DoyDoy Food Ordering and Management System Design Document

Version: 1.0

Document Revision History

Date	Version	ID	Author
25/05/2024	1.0	230303033	Fatih Çatalçam

Contents

1. Introduction

1.1 Purpose

This document provides a comprehensive design for the imaginary company DoyDoy Food Ordering and Management System, which is designed to manage the operations and control of the DoyDoy restaurant chain.

1.2 System Overview

The DoyDoy Management System facilitates efficient management of restaurant operations including staff management, inventory control, and order processing.

1.3 Design Objectives

- Usability: Provide an intuitive interface for restaurant managers and staff.

- Scalability: Ensure the system can scale with the growth of the restaurant chain.

- Security: Protect sensitive data such as employee information and financial records.

1.4 Definitions, Acronyms, and Abbreviations

- OOP: Object-Oriented Programming

- GUI: Graphical User Interface

2. Design Overview

2.1 Introduction

This section provides an overview of the design considerations and the overall architecture of the DoyDoy Management System.

2.2 Environment Overview

The system will operate in a client-server environment, where the client applications interact with a centralized server.

2.3 System Architecture

2.3.1 Top-level system structure

- Client Application: Interfaces for restaurant staff and management.

- Server Application: Handles business logic and data storage.

2.3.2 Subsystems

- Employee Management Subsystem

- Inventory Management Subsystem

- Order Processing Subsystem

2.4 Constraints and Assumptions

- The system will not include online ordering functionality.

- User authentication will be managed through predefined roles and permissions.

3. Interfaces and Data Stores

3.1 System Interfaces

3.1.1 User Interface

The system will have graphical user interfaces (GUIs) for different types of users, including managers and regular staff.

3.1.2 API Interfaces

APIs will be provided for integrating with third-party systems, such as payment gateways.

3.2 Data Stores

All data will be stored in a relational database. Key data stores include:

- employees.txt: Stores employee records.

- inventory.txt: Stores inventory records.

- orders.txt: Stores order records.

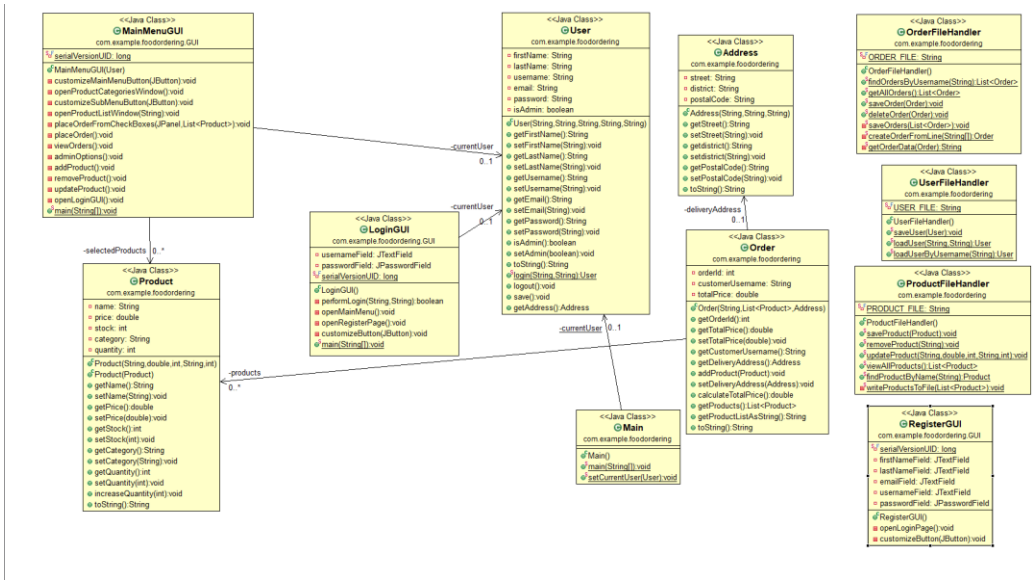
4. Structural Design

4.1 Design Explanation and Rationale

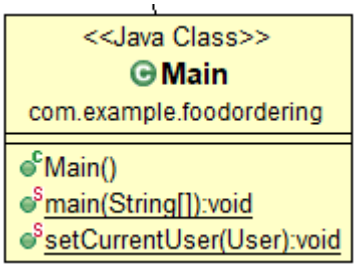
The DoyDoy Management System is designed using OOP principles, leveraging polymorphism and encapsulation to create a modular and maintainable codebase.

