

Online Food Ordering System

Data Structures and Algorithms Project

Submitted by: Abdullah Mudassar

University Project Documentation

Submission Date: September 2023

Table of Contents

1. Abstract
2. Problem Statement
3. Proposed Solution
4. Objectives
5. System Requirements
6. Flowchart of the Online Food Ordering System
7. Data Structures Used
8. Implementation
9. Sample Output
10. Testing
11. Complexity Analysis
12. Limitations
13. Future Enhancements
14. Learning Outcomes
15. Acknowledgment
16. Conclusion
17. References

Abstract

This project implements an Online Food Ordering System using C++. The purpose is to apply core data structures such as Binary Search Trees, Queues, Stacks, and Linked Lists in a practical application. The system allows users to browse menu items, search availability, place orders, cancel orders, and track delivery history efficiently.

Problem Statement

Traditional food ordering processes are often manual, time-consuming, and error-prone. They lack efficient mechanisms to manage orders, track delivery, and handle cancellations. This creates delays and reduces customer satisfaction.

Proposed Solution

The Online Food Ordering System addresses these problems by applying efficient data structures. Menu items are managed using a Binary Search Tree, active orders are handled with a Queue, cancellations with a Stack, and delivery history with a Linked List. This ensures that order processing, tracking, and retrieval are efficient.

Objectives

- Apply data structures to a real-world style project.
- Build a console-based system in C++.
- Enable efficient searching, ordering, and order cancellation operations.
- Demonstrate practical understanding of DSA concepts.

System Requirements

Software: C++ Compiler (Code::Blocks, VS Code, or Visual Studio)

Hardware: PC with minimum 4GB RAM

Flowchart of the Online Food Ordering System

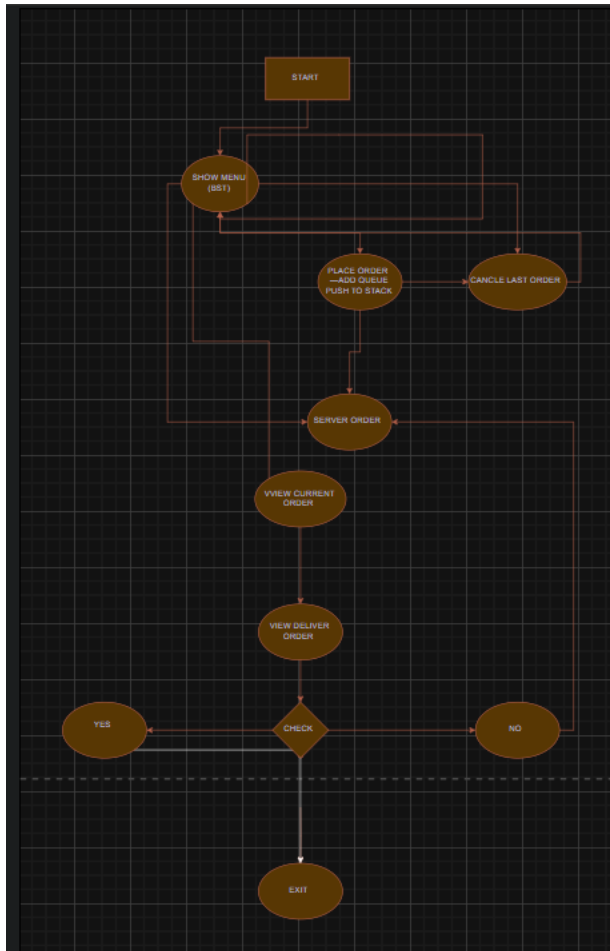


Figure: Flowchart showing the overall program flow, including menu display, order placement, cancellation, and exit.

