Vivekanand Education Society's Institute Of Technology

Department Of Information Technology

DSA mini Project

A.Y. 2025-26

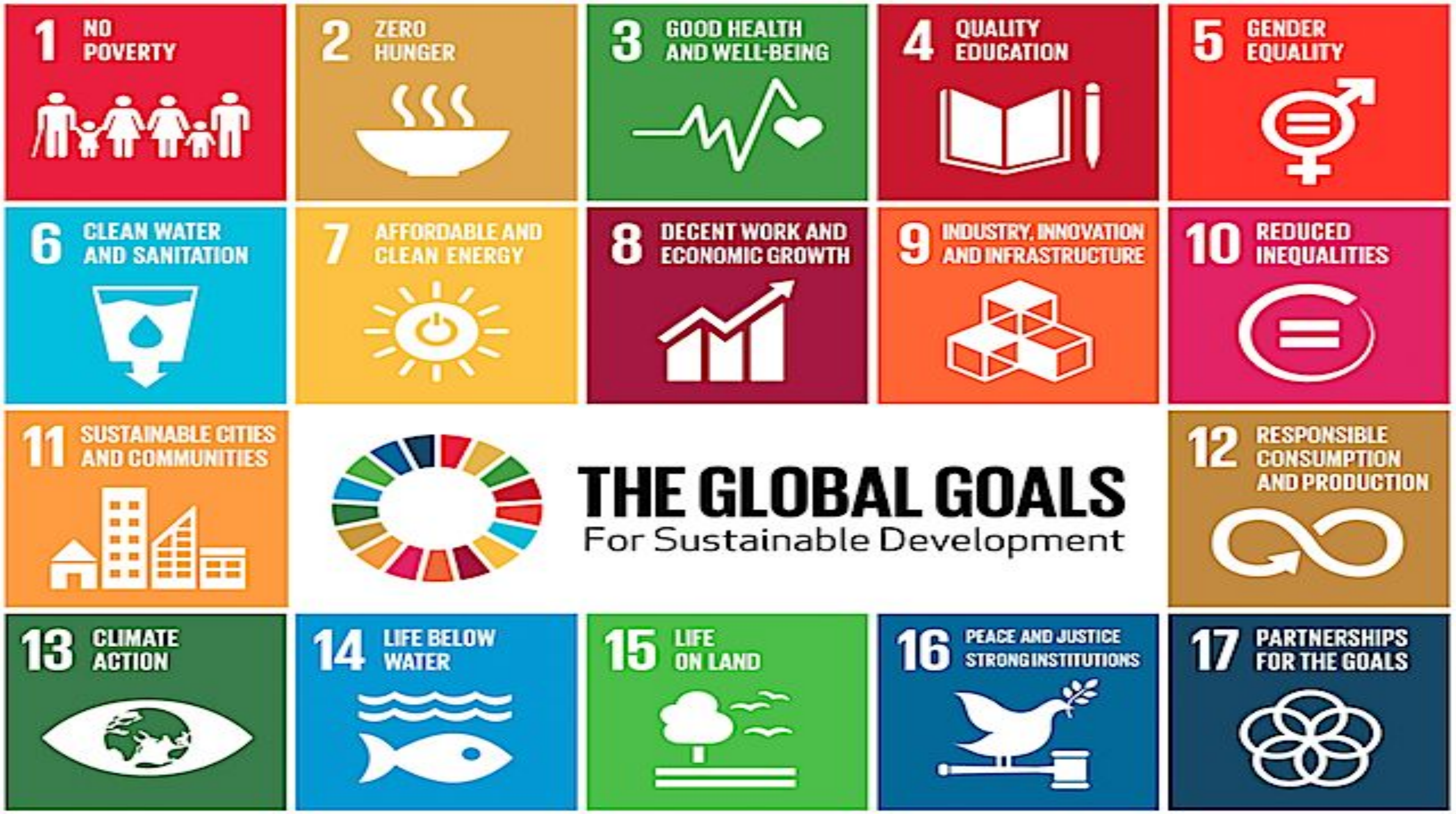
Title:

Sustainability Goal :

Domain: Data Structures and Algorithms

Member: Krishna Gavali

Mentor Name: Kajal Jewani


1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
THE GLOBAL GOALS
For Sustainable Development
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE AND JUSTICE STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS

Content

Introduction to Project

The **Bus Ticket Management System** is a console-based application developed in Java that helps automate the process of booking, canceling, and managing bus tickets.
It efficiently uses **Data Structures** like **Linked List**, **Queue**, **Lists** and **Stack** to handle seat allocation, waiting lists, and cancellation history dynamically.