4.2 Class Diagram

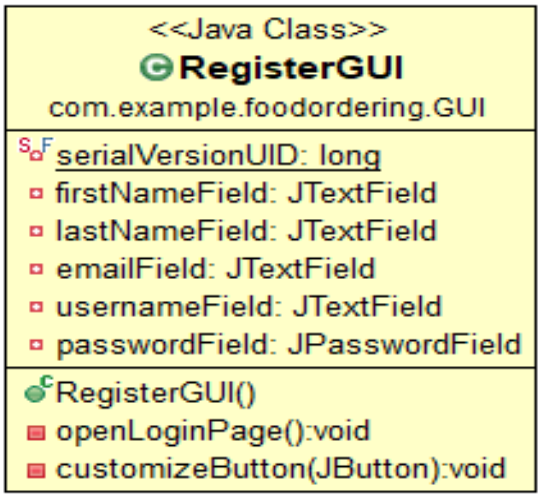
Class Diagram as a whole:



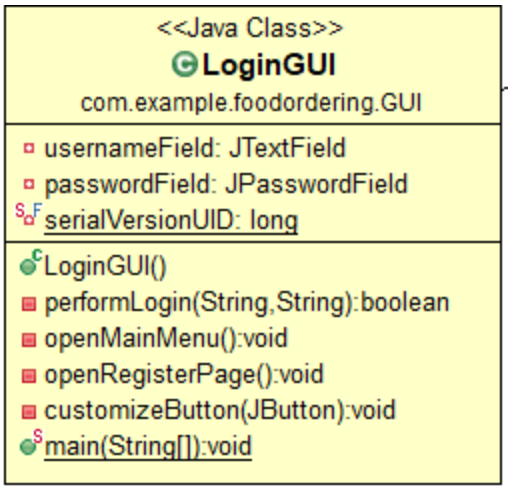
Main



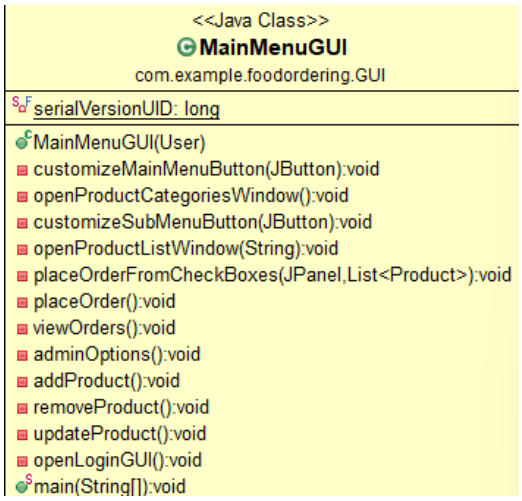
Register GUI



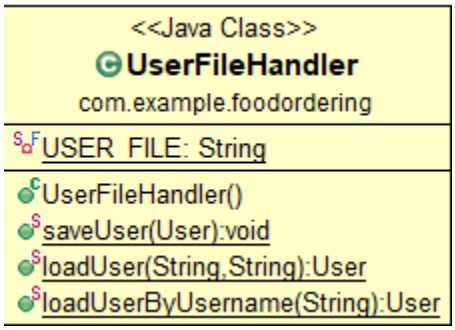
Login GUI



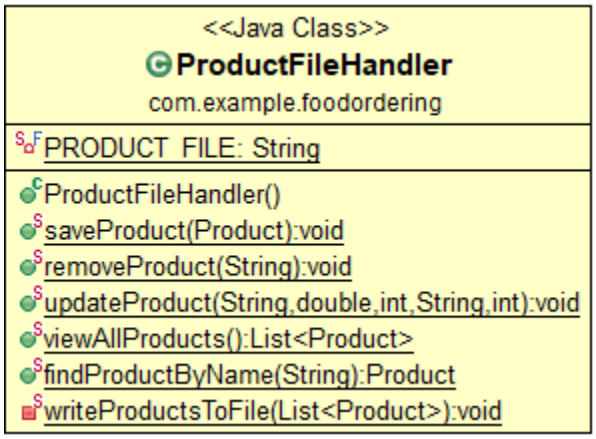
MainMenu GUI



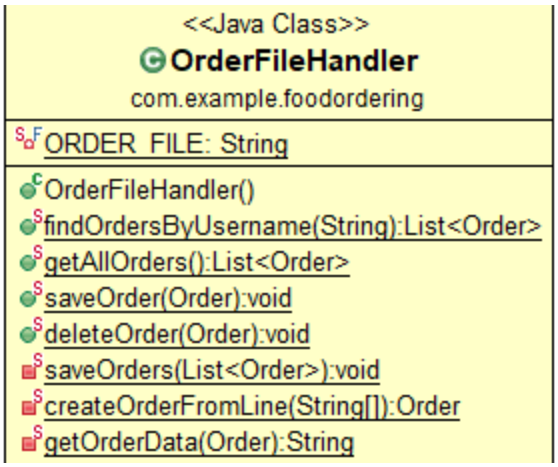
UserFileHandler



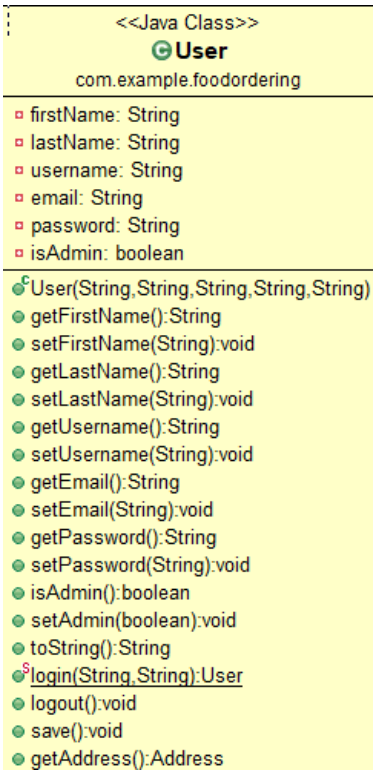
ProductFileHandler



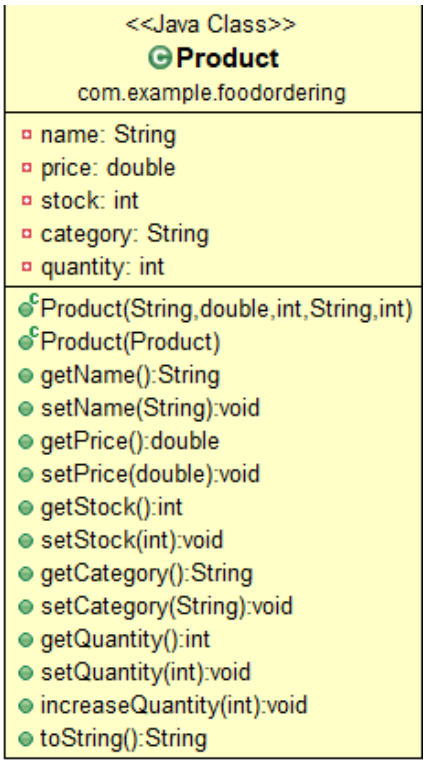
