# Food Delivery System - Project Documentation

**Date:** May 10, 2025

**Course:** CSCI217

**Instructor:** [Dr. Sameh / Eng. Nada

**Team Members:**

Karim Bassem 231000797

Nadeem Diaa 231000

Ahmed Fahmy 231000

Ahmed Shalaby 231000

Yassin Mashhour 231000

## 1. Introduction

This document provides a comprehensive overview of the Food Delivery System application. The system is a JavaFX-based desktop application designed to simulate a basic food delivery service. It allows users to browse a menu, place orders, and track their status. Administrators have privileges to manage menu items and view sales data.

The primary goal of this project is to demonstrate proficiency in Java GUI development, object-oriented programming principles, and basic software engineering practices, including error handling and documentation.

## 2. Features Implemented

The Food Delivery System implements the following core features:

### 2.1. User Features

* **View Restaurant Menu:** Users can browse a visually organized menu displaying food items with their names, prices, images, and categories.
* **Place an Order:** Users can add items from the menu to a shopping cart, view the total price, and submit the order.
* **Cancel an Order:** Users can look up an existing order by its ID and cancel it if it’s not already processed or canceled.
* **View Order Status:** Users can check the current status of their order using the order ID.

### 2.2. Admin Features

* **Admin Login:** A secure login interface for administrators.
* **Add Menu Item:** Administrators can add new items to the menu, specifying ID, name, price, category, and an image.
* **Edit Menu Item:** Administrators can load existing menu items by ID and modify their details, including name, price, category, and image.
* **Remove Menu Item:** Administrators can remove items from the menu.
* **View Sales:** Administrators can view the total sales for the current day and a historical log of sales categorized by date.

### 2.3. General System Features

* **Error Checking:** Input validation is implemented across various forms to ensure data integrity and guide users (e.g., non-negative prices, non-empty fields, unique item IDs).
* **Graphical User Interface (GUI):** The application is developed using JavaFX, providing an interactive and user-friendly interface.

## 3. System Design

### 3.1. Architecture

The application primarily follows a Model-View-Controller (MVC) architectural pattern:

* **Model:** Represents the data and business logic.
  + MenuItems.java: Defines the structure for menu items (ID, name, price, category, image path).
  + Order.java: Defines the structure for customer orders (order ID, list of items, total price, customer details, status, timestamp).
* **View:** Represents the user interface.
  + FXML files (HomePage.fxml, Menu.fxml, AddItem.fxml, etc.): Define the layout and components of each screen.
* **Controller (JavaFX Controllers):** Handle user input, interact with the Model, and update the View.
  + HomePageController.java, MenuController.java, AddItemController.java, etc.: Contain the logic for their respective FXML views.
* **Controller (Business Logic Controllers):** Encapsulate core business operations.
  + MenuItemsController.java: Manages CRUD operations for menu items.
  + OrderController.java: Manages order creation, retrieval, and status updates.
  + SalesController.java: Manages tracking and retrieval of sales data.

### 3.2. Key Components and Modules

* **com.fooddeliverysystem (Main Package):**
  + App.java: The main application class that initializes JavaFX and loads the primary scene.
  + JavaFX View Controllers (e.g., HomePageController, MenuController): Manage interactions for specific FXML views.
* **com.fooddeliverysystem.model (Model Package):**
  + Contains data classes like MenuItems and Order.
* **com.fooddeliverysystem.controller (Business Logic Controller Package):**
  + Contains classes like MenuItemsController, OrderController, and SalesController that handle the core application logic separate from direct UI manipulation.
* **Resources (src/main/resources):**
  + FXML files defining the UI.
  + Image assets used in the application (e.g., Logo.png, AdminLogo.png, item images).

## 4. User Interface (Simulation Snapshots)

This section provides snapshots of the application’s user interface, demonstrating its functionality.

