Manuel S. Enverga University Foundation
Lucena City

Granted Autonomous Status
CHED CEB Res. 076-2009

# Data Structures and Algorithms

## Semi-Final Examination

# Jollibee Ordering System: Using QT Creator Application GUI to Implement Stacks and Queues Operations

**Members:**

Deseo, Earl Dane

Domogma, Peter Bob R.

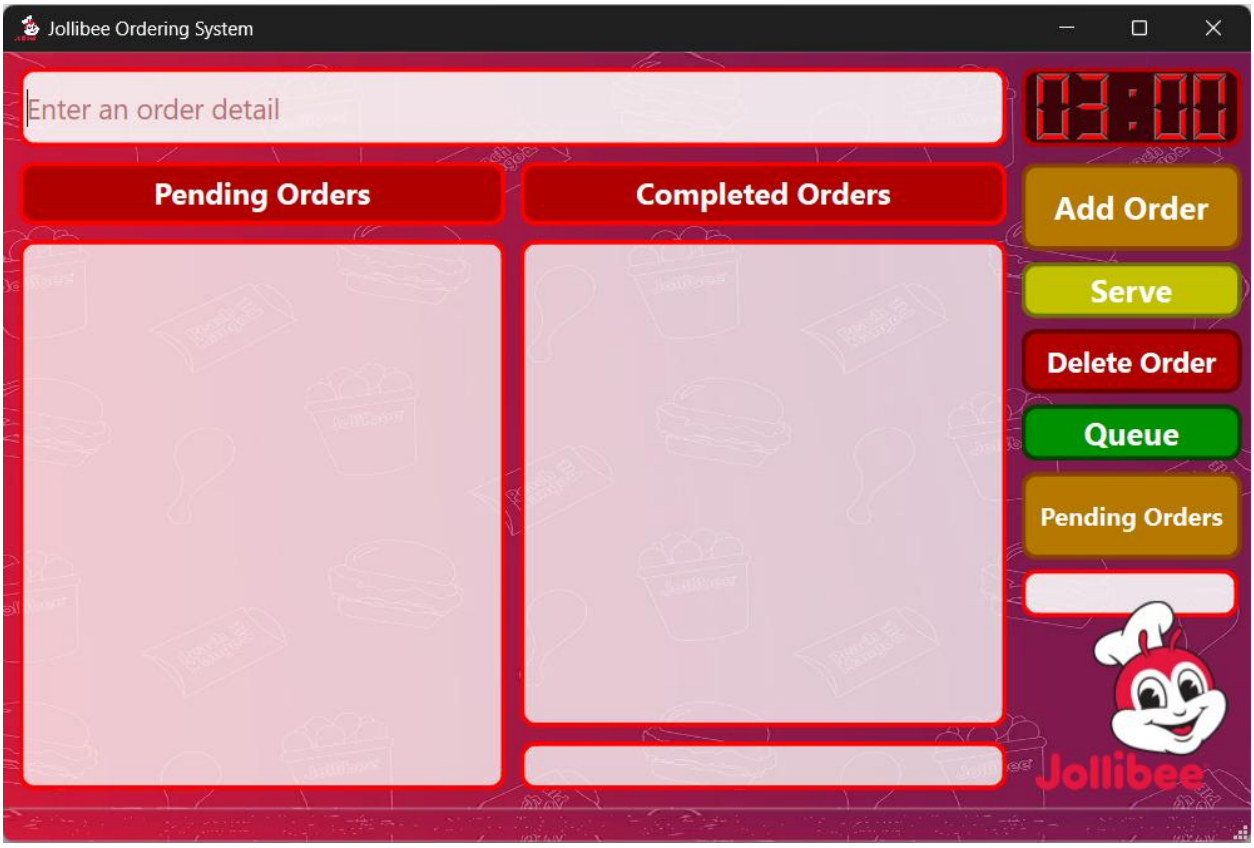
Samsom, Nikko Errol S.