OrderFileHandler



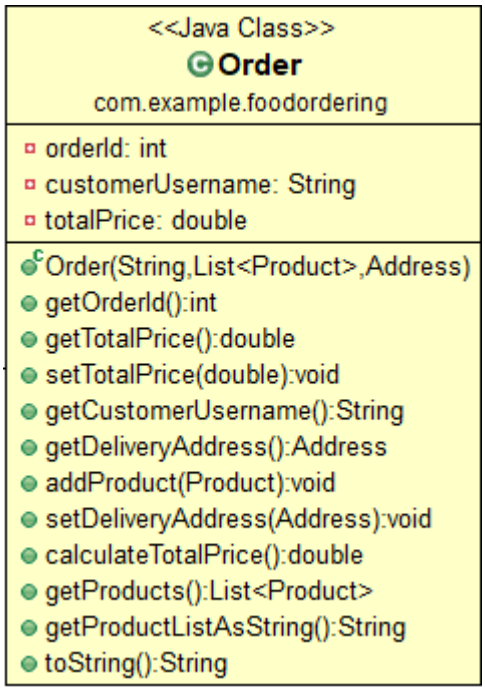
User



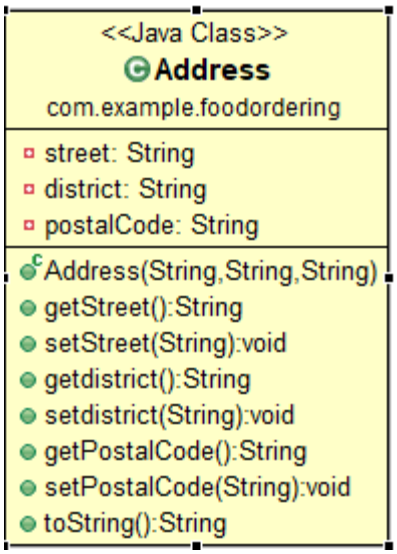
Product



Order



Address



4.3 Class Descriptions

4.3.1 Class: Employee

- Purpose: To model the relevant aspects of an employee.

- Attributes:

  - name (String): Stores the employee's name.

- id (String): Stores the unique employee ID.

- position (String): Stores the employee's position.

- Methods:

