System Requirements Specification Index

For

Music Festival Console

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

Pizza Shop Calculator Console
System Requirements Specification

1 PROJECT ABSTRACT

Due to a surge in international artists touring India, BookMyShow, India's leading entertainment ticketing platform, requires a robust solution to efficiently manage their concert venue operations. The Concert Management System is a Python console application designed to streamline the complex task of managing large-scale concert events. This system helps venue managers track real-time ticket sales, monitor seating availability, and calculate revenue across premium, middle, and rear seating zones. The system performs critical functions including dynamic pricing calculations for different seating zones, real-time tracking of seat occupancy and availability, efficient management of seating arrangements, and generation of comprehensive sales and occupancy statistics. By automating these calculations and providing instant analytics, the system ensures accurate tracking for both sales monitoring and seating management, helping BookMyShow maintain its high standards of service for both artists and concert-goers.

BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	1. User needs to track ticket sales and calculate revenueThe application should handle discounts and split billing 2. Application should monitor available seats and occupancy 3. Console should handle different calculations using various arithmetic operators:- Multiplication (*) for calculating zone revenue, Addition (+) for combining total revenue, Subtraction (-) for calculating remaining seats, Division (/) for calculating occupancy percentages, Floor Division (//) for calculating complete rows, Modulus (%) for calculating remaining seats in rows

2 CONSTRAINTS

2.1 INPUT REQUIREMENTS

1. Zone A Tickets:
    - **Must be stored as integer in variable zone_a_sold**
    - **Must be non-negative and not exceed 200**
    - **Example: 150**
2. Zone B Tickets:
    - Must be stored as integer in variable zone_b_sold
    - Must be non-negative and not exceed 300
    - Example: 200
3. Zone C Tickets:
    - Must be stored as integer in variable zone_c_sold
    - Must be non-negative and not exceed 500
    - Example: 350
4. Zone Capacities:
    - Must be stored as integer in quantity
    - Must be stored as constants ZONE_A_CAPACITY = 200
    - Must be stored as constants ZONE_B_CAPACITY = 300
    - Must be stored as constants ZONE_C_CAPACITY = 500
    - Must be stored as constants SEATS_PER_ROW = 20
5. Zone Prices:
    - Must be stored as constants ZONE_A_PRICE = 5000
    - Must be stored as constants ZONE_B_PRICE = 3000
    - Must be stored as constants ZONE_C_PRICE = 1500

2.2 CALCULATION CONSTRAINTS

1. Revenue Calculation:
    - Use multiplication (*) for zone revenue
    - Use addition (+) for total revenue

- Store in total_revenue
- Example: (150 * 5000) + (200 * 3000) + (350 * 1500)

2. Remaining Seats:
   - Use multiplication (*) and division (/) for percentage
   - Use subtraction (-)
   - Store results in zone_a_left, zone_b_left, zone_c_left
   - Example: 200 - 150 = 50 seats left
3. Occupancy Percentage:
   - Use multiplication (*) and division (/)
   - Store results in a_occupancy, b_occupancy, c_occupancy
   - Example: (150 * 100) / 200 = 75.0%
4. Row Distribution:
   - Use floor division (//) for complete_rows
   - Use modulus (%) for remaining_seats
   - Example: 150 // 20 = 7 rows, 150 % 20 = 10 seats

2.3 Output Constraints

1. Display Format:

- Show revenue with ₹ symbol and comma separators
- Show occupancy percentages with one decimal place
- Each value must be displayed on a new line
- Use simple section headers with hyphen separators - Group information by zone

2. Required Output Format:
   - Must show exactly in this order:
   - Show "Sales Summary"
   - Show "Total Revenue: ₹{value}"
   - Show "Seating Status"
   - Show "Zone A:"
     - Show "Remaining Seats: {value}"
     - Show "Occupancy: {value}%"
     - Show "Complete Rows: {value}"
     - Show "Extra Seats: {value}"

- Show "Zone B:"
  - Show "Remaining Seats: {value}"
  - Show "Occupancy: {value}%"
  - Show "Complete Rows: {value}"
  - Show "Extra Seats: {value}"
- Show "Zone C:"
  - Show "Remaining Seats: {value}"
  - Show "Occupancy: {value}%"
  - Show "Complete Rows: {value}"
  - Show "Extra Seats: {value}"

4. TEMPLATE CODE STRUCTURE:

1. Calculation Functions:

- calculate_ticket_revenue() [*, +]
- calculate_seats_remaining() [-]
- calculate_zone_occupancy() [*, /]
- calculate_seats_per_row() [//, %]

2. Input Section:

- Get Zone A tickets sold (int)
- Get Zone B tickets sold (int)
- Get Zone C tickets sold (int)

3. Conversion Section:

- Calculate total revenue
- Calculate remaining seats
- Calculate occupancy percentages
- Calculate row distributions

4. Output Section:

- Display total revenue
- Display seating status for each zone
- Format all numbers appropriately

5. EXECUTION STEPS TO FOLLOW:

1. Run the program
2. Enter Zone A tickets sold
3. Enter Zone B tickets sold

4. Enter Zone C tickets sold
5. View Sales Summary showing total revenue
6. View Seating Status showing:
    - Remaining seats per zone
    - Occupancy percentage per zone
    - Row distribution per zone