Pabellosa, John Rymell (Cheer dance participant)

**November 2024**

## I. Chosen Application Theme

A Jollibee ordering system management that allows the user to enter and add an order detail that can timed queue for a minute/s to served and completed the orders.




## II. Rationale

The students chose the Jollibee Ordering System as their theme because it clearly demonstrates how stacks and queues work in a simple in a real-life example. Fast-food ordering is something most people are familiar with, loved with and grow up with, making it easier to understand these concepts. In this system, customer orders are added to a stack that queued and served in the order they entered, while completed and canceled by deleting orders are handled using a stack. This idea and concept help the users visualize the practical use of these data structures and pattern in an everyday scenario.

## III. Implemented Application and its Features

This application uses stack and queue data structures to manage customer orders efficiently. Here's a breakdown of the features based on the code provided and GUI below:

Application Features:

1. Adding Orders:
    - Users can add new orders through an input field.


    - Each order is automatically assigned a unique order number and added to the pending orders list (queue).


    - The system uses the FIFO (First-In-First-Out) principle for serving orders.

2. Serving Orders (Queue Dequeue Operation):
    - The system serves orders in the order they were placed, moving them from the pending list to the completed list.

- This queue where the first customer in line gets served first of course.

3. Timer Functionality:
   - A timer starts for each order, counting down from 3 minutes (configurable and editable).


   - If the time runs out and there are pending orders, the next order is automatically served.

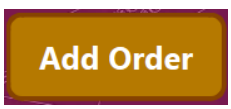
4. Deleting Orders:
   - Users can delete any order from the pending orders list, which removes it from the queue.


5. Updating Order Counts:
   - The system dynamically updates the count of pending and completed orders and displays this information.

6. Simple Stack in Completed or Recent Orders:
    - Completed orders are shown using a stack concept, allowing users to track the most recently served orders (LIFO - Last-In-First-Out), helpful for managing canceled orders or reviewing the latest transactions.


**UI Layout Design:**

1. Main Window Layout:
    - Right Panel:
        - Order Input Section:
            - A button named "Add Order" to add the order to the queue or pending orders.

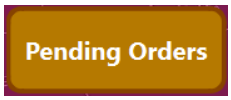


- Control Buttons:
  - "Queue" (Starts the timer for the first order).


  - "Serve" (Manually serves the next order).


  - "Delete Order" (Deletes the selected order).


  - “Pending Orders” List (Display the total pending orders):