- createEmployeeID(): Generates a unique employee ID.

- getName(): Returns the employee's name.

- setName(String name): Sets the employee's name.

4.3.2 Class: Manager (extends Employee)

- Purpose: To model the relevant aspects of a manager.

- Additional Attributes:

- password (String): Stores the manager's password.

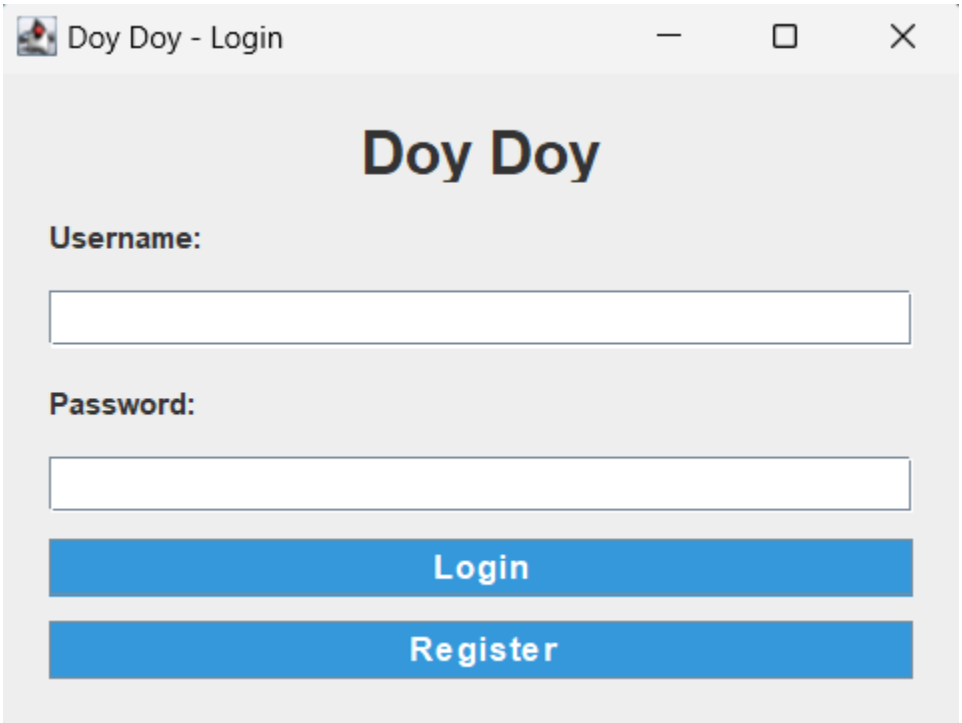
- Additional Methods:

- createManagerPassword(): Generates a unique password for the manager.