The system mimics a small-scale bus reservation process and even includes a basic **prediction module** that estimate show soon the bus will be full and predicts the chance of a waiting-list passenger getting a seat — adding an element of automation and intelligence.

Problem Statement

In traditional or manual ticket booking systems:

- Managing passenger data manually is slow and error-prone.
- Tracking available seats and waiting lists becomes difficult during peak hours.
- No real-time update or prediction of bus capacity exists.

Hence, there is a need for a **simple, data-structure-driven system** that can:

1. Dynamically allocate and free seats,
2. Maintain waiting lists automatically,
3. Predict bus occupancy trends.

Objectives of the project

- To automate bus ticket booking and cancellation using efficient data structures.
- To maintain a dynamic waiting list for passengers when seats are full.
- To allow **undo** operations for recent cancellations using a stack.
- To display current bookings and waiting passengers in real time.
- To implement a simple **predictive algorithm** that estimates:

  a. Time until the bus is full, and

  b. Chances for someone getting a seat who is in waiting list.

Requirements of the system (Hardware, software)

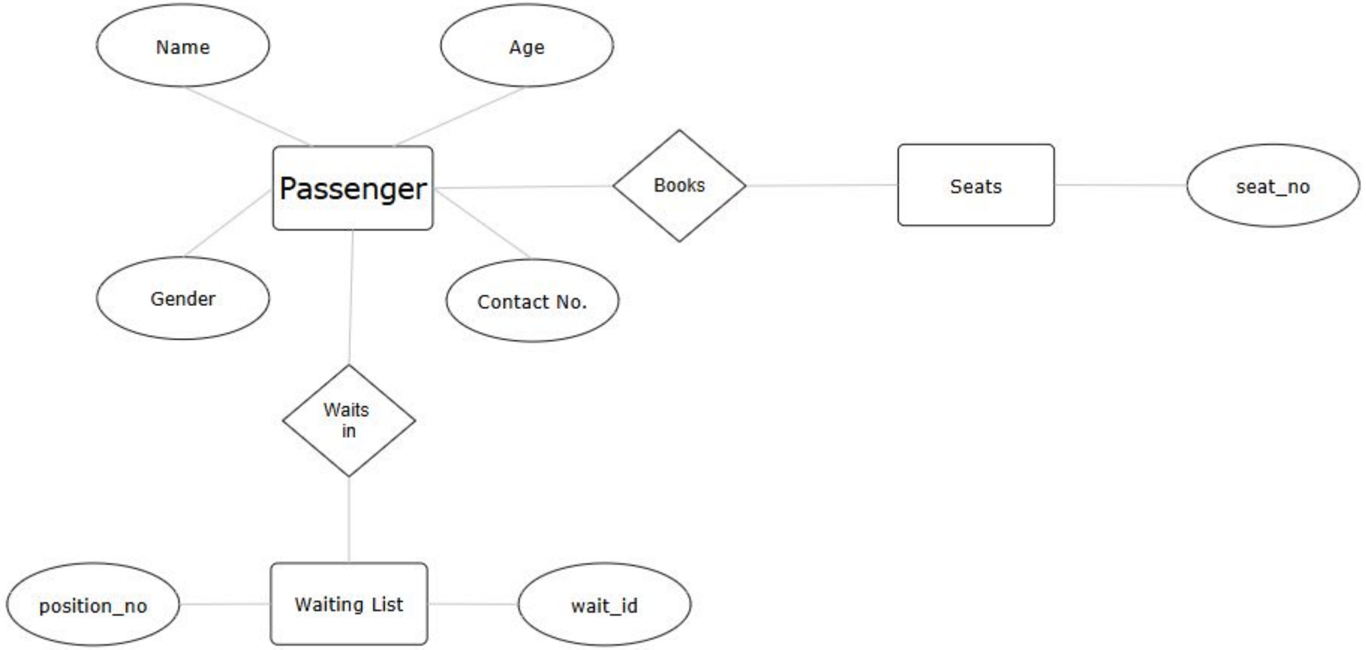
Hardware Requirements-

- Processor - Intel i3 or higher
- RAM- 4 gb
- Storage- 100 mb of free space
- Display- standard console

