TastyGo

CS 342 Visual Programming - Project

kingdom of saudi arabia

College of Sciences and Arts

in Rass

Deadline (week 17)

SUPERVISED BY :

D.Amal Alteaimi

STUDENTS :

Shaden Alkhalifah - [redacted]

Rama Alshaya - [redacted]

Wajd Alharbi - [redacted]

Taif Obaid Alharbi - [redacted]

Majd Alharbi - [redacted]

Contents

Objectives

To design a simple and interactive food ordering system with a Java-based graphical user interface (GUI).

To provide different user roles (Admin, Customer, Employee) with tailored features and views.

To allow users to browse restaurants, view categorized menus, and place orders.

To simulate the full order process including cart management, payment, and real-time tracking.

To handle input validation and common user-side exceptions gracefully

2.Introduction

Online food delivery systems have become a vital part of modern lifestyles, especially in fast-paced urban areas. This project simulates a real-world food ordering application called TastyGo, where users can interact with the system based on their roles as customers, admins, or employees. The application is fully developed in Java using Swing components to design an interactive and user-friendly GUI. It is also connected to a MySQL database to handle customer data, login credentials, and future scalability for orders and menu management. The system provides essential features like menu browsing, cart handling, payment simulation, and order tracking..

3.Application Overview

TastyGo is a desktop-based application that imitates the core functionality of real-life food ordering platforms. It supports multiple user roles:

• Customer: Can browse available restaurants, apply filters (location, food type, rating), view menus, manage a cart, proceed to a simulated payment, and track the order status.

• Admin: Can add or remove restaurants and employees via a dedicated admin dashboard.

• Employee: Has a simple interface to access system functionalities related to their role.

•Guest: Can access the system without login credentials, with limited features similar to a customer.

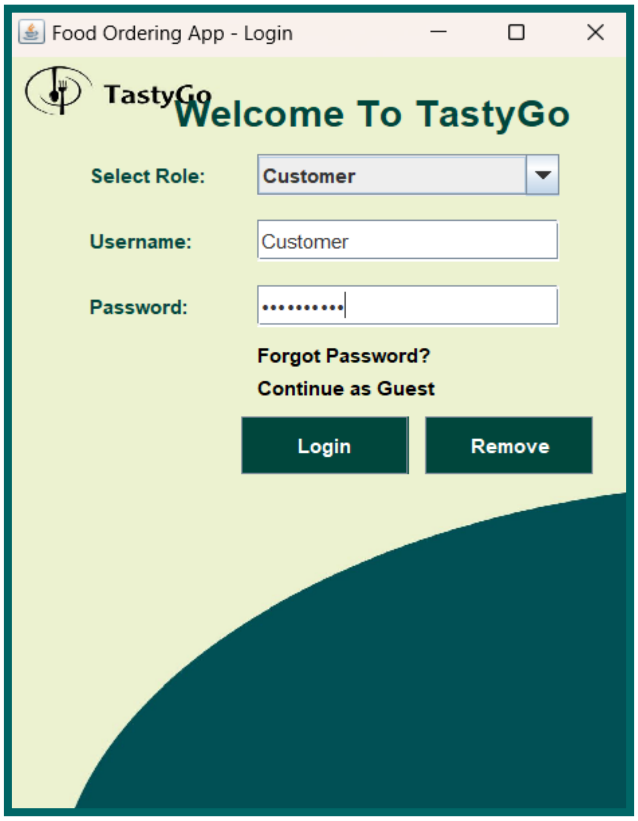
The app includes major real-world features like categorized food types, ratings, dynamic cart updates, payment validation, and live order tracking animations.

4.GUI Elements

The user interface is developed using Java Swing, with background images for enhanced visual experience. Each role has a distinct view, and various GUI components are used throughout the app :

4.1 Login Page

- JComboBox: To select user role.
- JTextField and JPasswordField: For entering login credentials.
- JButton: For login and clear actions.
- JLabel: Clickable labels for "Forgot Password" and "Continue as Guest" options.
- Event Handling: Validates login info, handles role-based redirection, and input errors.