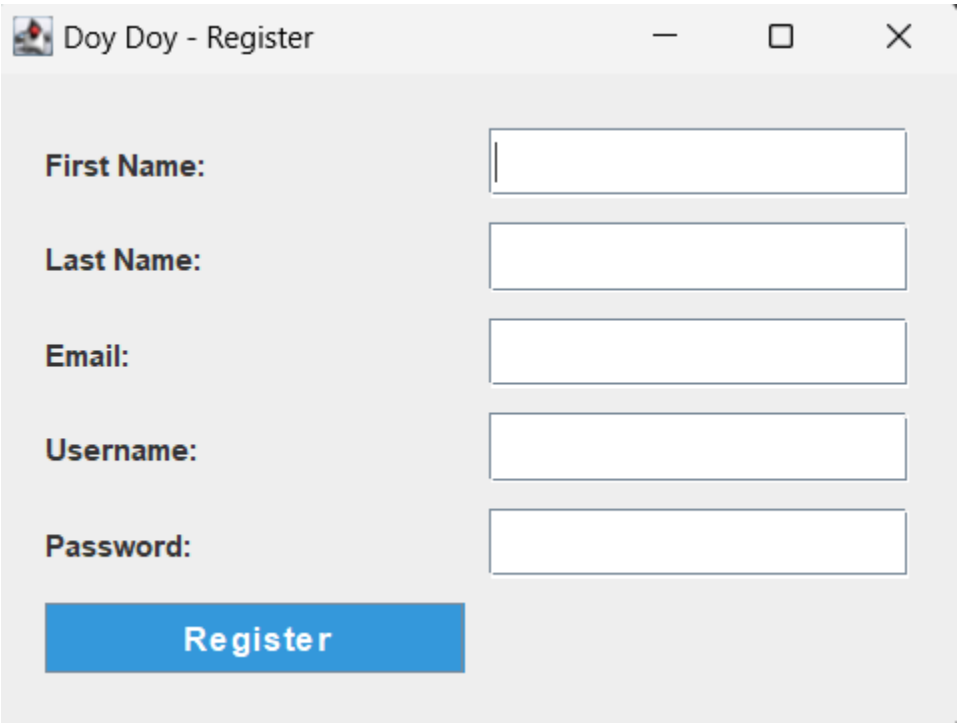
5. Dynamic Model

5.1 Scenarios

First off when we start the application The Login Page comes up.