Software Requirements-

- Operating System- Windows / macOS / Linux
- Language- JDK 8 or higher
- IDE / Compiler- intelliJ / vs code
- Libraries- Java Collections Frameworks

ER diagram of the proposed system



Data Structures and Concepts used

1. Linked List - To store confirmed passengers.
2. Queues - To manage waiting list.
3. Stack - Stores cancel history.
4. ArrayList - Stores booking and cancellation timestamps (used for prediction algorithm).
5. Algorithmic Concepts - Linked List traversal, Queue/Stack operations, prediction logic.

Algorithm Explanation

1. Booking - If seats are full then add the passenger to waitingList queue or add to end of linked list.

2. Cancellation - Remove that passenger’s node and push it into cancel history stack

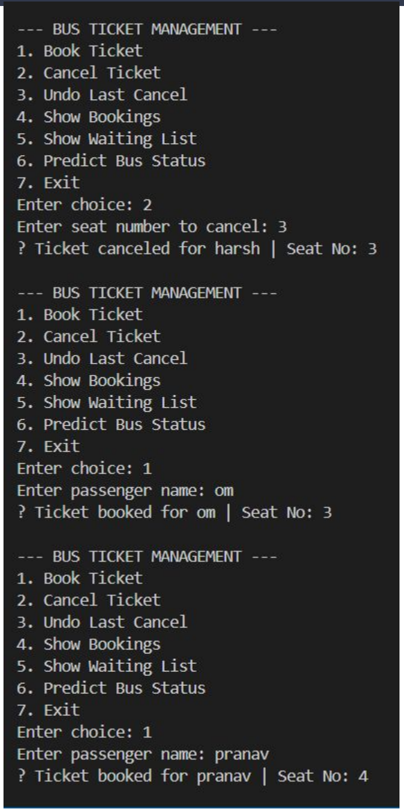
3. Undo cancellation - remove last from cancelHistory stack and re-book swing booking.

4. Prediction Algorithm - calculates the average booking interval to estimate how soon the bus will be full and predicts the chance of a waiting-list passenger getting a seat based on past cancellations and waiting list size.

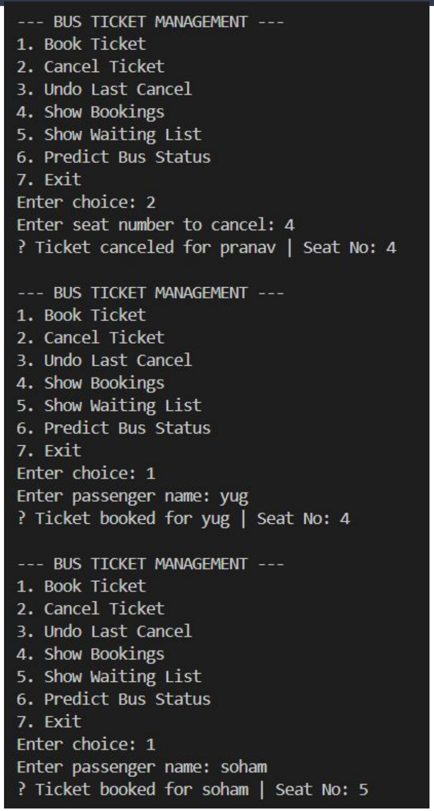
Time and Space Complexity

Operation	Time Complexity	Space Complexity	Explanation
Book Ticket	O(n)	O(n)	Traverses linked list to insert at end
Cancel Ticket	O(n)	O(n)	Searches linked list to remove passenger
Undo Cancel	O(1)	O(1)	Stack pop + reinsert
Add to Waiting list	O(1)	O(n)	Queue enqueue operation
Prediction Calculation	O(n)	O(1)	Looping through bookingTimes once