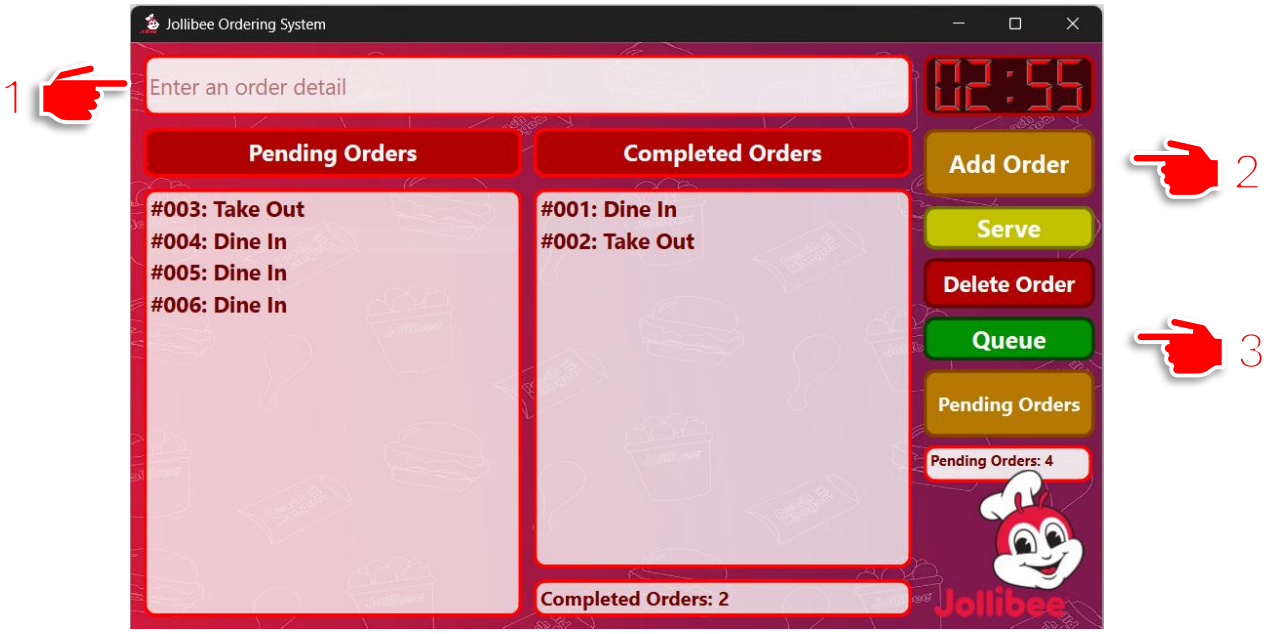

2. Top Panel:
   - Timer Display:
     - An LCD-style timer showing the countdown for the current order in the format.


3. Bottom Panel:
   - Order Counters:
     - Labels for showing the total number of pending and completed orders.

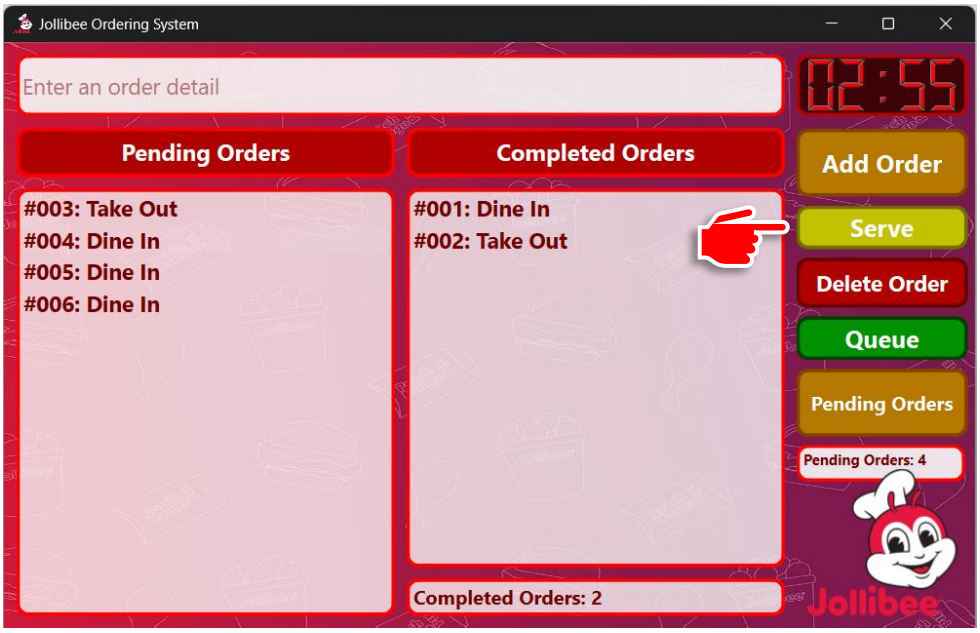