Food Ordering App - Login

TastyGo Welcome To TastyGo

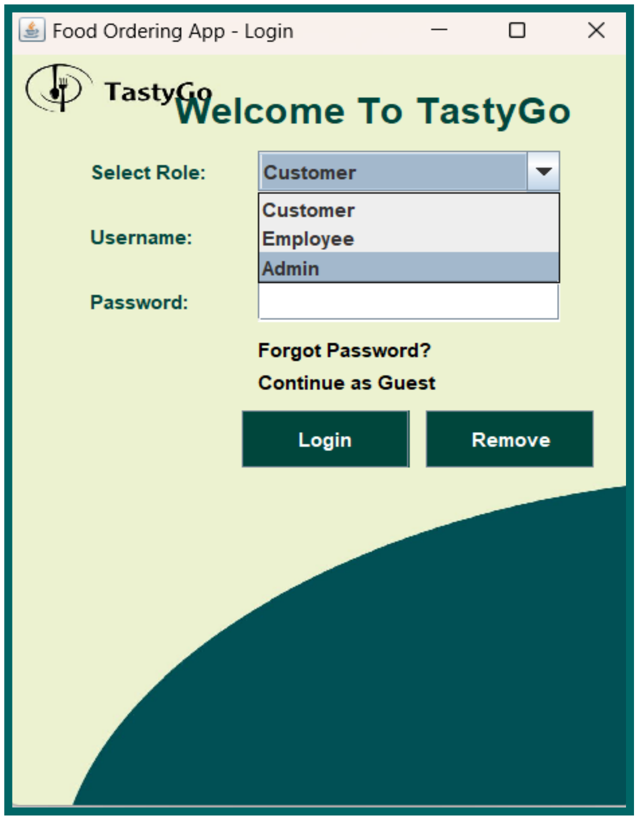
Select Role: Customer

Username: Customer

Password: ••••••••••

Forgot Password?

Continue as Guest


Food Ordering App - Login

TastyGo Welcome To TastyGo

Select Role: Customer

Customer
Employee
Admin

Username:

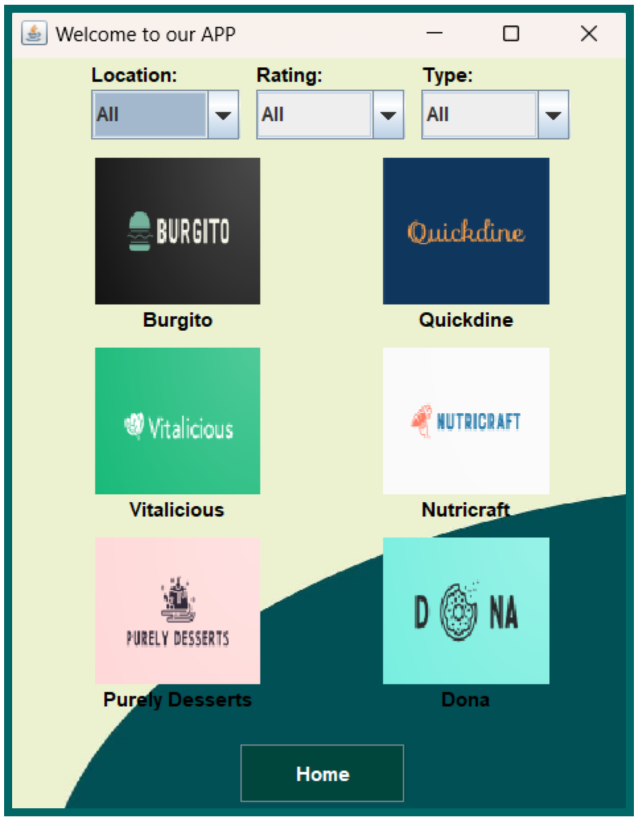
Password:

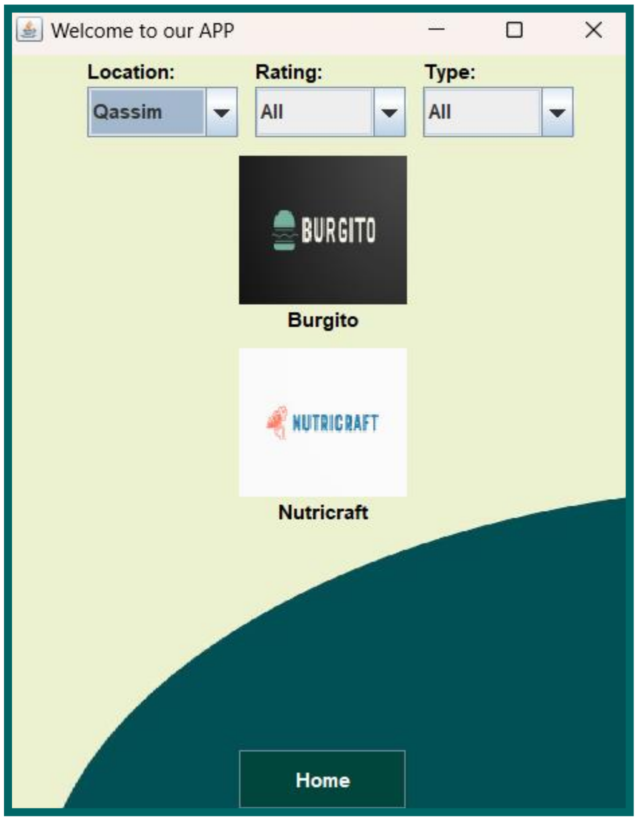
Forgot Password?

Continue as Guest

4.2 Customer Dashboard

- JComboBox: Filters for food category, location, and rating.
- JPanel: Dynamically updated to display restaurant cards with images and clickable names.
- Filtering Logic: Filters work in combination, updating the restaurant view in real time.





4.3 Restaurant Menus

• Each restaurant (e.g., QuickDine or Purely Desserts) has its own menu page.

• Menu items include:

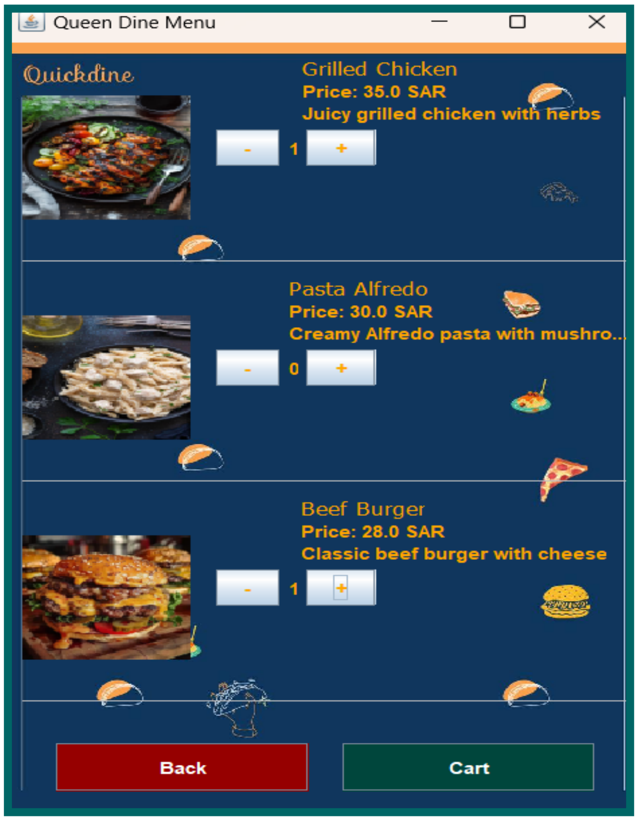
- Image

- Name, description, and price

- + and – buttons to change quantity

- Cart button opens a dialog showing selected items and total cost.

- Proceed to Payment directs the user to the Payment Page.