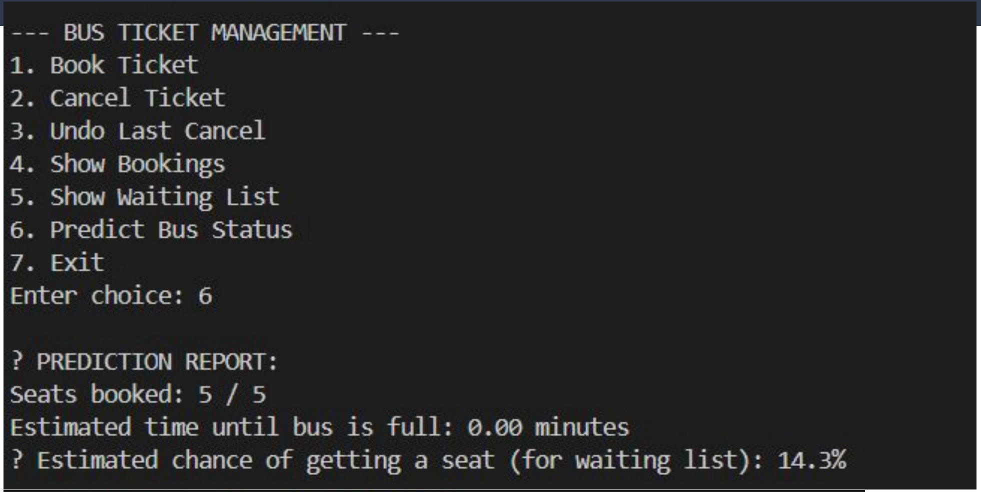
Output



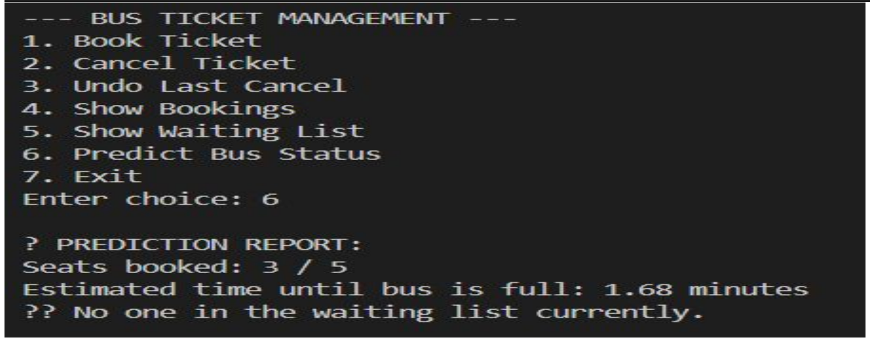
```
--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 2
Enter seat number to cancel: 3
? Ticket canceled for harsh | Seat No: 3

--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 1
Enter passenger name: om
? Ticket booked for om | Seat No: 3

--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 1
Enter passenger name: pranav
? Ticket booked for pranav | Seat No: 4
```



```
--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 2
Enter seat number to cancel: 4
? Ticket canceled for pranav | Seat No: 4

--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 1
Enter passenger name: yug
? Ticket booked for yug | Seat No: 4

--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 1
Enter passenger name: soham
? Ticket booked for soham | Seat No: 5
```



```
--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 6

? PREDICTION REPORT:
Seats booked: 5 / 5
Estimated time until bus is full: 0.00 minutes
? Estimated chance of getting a seat (for waiting list): 14.3%
```



```
--- BUS TICKET MANAGEMENT ---
1. Book Ticket
2. Cancel Ticket
3. Undo Last Cancel
4. Show Bookings
5. Show Waiting List
6. Predict Bus Status
7. Exit
Enter choice: 6

? PREDICTION REPORT:
Seats booked: 3 / 5
Estimated time until bus is full: 1.68 minutes
?? No one in the waiting list currently.
```

Future Scope

- Database integration (MySQL/SQLite) for permanent storage
- Graphical User Interface for ease of use
- Give location of the bus they have booked
- Integration with SMS/email notification system

Conclusion

The **Bus Ticket Management System** automates seat booking, cancellations, and waiting lists using **Linked Lists, Queues, and Stacks**.
It efficiently manages passenger data and introduces a basic prediction feature to estimate bus occupancy.
This project proves how core Data Structures can be applied to solve real-life problems effectively.
It is simple, efficient, and can be expanded with features like multiple buses, databases, or a GUI in the future.

References

- Data Structures Using C – Reema Thareja
- GeeksforGeeks: https://www.geeksforgeeks.org – Linked List, Stack, Queue implementation in C
- Stack Overflow: https://stackoverflow.com