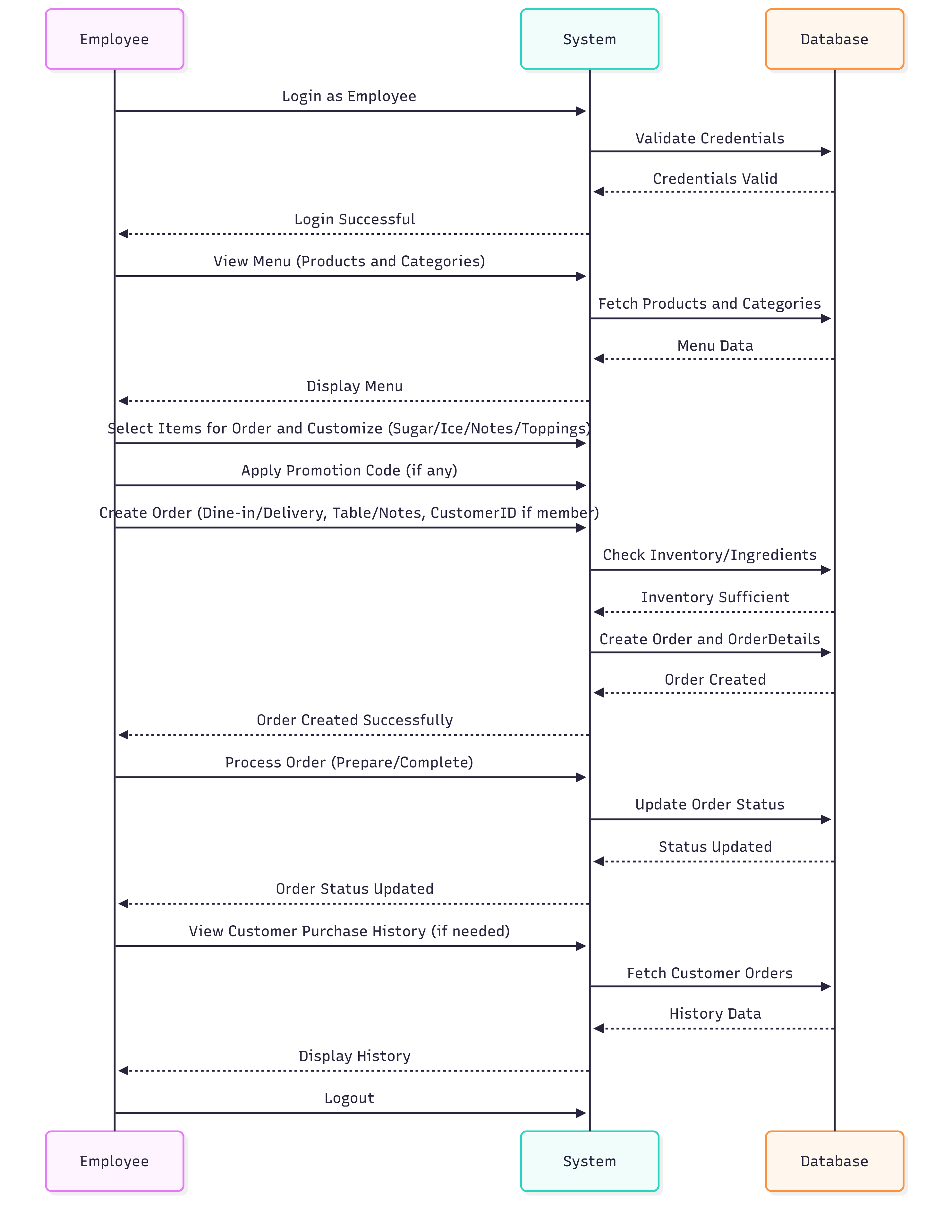
****

****

**Class Diagram:**

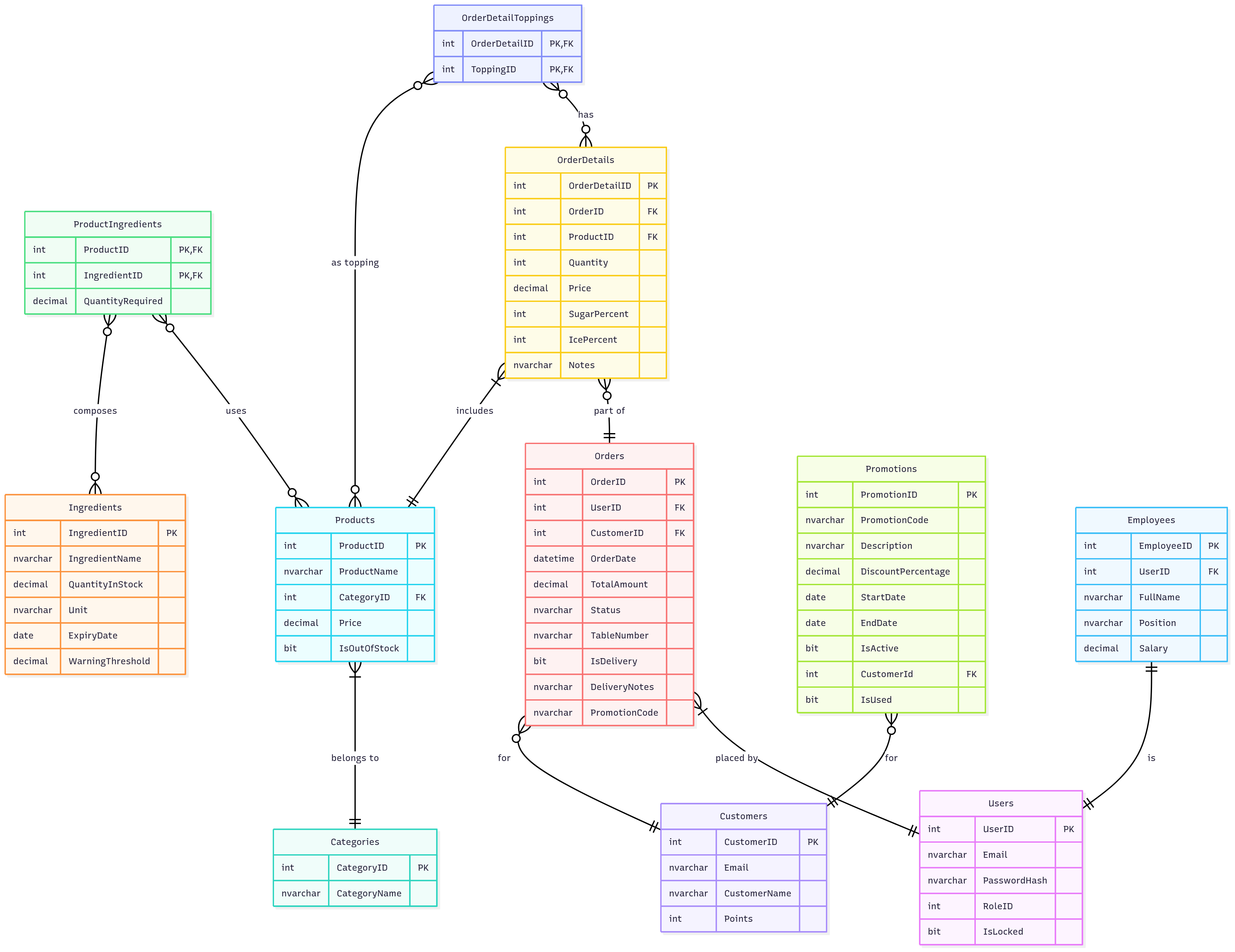


**Sequence Diagram:**

****

## Database Design

**Entity-Relationship Diagram (ERD):**

****

**Database Schema and Table Descriptions:**

CREATE DATABASE PRN\_Project\_Coffee\_Shop;

GO

USE PRN\_Project\_Coffee\_Shop;

GO

-- Table for Users (Admin and Staff)

-- RoleID: 1 = Admin, 2 = Employee

CREATE TABLE Users (

UserID INT PRIMARY KEY IDENTITY(1,1),

Email NVARCHAR(100) NOT NULL UNIQUE,

PasswordHash NVARCHAR(255) NOT NULL,

RoleID INT NOT NULL,

IsLocked BIT NOT NULL DEFAULT 0

);

GO

-- Table for Product Categories (Coffee, Tea, Cake)

CREATE TABLE Categories (

CategoryID INT PRIMARY KEY IDENTITY(1,1),

CategoryName NVARCHAR(100) NOT NULL UNIQUE

);