### 4.1. Home Page

*Description: The main landing page of the application with navigation options.*

### 

### 4.2. Menu Page

*Description: Displays the restaurant menu with food items.*

### 

### 4.3. Add Item Page

*Description: Admin interface to add new menu items.*

### 

### 4.4. Edit Item Page

*Description: Admin interface to edit existing menu items.*

### 

### 4.5. Remove Item Page

*Description: Admin interface to remove menu items.*

### 

### 4.6. View Sales Page

*Description: Admin interface to view daily and historical sales data.*

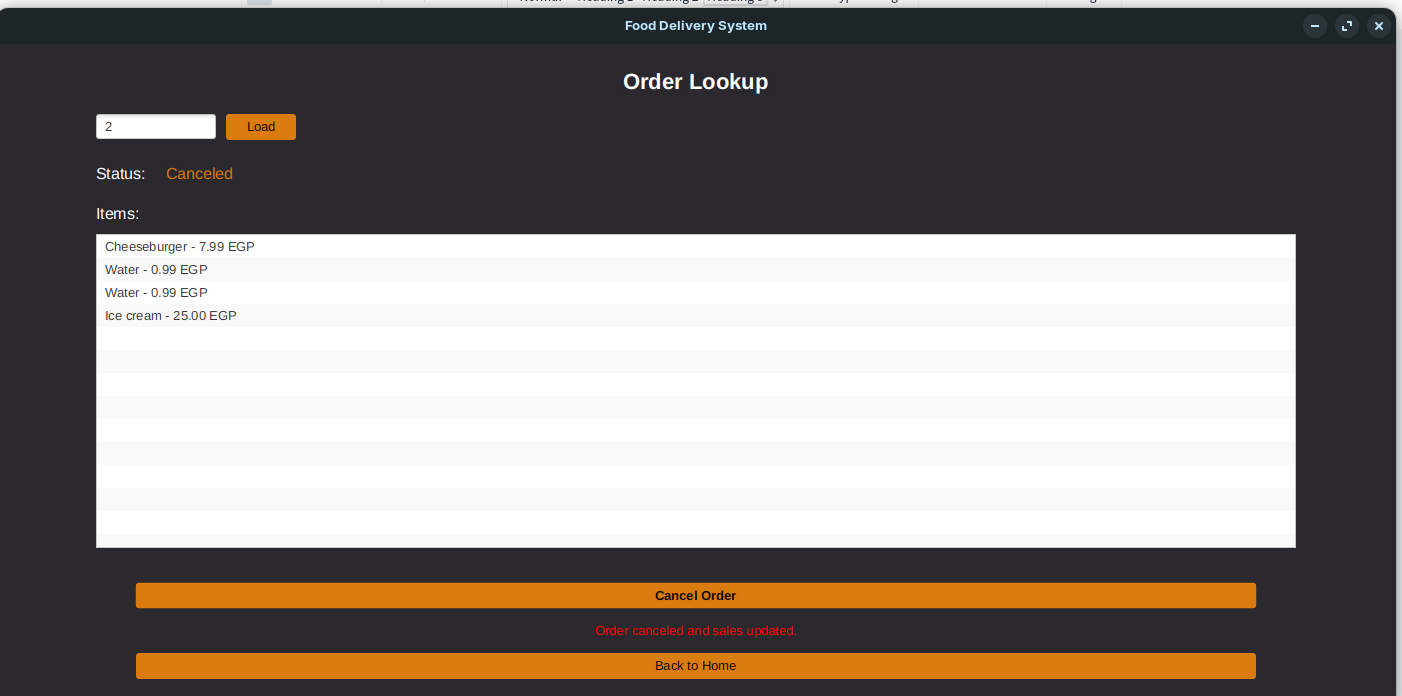
### 

### 4.7. Place Order Page

*Description: User interface to place an order and view the cart.*

### 4.8. Cancel Order Page

*Description: User interface to cancel an existing order.*



## 5. Error Handling

The application includes robust error handling and input validation:

* Non-negative prices are enforced for menu items.
* Unique item IDs are required for menu items.
* All required fields must be filled before submitting forms.
* Invalid inputs (e.g., negative prices, empty fields) are flagged with appropriate error messages.