## IV. Three Test Cases for the Chosen Application

### A. Test Case 1: Functionality Test.

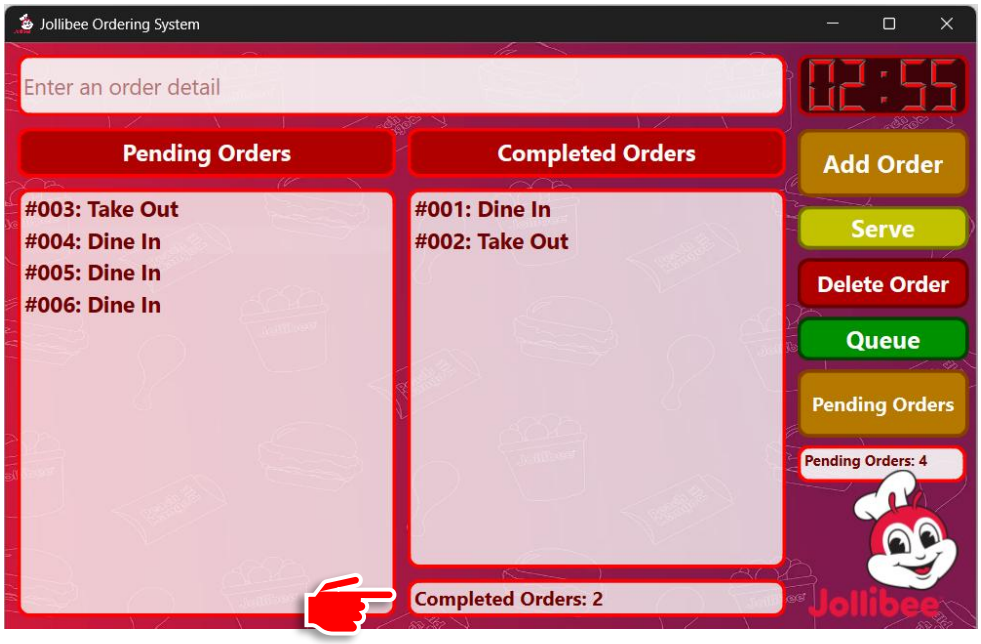
1. Entering an order detail into the Pending Orders list to initiate the queue process, allowing the orders to progress to the Completed Orders list.