Data Structures Used

Data Structure

Usage

Binary Search Tree (BST)

Store and search menu items

Queue

Manage active orders (FIFO)

Stack

Handle last order cancellations (LIFO)

Linked List

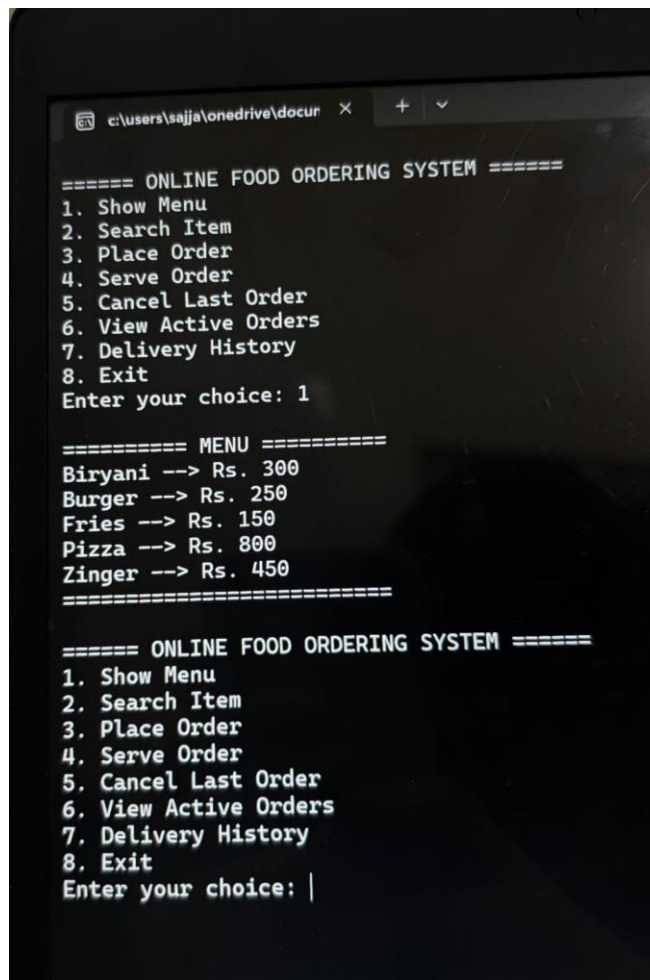
Maintain delivery history

Implementation

The project is implemented in C++ using OOP principles. Key components include:

- MenuSystem class (BST for storing items)
- OrderQueue class (Queue for managing orders)
- CancelStack class (Stack for cancellations)
- DeliveryHistory class (Linked List for deliveries)

Sample Output



```
c:\users\sajja\onedrive\docur x + v

===== ONLINE FOOD ORDERING SYSTEM =====
1. Show Menu
2. Search Item
3. Place Order
4. Serve Order
5. Cancel Last Order
6. View Active Orders
7. Delivery History
8. Exit
Enter your choice: 1

===== MENU =====
Biryani --> Rs. 300
Burger --> Rs. 250
Fries --> Rs. 150
Pizza --> Rs. 800
Zinger --> Rs. 450
=====

===== ONLINE FOOD ORDERING SYSTEM =====
1. Show Menu
2. Search Item
3. Place Order
4. Serve Order
5. Cancel Last Order
6. View Active Orders
7. Delivery History
8. Exit
Enter your choice: |
```

Testing

Test Case	Input	Expected Output	Actual Output	Result
Search unavailable item	Sandwich	Item not available	Item not available	Pass
Cancel with no orders	Cancel	No order to cancel	No order to cancel	Pass

Complexity Analysis

- BST Search: $O(\log n)$ average, $O(n)$ worst case
- Queue Enqueue/Dequeue: $O(1)$
- Stack Push/Pop: $O(1)$
- Linked List Traversal: $O(n)$

Limitations

- Console-only system
- Data not persistent after exit
- Supports single restaurant menu

Future Enhancements

- Add file/database storage
- Implement GUI for usability
- Expand to multiple restaurants

Learning Outcomes

- Practical application of data structures
- Improved C++ coding and debugging skills
- Understanding of system design and modular programming

Acknowledgment

I thank my instructors and peers for their guidance and support in completing this project.

Conclusion

The Online Food Ordering System demonstrates how core data structures can be applied in building real-world applications. Through BST, Queue, Stack, and Linked List, operations such as searching, ordering, and tracking were efficiently implemented.

References

Foodpanda, 'How Online Food Ordering Works,' [Online].

Uber Eats, 'Technology Behind Food Delivery,' [Online].

IEEE Transactions on Software Engineering, Efficient Data Structures for Online Food Ordering Systems, 2024.

GitHub Repository: <https://github.com/AbdullahMudassar/Online-Food-Ordering-System-DSA.git>