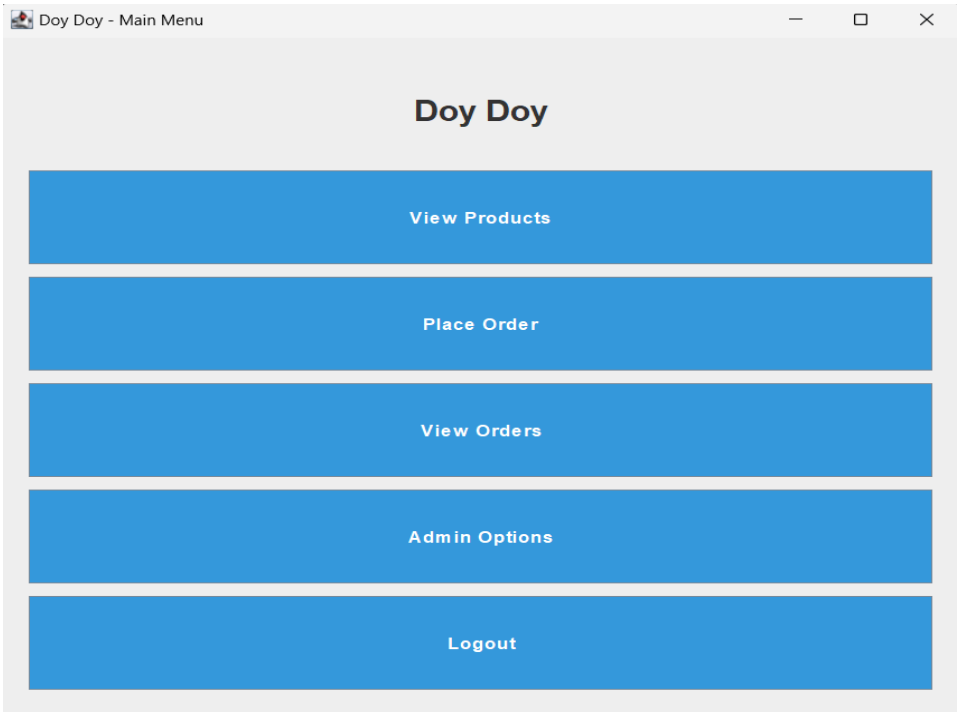


We need to register to be able to login the app, so we go to Register page.



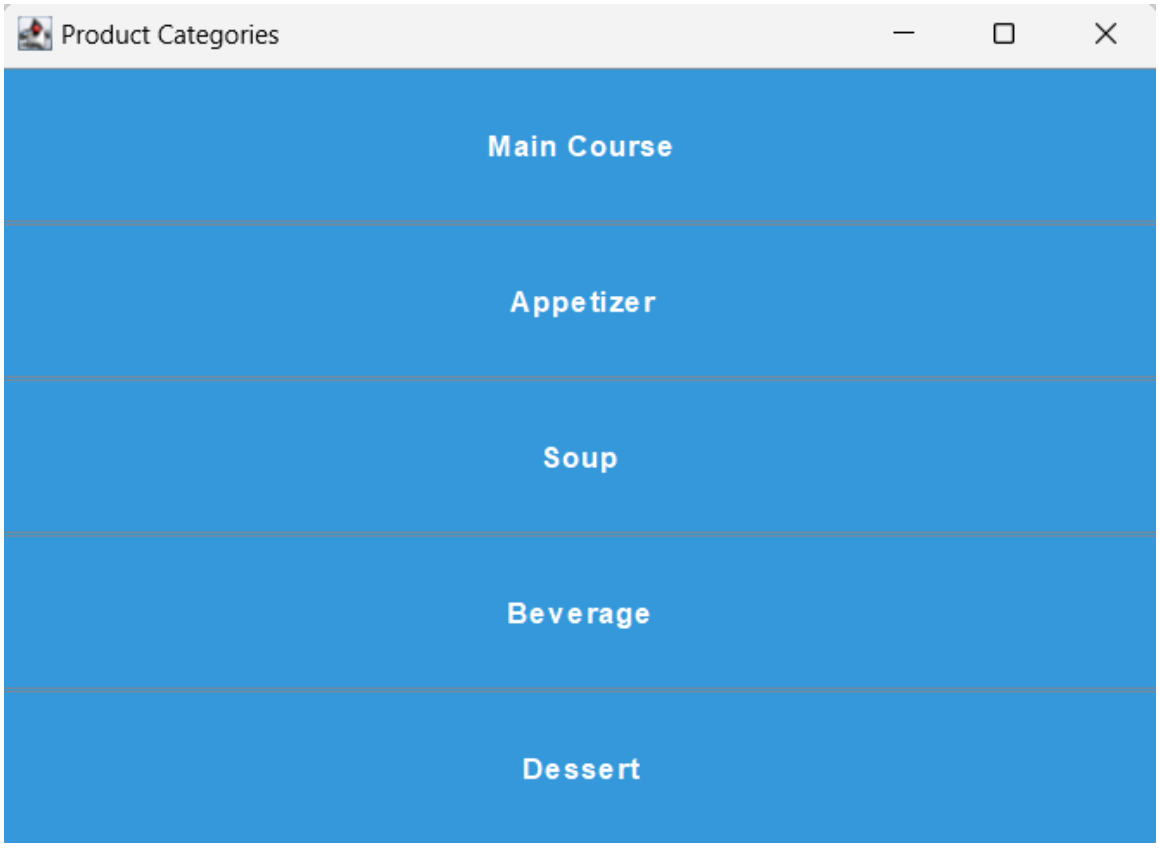
After we register, we go back to the login page again and login successfully.