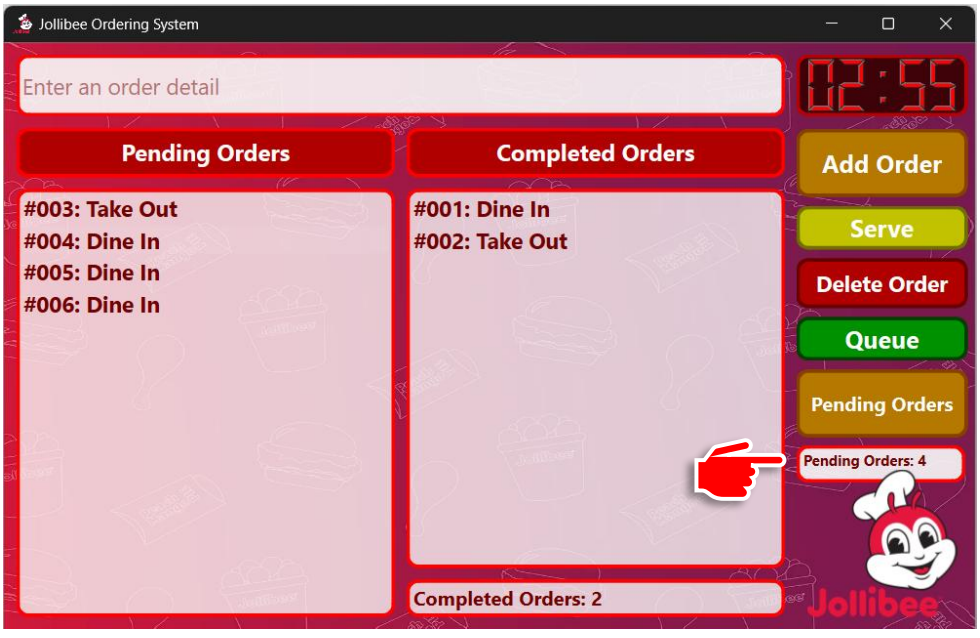

2. Enable the user to serve the orders to be completed.



3. Allowing the user to track the total count of Completed Orders.

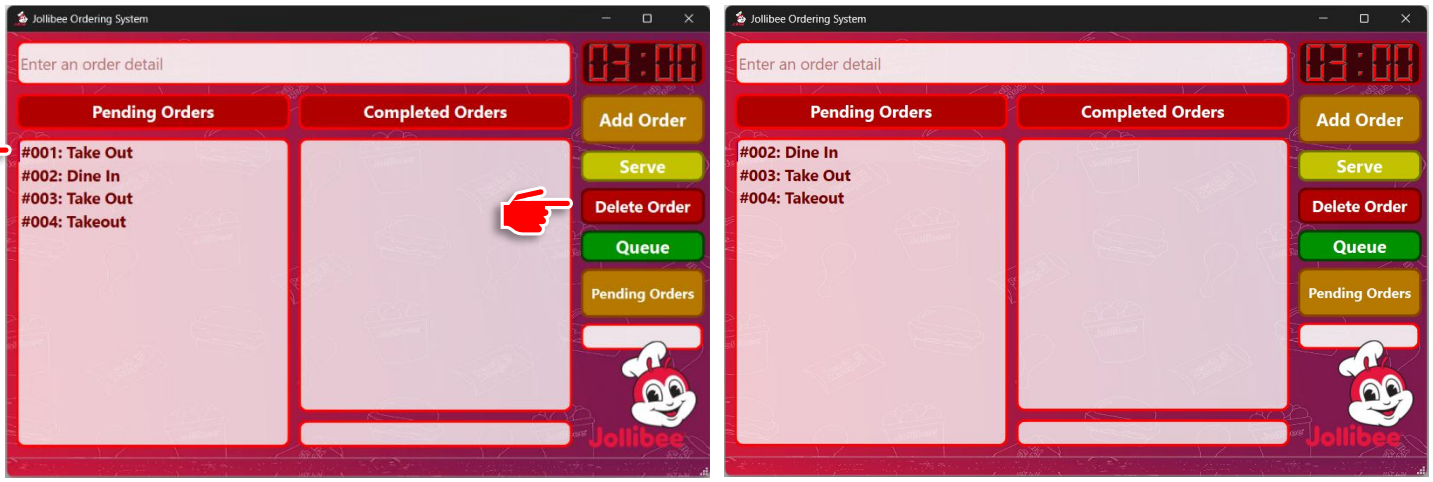



4. Allowing the user to track the total count of Pending Orders.



**B. Test Case 2: Deleting Test (Pop Operation)**

1. Allowing the user to remove or delete the selected items/orders.




**C. Test Case 3: Queue timed out serving.**

**1. Entering more orders to the queue to served after the queue timed out.**

## V. Challenges Faced during development

### A. Development

- **Stack and Queue implementation -** Implementing the logic to manage order states (Pending, Queued, and Completed) efficiently.
- **No crushes and bug -** Ensuring that orders are not lost during application restarts or crashes and bugged during the process.
- **Debugging – some widgets are not in the right labeled and wrong spelling that leads to confusion and bug/errors.**

### B. Designing

- **QT CSS Style Sheet** – Aligning with Jollibee's theme that's leads to some confusion while design in CSS style sheet but it's kind of fun too.
- **Color Palette -** Designing a visually appealing and user-friendly coloring choice that aligns with Jollibee's theme and coloring.
- **Image lay outing –** Lay outing the image, logos. Background image in the right current layout and position.
- **Location file path –** Getting a hard time to find the right path of the image sources.

## VI. Roles and contributors

**DEVELOPER:** Deseo, Earl Dane – implemented the source code operations and functions, UI layout

**DESIGNER:** Domogma, Peter Bob R. – GUI Design and UI layout, sources files/images, header layout, Documentation and header code

**TESTER:** Samson, Nikko Errol S. – Documentation, testing the functionality of the application, provide ideas and feedbacks.