Queen Dine Menu

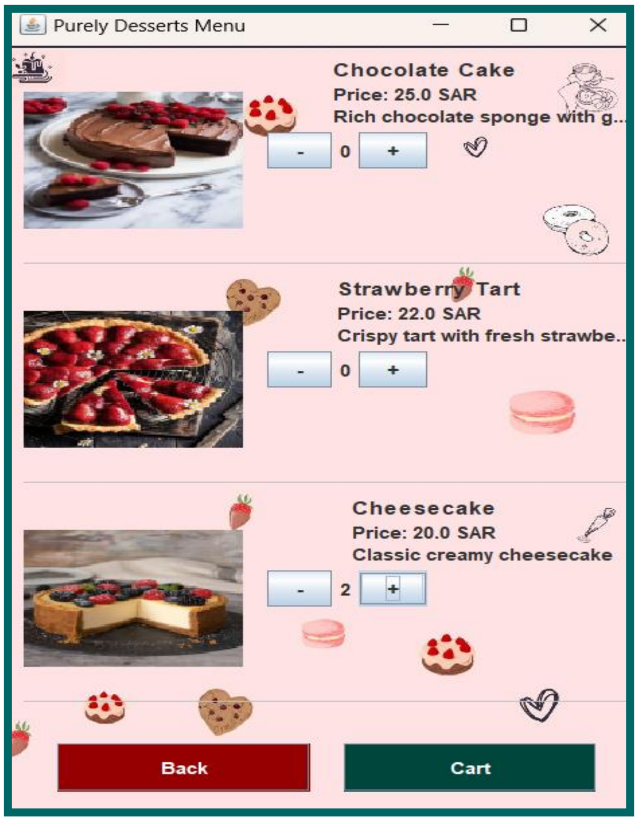
Quickdine

Grilled Chicken
Price: 35.0 SAR
Juicy grilled chicken with herbs
- 1 +

Pasta Alfredo
Price: 30.0 SAR
Creamy Alfredo pasta with mushro...
- 0 +

Beef Burger
Price: 28.0 SAR
Classic beef burger with cheese
- 1 +

Back Cart


Purely Desserts Menu

Chocolate Cake
Price: 25.0 SAR
Rich chocolate sponge with g...
- 0 +

Strawberry Tart
Price: 22.0 SAR
Crispy tart with fresh strawbe...
- 0 +

Cheesecake
Price: 20.0 SAR
Classic creamy cheesecake
- 2 +

Back Cart

4.4 Admin Dashboard

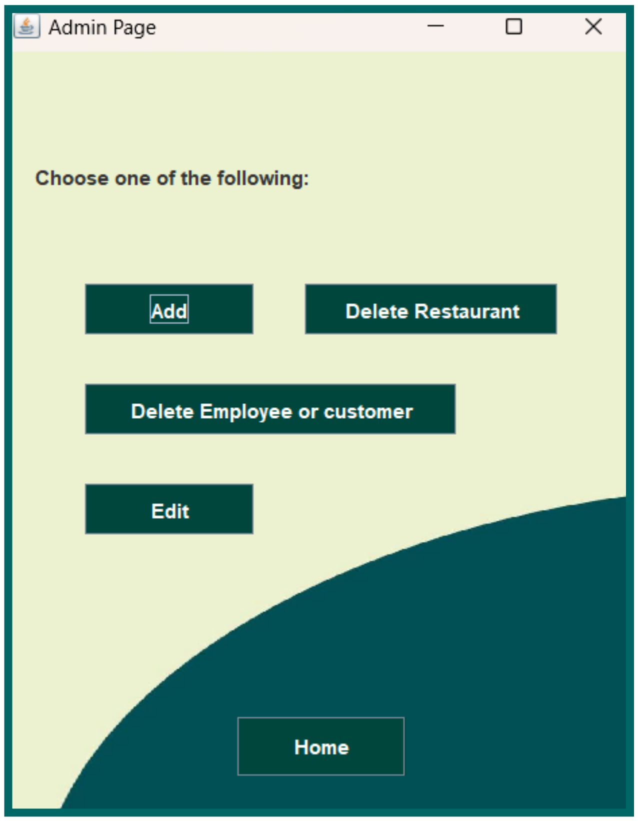
Includes JButtons to Add, Delete, or Edit restaurants and employees.