It takes us to the Main page.



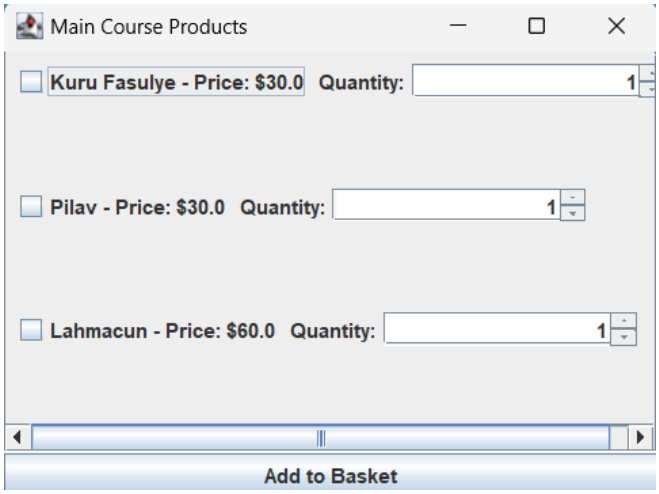
Here we can View Products that Administrator account added, Place orders, View Orders, Check out Admin Options and Logout.

Let's take a closer look to the View Products



We can see here our 5 main categories.

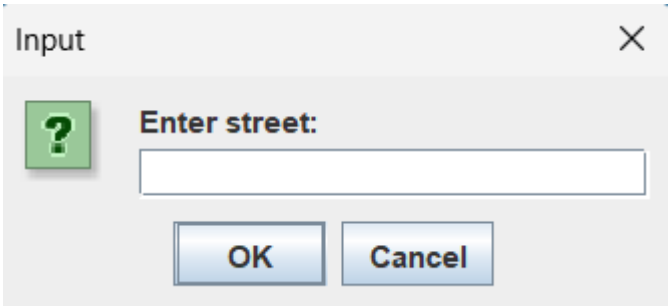
For example, let's take a look at Main Courses

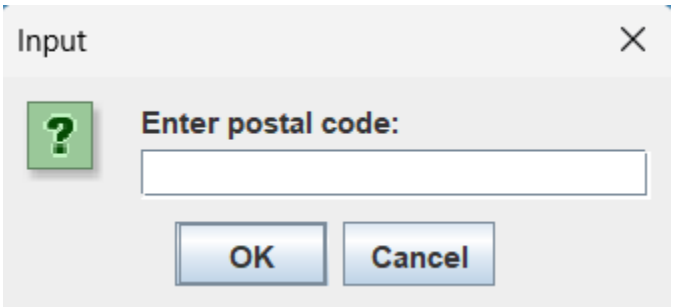


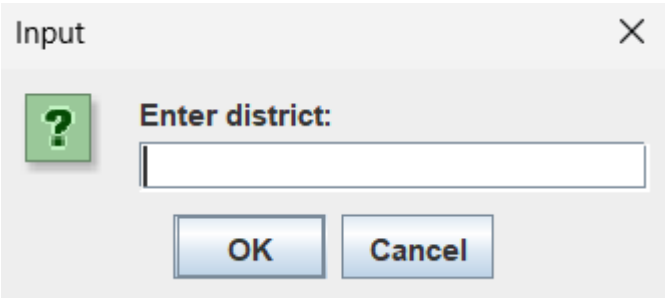
Here we can choose the food and its quantity that we will be going to add to our basket.

Now let's order the food that we added to our basket.

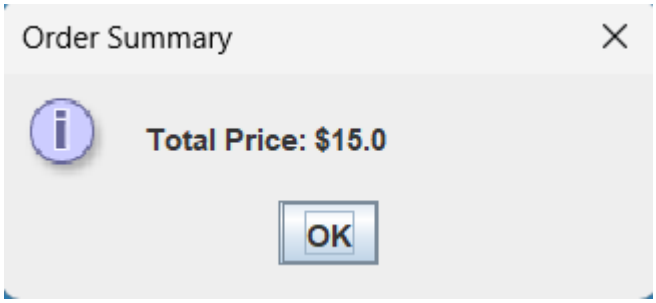
First off it asks for us or address details.