GO

-- Table for Ingredients

CREATE TABLE Ingredients (

IngredientID INT PRIMARY KEY IDENTITY(1,1),

IngredientName NVARCHAR(100) NOT NULL UNIQUE,

QuantityInStock DECIMAL(10, 2) NOT NULL,

Unit NVARCHAR(20) NOT NULL,

ExpiryDate DATE,

WarningThreshold DECIMAL(10, 2) NOT NULL

);

GO

-- Table for Products (Menu Items)

CREATE TABLE Products (

ProductID INT PRIMARY KEY IDENTITY(1,1),

ProductName NVARCHAR(100) NOT NULL,

CategoryID INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

IsOutOfStock BIT NOT NULL DEFAULT 0,

FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)

);

GO

-- Junction table for Product and Ingredients (Many-to-Many)

CREATE TABLE ProductIngredients (

ProductID INT NOT NULL,

IngredientID INT NOT NULL,

QuantityRequired DECIMAL(10, 2) NOT NULL,

PRIMARY KEY (ProductID, IngredientID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID),

FOREIGN KEY (IngredientID) REFERENCES Ingredients(IngredientID)

);

GO

-- Table for Customers (Members)

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY IDENTITY(1,1),

Email NVARCHAR(100) NOT NULL UNIQUE,

CustomerName NVARCHAR(100),

Points INT DEFAULT 0

);

GO

-- Table for Orders

CREATE TABLE Orders (

OrderID INT PRIMARY KEY IDENTITY(1,1),

UserID INT NOT NULL,

CustomerID INT, -- Nullable for non-member customers

OrderDate DATETIME NOT NULL DEFAULT GETDATE(),

TotalAmount DECIMAL(10, 2) NOT NULL,

Status NVARCHAR(50) NOT NULL, -- e.g., Pending, Preparing, Completed

TableNumber NVARCHAR(10), -- For dine-in orders

IsDelivery BIT NOT NULL,

DeliveryNotes NVARCHAR(MAX), -- For delivery orders

PromotionCode NVARCHAR(50),

FOREIGN KEY (UserID) REFERENCES Users(UserID),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

GO

-- Table for Order Details (Items in an order)

CREATE TABLE OrderDetails (

OrderDetailID INT PRIMARY KEY IDENTITY(1,1),

OrderID INT NOT NULL,

ProductID INT NOT NULL,

Quantity INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

SugarPercent INT NOT NULL DEFAULT 100,

IcePercent INT NOT NULL DEFAULT 100,

Notes NVARCHAR(MAX), -- Customer notes for the item

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

GO

-- Junction table for OrderDetail and Toppings (Many-to-Many)

CREATE TABLE OrderDetailToppings (

OrderDetailID INT NOT NULL,

ToppingID INT NOT NULL, -- This is a ProductID from the Products table

PRIMARY KEY (OrderDetailID, ToppingID),

FOREIGN KEY (OrderDetailID) REFERENCES OrderDetails(OrderDetailID),

FOREIGN KEY (ToppingID) REFERENCES Products(ProductID)

);

GO

-- Table for Promotions

CREATE TABLE Promotions (

PromotionID INT PRIMARY KEY IDENTITY(1,1),

PromotionCode NVARCHAR(50) NOT NULL UNIQUE,

Description NVARCHAR(MAX),

DiscountPercentage DECIMAL(5, 2) NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

IsActive BIT NOT NULL DEFAULT 1,

CustomerId INT,

IsUsed BIT NOT NULL DEFAULT 0,

FOREIGN KEY (CustomerId) REFERENCES Customers(CustomerID)

);

GO

-- Table for Employees

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY IDENTITY(1,1),

UserID INT NOT NULL,

FullName NVARCHAR(100) NOT NULL,

Position NVARCHAR(50) NOT NULL, -- e.g., Barista, Cashier, Cleaner

Salary DECIMAL(10, 2) NOT NULL,

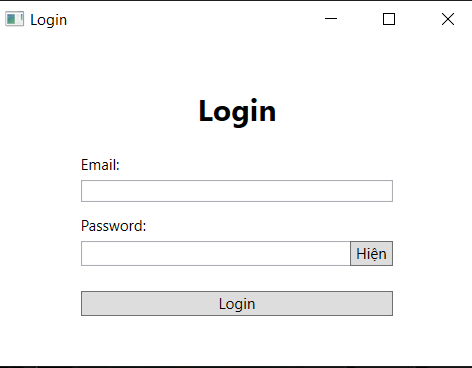
FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