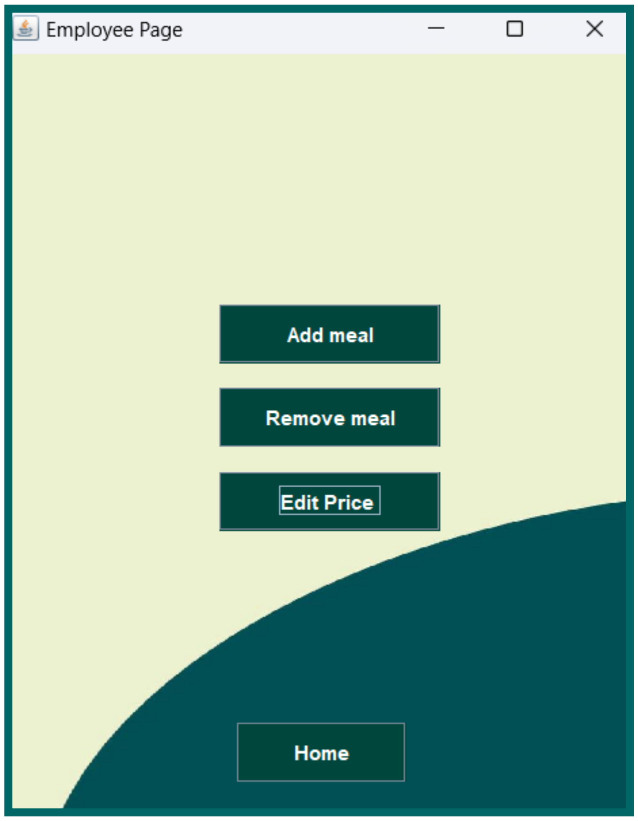
• Sub-windows appear for each action with input fields and confirmation dialogs.

• Clear feedback is shown for invalid actions (e.g., trying to delete a restaurant with no name entered).

4.5 Employee Dashboard

Simple interface with a “Home” button to return to the login screen





4.6 Payment Page

- JTextField: For card number, expiry date, and CVV.
- Input fields use TitledBorder for clarity.
- On payment confirmation, the system transitions to order tracking.

4.7 Order Tracking Page

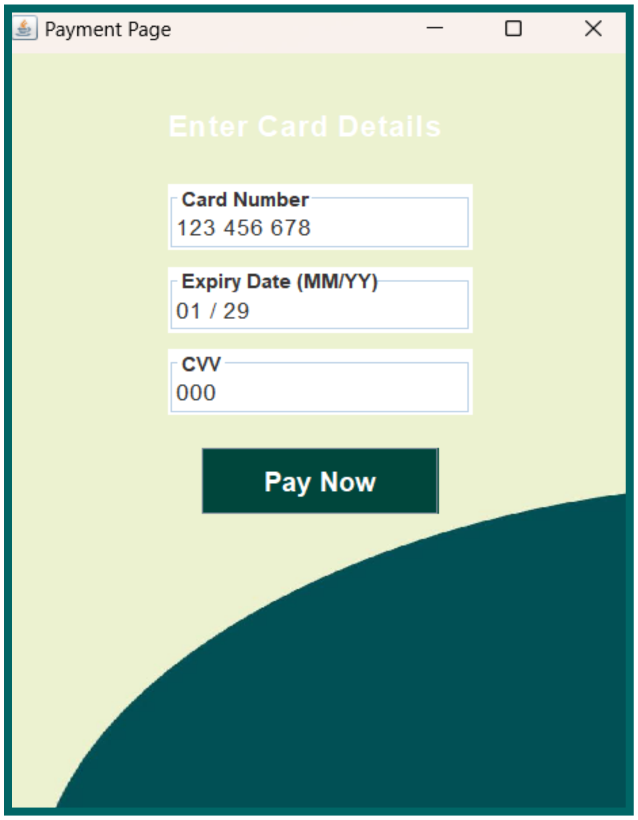
- Progress bar shows real-time updates for:

- Order Confirmation

- Preparation

- Delivery

- Threading is used to simulate time-based updates to the status.