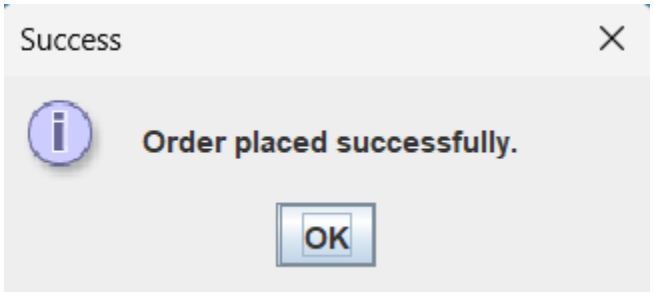




And than gives us the information of the price



After we click OK, we get an information pop up.

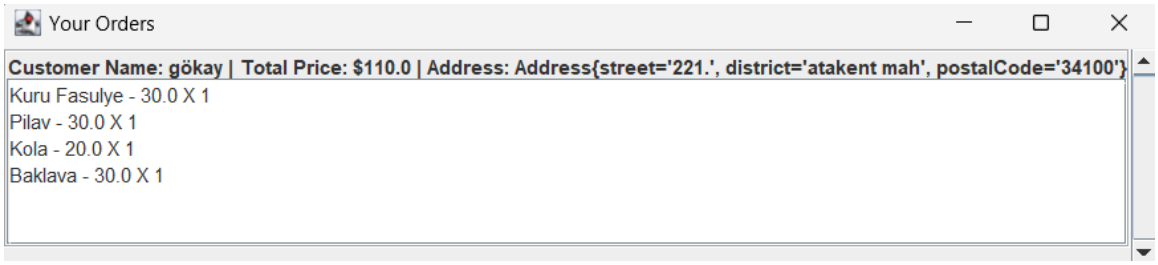


Now Its Time to View Our Order

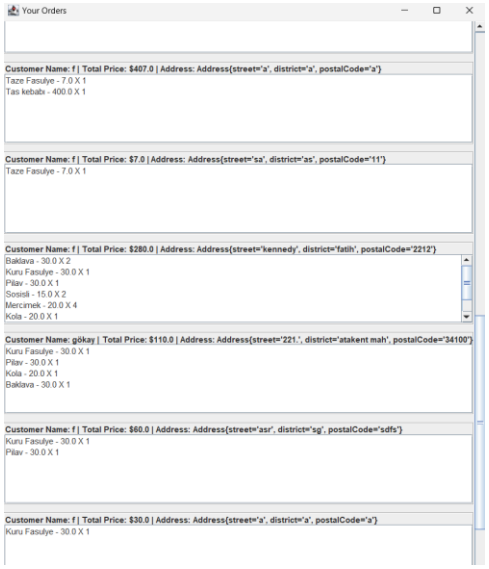
Normal users can only see their orders, but Administrator account can see all orders.

Here is the example.

This is what an normal account sees.

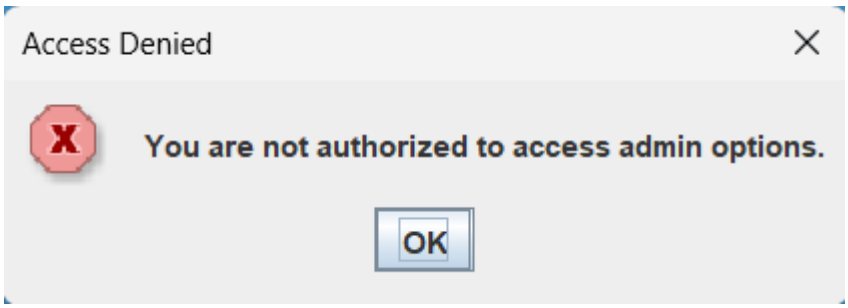


And this is what an administrator account sees.



Now let's get to the Admin Options

Normal users can't access admin options when they try to, they get an error like this.



Now this is what Administrator accounts see.

Admin accounts can Add products, Remove Products and update them.