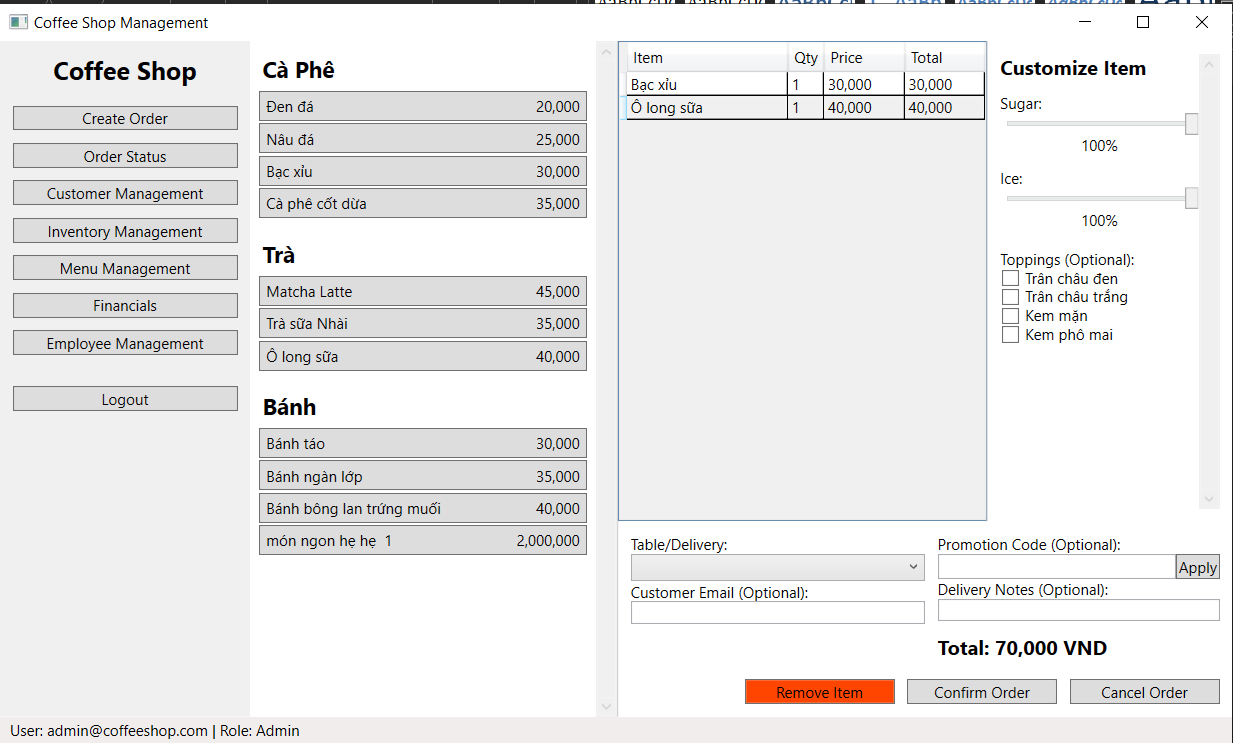
GO

## User Interface (UI) Mockups