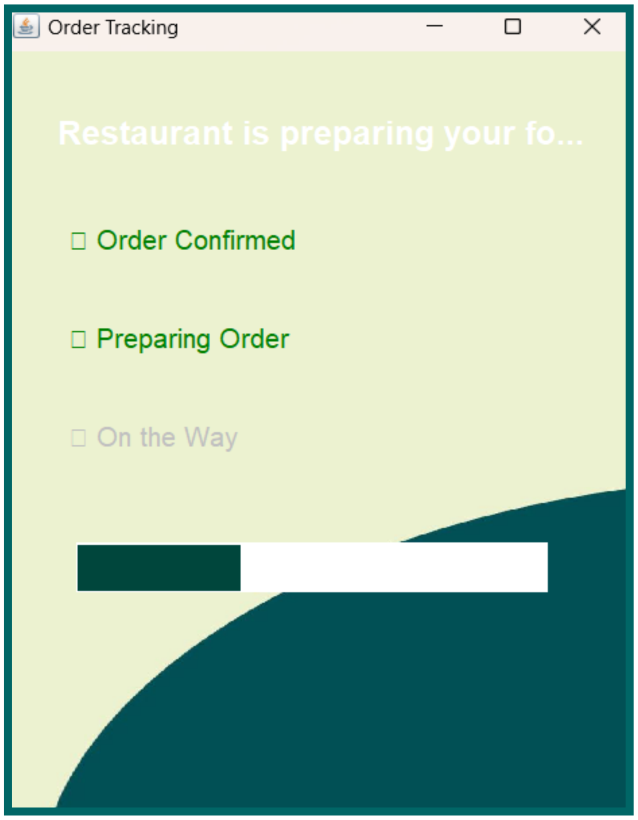

Payment Page

Enter Card Details

Card Number
123 456 678

Expiry Date (MM/YY)
01 / 29

CVV
000


Order Tracking

Restaurant is preparing your fo...

□ Order Confirmed

□ Preparing Order

□ On the Way

5. Database Connection

Step 1: Download MySQL JDBC Driver
To establish a connection between our Java application and a MySQL database, we first downloaded the official JDBC driver:
• We visited: https://dev.mysql.com/downloads/connector/j/
• We downloaded the MySQL Connector/J ZIP file.
• After extracting it, we saved the path to the mysql-connector-java-x.x.x.jar file for later use.

Step 2: Create a Java Project (Using Ant) in NetBeans
• We opened NetBeans and created a new project:
File > New Project.
• We selected:
Java with Ant > Java Application.
• We named the project and chose a suitable location for saving it.

Step 3: Add MySQL JDBC Driver to NetBeans Services
Before connecting our project to the database, it can be helpful to register the MySQL driver in NetBeans Services for testing connections :
• In NetBeans, go to the Services tab.
• Expand the Databases section.
• Right - click on Drivers, then choose New Driver.
• Click Add and locate the extracted mysql - connector - java - x.x.x.jar file.
• Click Next, and give the driver a name(e.g., MySQL JDBC Driver).
• Finish the process and the driver will now appear under the Drivers list.

Step 4: Add JDBC Driver to the Project
To allow our Java application to communicate with MySQL, we added the JDBC driver as follows:

- Right-clicked the project name in the Projects pane.
- Selected Properties.
- Navigated to the Libraries section.
- Clicked on Add JAR/Folder.
- Located and added the mysql-connector-java-x.x.x.jar file.
- Clicked OK to apply the changes

6. Exception Handling

The application includes several layers of input validation and error handling:

Login Page:

• Password must be at least 8 characters and start with a letter.

• Empty fields show error messages.

Payment Page:

• All fields are required; alerts are shown if any are missing.

• Admin Dashboard:

• Confirmation dialogs for deletion.

• Validation to prevent actions with empty inputs.

Cart Dialog:

• Shows warning if the user attempts to check out with an empty cart.

• Order Tracking:

• Uses SwingUtilities.invokeLater() to avoid GUI freezing while simulating progress updates.

7. Testing and Validation

We performed manual testing by simulating the following scenarios:

- Logging in with valid and invalid credentials.
- Logging in with different roles and checking redirection.
- Navigating menus and updating item quantities.
- Proceeding to payment with missing and complete data.
- Tracking order progress after successful payment.
- Admin functions for adding and removing restaurants.
- Error messages under multiple edge cases (empty fields, short passwords, etc.).

All GUI elements functioned correctly and responded well to user input.

8. Challenges and Solutions

• Image Scaling: Loading and resizing images without distortion was challenging. Used Image.SCALE_SMOOTH to ensure quality.

• UI Freezing During Order Tracking: Solved using background threads and SwingUtilities.invokeLater.

• Input Validation: Added multiple checks for better user experience and error prevention.

• Cart Logic: Built logic to calculate totals dynamically based on quantities and to ignore zero-quantity items.

9. Conclusion

The TastyGo application effectively simulates an online food ordering system with interactive GUI, multi-role support, and essential ordering features. While it currently lacks backend database support, the application provides a solid foundation for future development. It meets all core requirements of the CS342 project and demonstrates practical application of Java GUI development, event handling, and exception management