

Matthew Pereira

6245576

Data Structures and Object Oriented Programming

May 11, 2025

# **Java Food Ordering System**

## **Table of Contents:**

Pg 3. Project Description

Pg 3. Program Features

Pg 4. Screenshots and Execution

Pg 9. Challenges Faced

Pg 9. Learning Outcomes

## **Project Description:**

This project simulates a food ordering system where the user utilises three given user roles and role codes (customers, administrators and drivers) to manage a menu-based order management system or to order food online.

Users and passwords are hardcoded:

- Customer: (cust, cust123)
- Admin: (admin, admin123)
- Driver: (driver, driver123)

## **Program Features:**

Customers can view menu items, select food and place orders (either takeout or delivery), apply discount codes which, when valid give a 15% discount, and track the status of their orders, including updated delivery and collection times.

Administrators log in with a special role code and have the ability to add new discount codes and update the pick-up time for each takeout order.

Drivers also log in with a role code and automatically receive delivery orders in a pending orders queue. They can view these pending deliveries which allow them to see the full order details and customer address to which they can deliver by typing “drive” changing the order status to delivered and saving the order to a csv file.

## Screenshots and Execution:

Customer dashboard + view menu

```
=== LOGIN ===
Username: cust
Password: cust123
Welcome Customer!

--- Customer Menu ---
1. View Menu
2. Place Order
3. View Order Statuses
4. Logout
Choose an option: 1
Menu Items:
- Burger (Fast Food) - $9.99
- Fries (Fast Food) - $3.49
- Tomato Salad (Italian) - $6.25
- Pepperoni Pizza (Italian) - $12.75
- Sushi (Japanese) - $35.5
- Tacos (Mexican) - $8.25
- Pad Thai (Thai) - $14.5
- Steak (Dinner) - $50.0
- Smoothie (Drinks) - $5.75
- Coffee (Drinks) - $2.5
- Nutella Cookie (Dessert) - $3.0
```

## Customer + placed delivery order

```
=== LOGIN ===
Username: cust
Password: cust123
Welcome Customer!

--- Customer Menu ---
1. View Menu
2. Place Order
3. View Order Statuses
4. Logout
Choose an option: 2

Choose items (comma-separated index):
1. Burger - $9.99
2. Fries - $3.49
3. Tomato Salad - $6.25
4. Pepperoni Pizza - $12.75
5. Sushi - $35.50
6. Tacos - $8.25
7. Pad Thai - $14.50
8. Steak - $50.00
9. Smoothie - $5.75
10. Coffee - $2.50
11. Nutella Cookie - $3.00
1,2,3
Takeout or Delivery: Delivery
Enter discount code (or leave empty):
Order saved successfully.
Delivery Order ID: 1 Total Price: 25.23 Driver: driver Address: 123 Maple St Status: Pending

--- Customer Menu ---
1. View Menu
2. Place Order
3. View Order Statuses
4. Logout
Choose an option: |
```

## Driver view order statuses

```
=== LOGIN ===
Username: driver
Password: driver123
Welcome Driver!
Driver dashboard

--- Driver Menu ---
1. View Pending Orders
2. Deliver Next Order
3. Logout
Choose an option:
Invalid choice.

--- Driver Menu ---
1. View Pending Orders
2. Deliver Next Order
3. Logout
Choose an option: 1
Delivery Order ID: 1 Total Price: 25.23 Driver: driver Address: 123 Maple St Status: Pending
```

Driver delivered customer placed order

```
--- Driver Menu ---
1. View Pending Orders
2. Deliver Next Order
3. Logout
Choose an option: 2
Type 'drive' to deliver next order: drive
Order 1 has been delivered to address: 123 Maple St
```

After order is delivered status changes to delivered on customer menu

```
--- Customer Menu ---
1. View Menu
2. Place Order
3. View Order Statuses
4. Logout
Choose an option: 3
Order ID: 1 Type: Delivery Status: Delivered Total: $25.23
Order ID: 2 Type: Takeout Status: Pending Total: $42.5 Pickup Time: 60 minutes from now
```

Admin menu + created discount code

```
--- Admin Menu ---
1. Add Discount Code
2. Edit Pickup Time for Takeout Order
3. Logout
Choose an option: 1
Enter discount code to add: bananafrenzy
Discount code added: bananafrenzy
```

Customer menu + discount code applied for take out

```
=== LOGIN ===
Username: cust
Password: cust123
Welcome Customer!

--- Customer Menu ---
1. View Menu
2. Place Order
3. View Order Statuses
4. Logout
Choose an option: 2

Choose items (comma-separated index):
1. Burger - $9.99
2. Fries - $3.49
3. Tomato Salad - $6.25
4. Pepperoni Pizza - $12.75
5. Sushi - $35.50
6. Tacos - $8.25
7. Pad Thai - $14.50
8. Steak - $50.00
9. Smoothie - $5.75
10. Coffee - $2.50
11. Nutella Cookie - $3.00
8
Takeout or Delivery: takeout
Enter discount code (or leave empty): bananafrenzy
Order saved successfully.
Takeout Order ID: 2 Total Price: 42.5 Pickup Time: 60 minutes from now Status: Pending
```

Admin edits order 2 takeout pick up time

```
--- Admin Menu ---
1. Add Discount Code
2. Edit Pickup Time for Takeout Order
3. Logout
Choose an option: 2
Takeout Order ID: 2 | Current Pickup Time: 60 minutes from now
Enter Order ID to edit: 2
Enter new pickup time (max 75 minutes from now): 35
Pickup time for Order 2 set to: 35 minutes from now
```

Menu Items loaded from csv

1	Burger, Fast Food, 9.99
2	Fries, Fast Food, 3.49
3	Tomato Salad, Italian, 6.25
4	Pepperoni Pizza, Italian, 12.75
5	Sushi, Japanese, 35.50
6	Tacos, Mexican, 8.25
7	Pad Thai, Thai, 14.50
8	Steak, Dinner, 50.00
9	Smoothie, Drinks, 5.75
10	Coffee, Drinks, 2.50
11	<u>Nutella</u> Cookie, Dessert, 3.00

Orders placed saved to order history csv

1	
2	1, Delivery, 55.5, Steak, Dinner, 50.0
3	1, Address:, 123 Maple St
4	1, Delivery, 25.23, Burger, Fast Food, 9.99
5	1, Delivery, 25.23, Fries, Fast Food, 3.49
6	1, Delivery, 25.23, Tomato Salad, Italian, 6.25
7	1, Address:, 123 Maple St
8	2, Takeout, 42.5, Steak, Dinner, 50.0
9	2, Pickup Time:, 60 minutes from now
10	



### **Challenges Faced:**

One of the biggest challenges I faced during this project was ensuring consistent file access across different devices. It was often unclear which file path was being used, leading to difficulties in loading the necessary information. Eventually I solved this using the URI class which I found through research on google and it allowed me to access the .csv files through the target directory. However, by accessing the .csv files through the target directory did not update new changes in the resources folder.

Another part I struggled with was organizing the user interaction logic. I wanted the program to flow naturally depending on who was logged in, and it took many attempts to get the structure and layout to feel right. On top of that, I also ran into smaller challenges such as input errors, price formatting issues, and accidental duplicate outputs.

### **Learning Outcomes:**

Through this project, I developed a stronger understanding of OOP in Java as well as gaining practical experience by organizing my code into a clean structure and how to restructure after unexpected setbacks. I also became more confident debugging and managing a project with multiple user roles and classes.