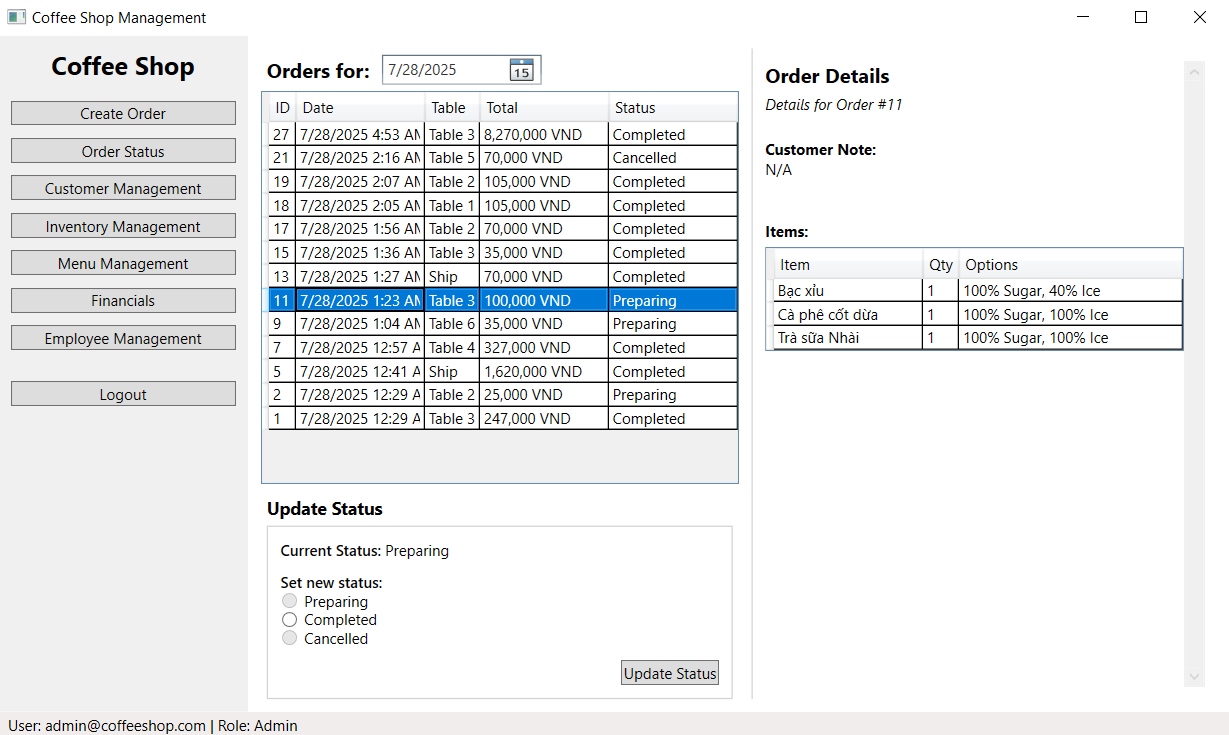
Login:



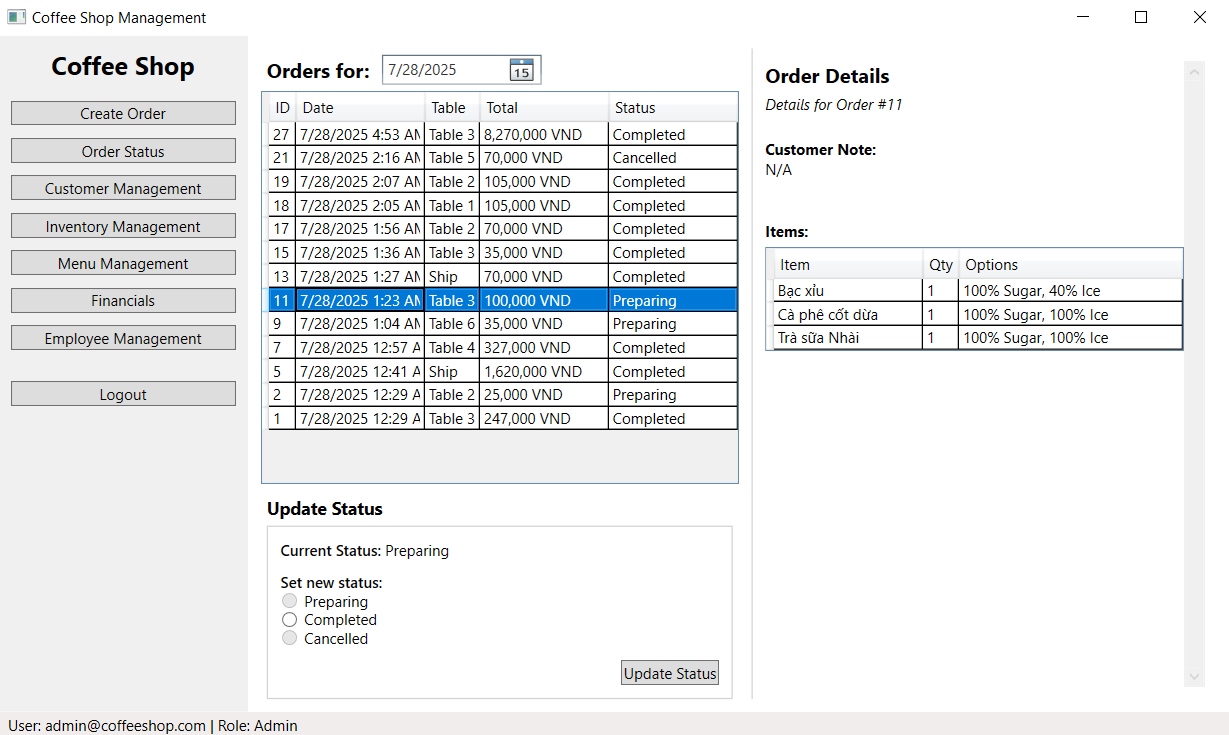
Create order:



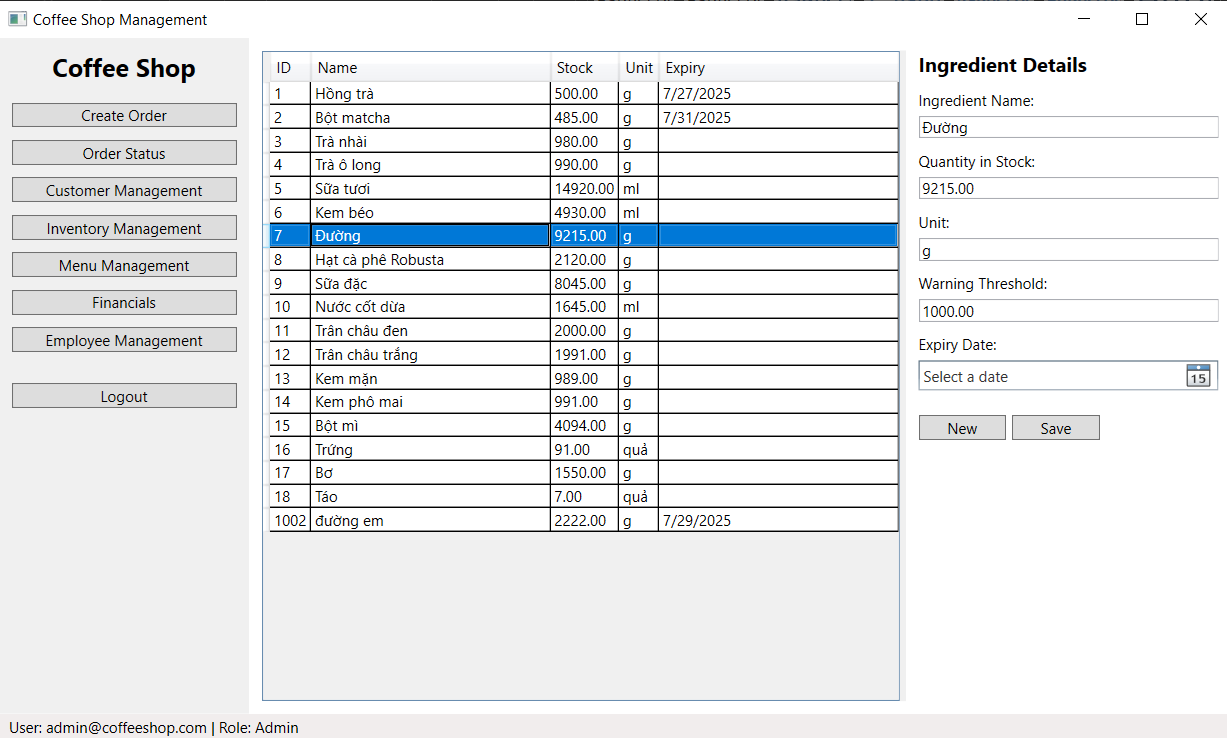
Order status:



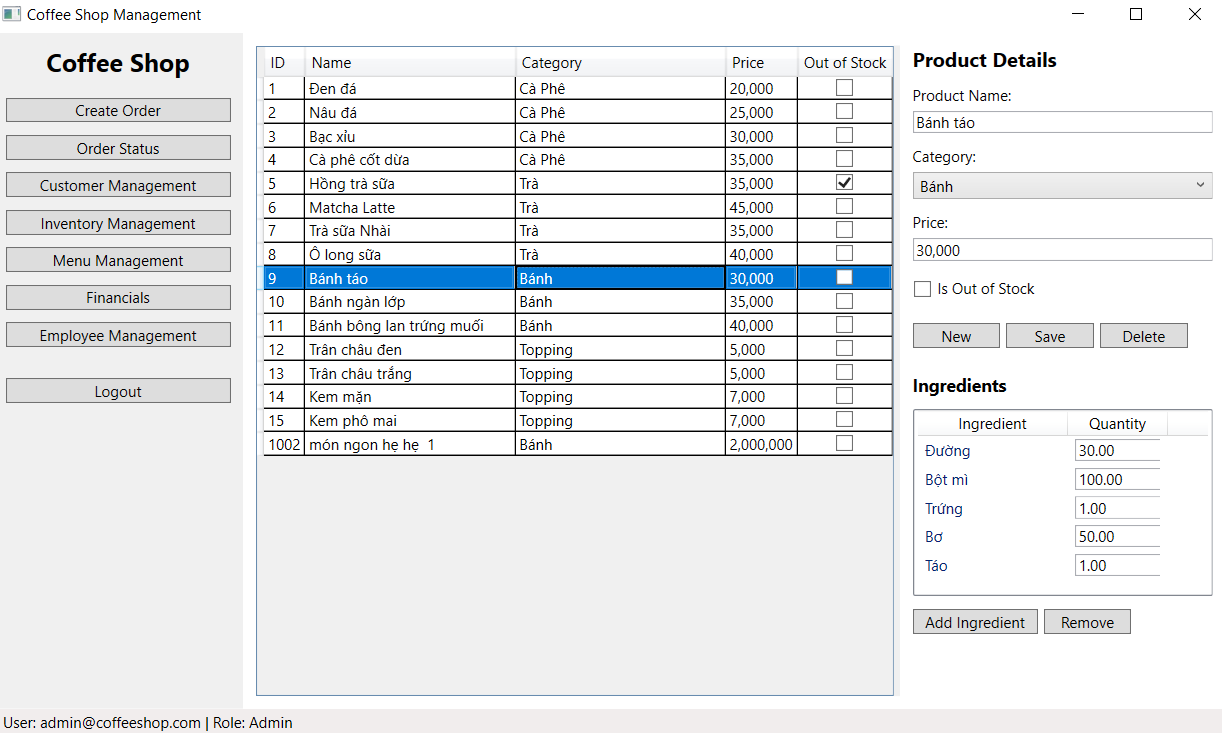
Customer management:



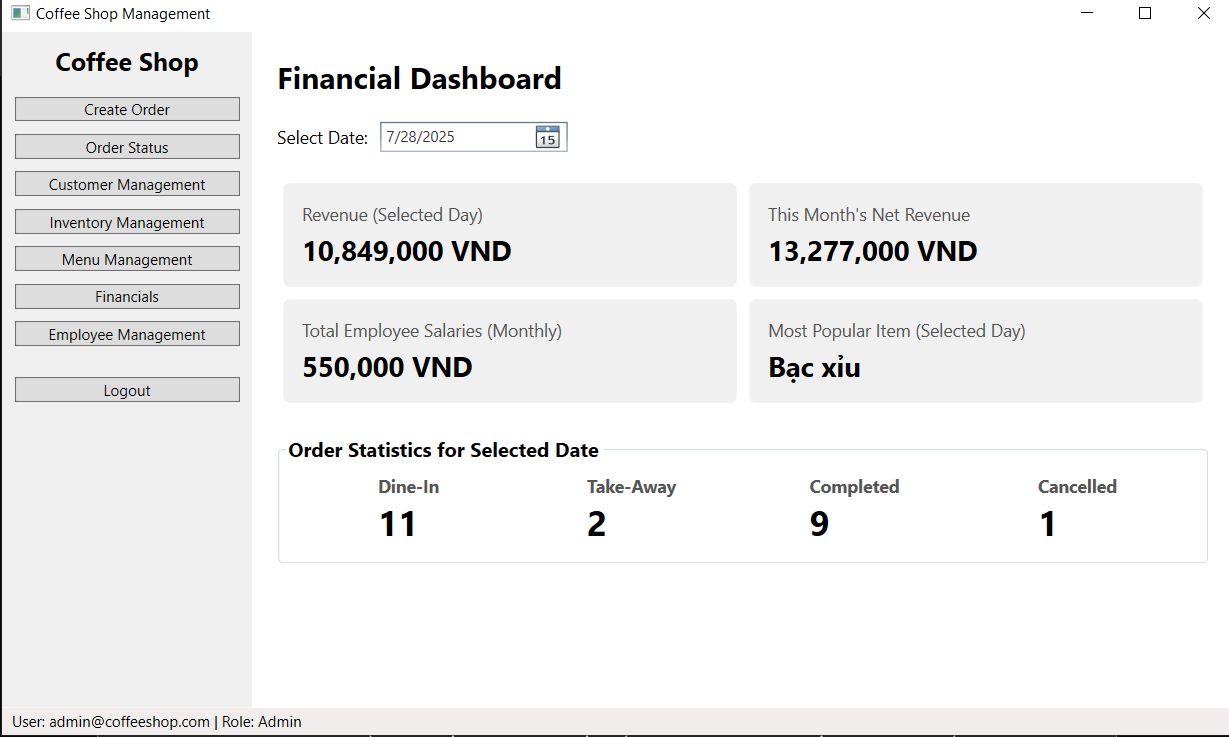
Inventory management:



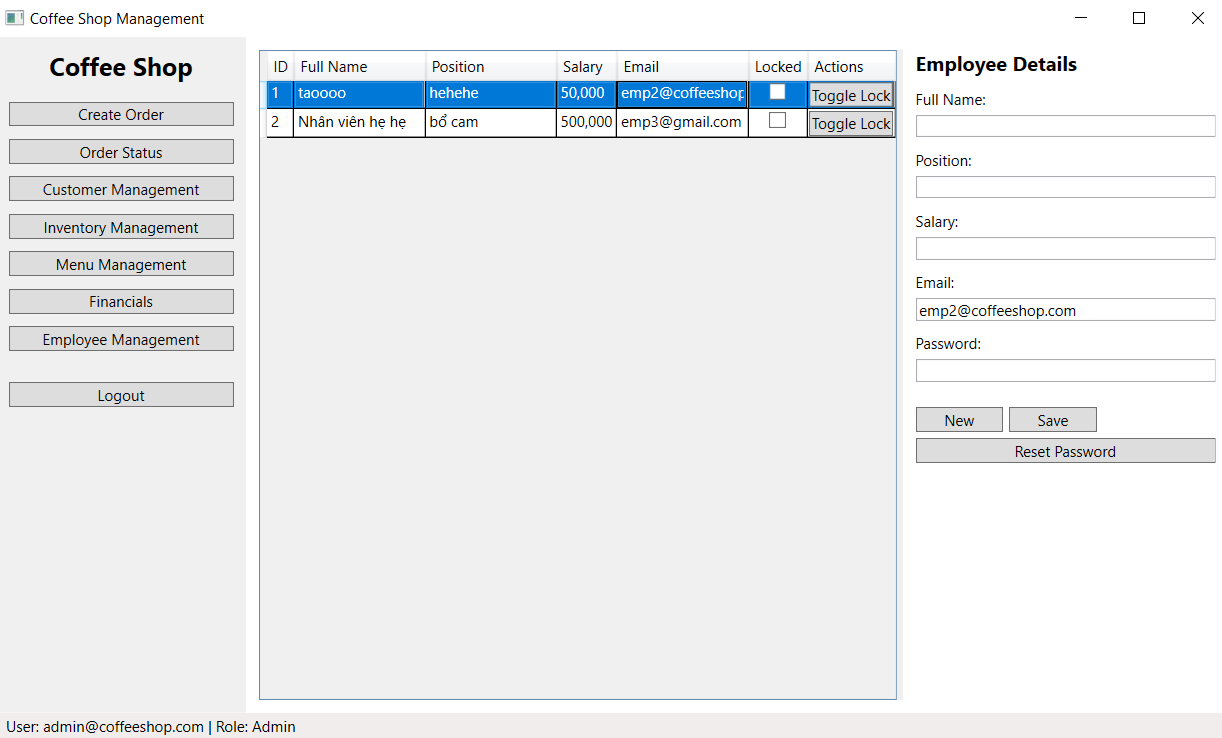
Menu management:



Financials:



Employee Management:



## CI/CD Planning

* Tools: The project will use Git for version control, with the central repository hosted on GitHub. The CI/CD pipeline will be implemented using GitHub Actions.
* Pipeline:
  1. Trigger: The workflow will be triggered on every push to the main branch or on creation of a pull request.
  2. Build: The first step will be to build the .NET solution (PRN\_Project\_Coffee\_Shop.sln) to ensure the code compiles successfully.
  3. Test: The pipeline will then execute the unit tests in the CoffeeShop.Tests project using the MSTest framework.
  4. Notification: The results of the build and test runs will be reported back to the pull request or commit status on GitHub.
* Integration: This automated process ensures that new code is continuously validated against the existing codebase, preventing integration issues and maintaining code quality.

## Team Contributions

As this is a solo project, all aspects of the system analysis and architectural design were handled by the sole developer:

* Nguyễn Trọng Hiệp (HE190006): Responsible for defining the system overview, creating all UML and ERD diagrams, designing the database schema, mocking up the UI, and planning the CI/CD pipeline.