

ROUND 3 Expected Questions

TASK 1:

Problem Statement

- You are tasked with developing a simple railway reservation system for Zoho Railway. The system should support the following functionalities at Level 3:

(i) Book a Ticket:

Input: User's name, age, preferred berth type (Lower, Middle, Upper), and the number of tickets.

Constraints:

- A single user cannot book more than 5 tickets at a time.
- The system should prioritize booking the preferred berth type if available.
- If the preferred berth type is not available, the system should book the next available berth.
- If no berths are available, the system should add the user to a waiting list.

(ii) Cancel a Ticket:

Input: User's name and booking ID.

Constraints:

- The system should remove the user's booking and update the availability of the respective berths.
- If there are users on the waiting list, the system should allocate the newly available berths to the users in the order they were added to the waiting list.

(iii) View Available Berths:

Output: A list of available berths categorized by berth type (Lower, Middle, Upper).

(iv) View Booked Tickets:

Output: A list of all booked tickets including user details (name, age, berth type, and booking ID).

(v) View Waiting List:

Output: A list of users on the waiting list including user details (name, age, and preferred berth type).

Detailed Requirements

Booking a Ticket

- The system should allow users to book tickets by specifying their name, age, preferred berth type, and the number of tickets they wish to book.
- The system should check for the availability of the preferred berth type.
- If the preferred berth type is available, the system should allocate it to the user.
- If the preferred berth type is not available, the system should allocate the next available berth type.
- If no berths are available, the system should add the user to a waiting list.
- The system should generate a unique booking ID for each booking.

Canceling a Ticket

- The system should allow users to cancel their tickets by providing their booking ID.
- Upon cancellation, the system should update the availability of the berths.
- If there are users on the waiting list, the system should allocate the newly available berths to the waiting list users based on their preferences and the order they were added.

Viewing Available Berths

- The system should display the number of available berths for each berth type (Lower, Middle, Upper).

Viewing Booked Tickets

The system should display a list of all booked tickets with the following details:

- User's name
- User's age
- Berth type
- Booking ID

Viewing Waiting List

The system should display a list of users on the waiting list with the following details:

- User's name
- User's age
- Preferred berth type

Constraints

- The total number of berths is fixed and categorized into Lower, Middle, and Upper berths.
- The system should handle multiple users booking and canceling tickets concurrently.

- The waiting list should have a maximum capacity. If the waiting list is full, the system should notify the user that no more bookings can be taken at the moment.

Example Scenarios

Scenario 1: Booking a Ticket

Input: Name: Alice, Age: 30, Preferred Berth: Lower, Number of Tickets: 2

Output: Booking successful, Booking ID: 101, Berths Allocated: Lower

Scenario 2: Canceling a Ticket

Input: Name: Alice, Booking ID: 101

Output: Cancellation successful, Berths updated

Scenario 3: Viewing Available Berths

Output:

Lower: 10

Middle: 15

Upper: 20

Scenario 4: Viewing Booked Tickets

Output:

Name: Bob, Age: 25, Berth: Upper, Booking ID: 102

Name: Charlie, Age: 40, Berth: Middle, Booking ID: 103

Scenario 5: Viewing Waiting List

Output:

Name: Dave, Age: 35, Preferred Berth: Lower

Name: Eve, Age: 29, Preferred Berth: Upper

Assumptions

- There are no class distinctions (e.g., First Class, Second Class) in this system.
- Each berth can only accommodate one person.
- The system maintains the integrity and consistency of booking and cancellation operations.

TASK 2:

Problem Statement

You are tasked with developing a taxi booking system for Zoho. The system should support the following functionalities:

(i) Book a Taxi:

Input: User's name, pickup location, drop location, and booking time.

Constraints:

- The system should assign the nearest available taxi to the user.
- If multiple taxis are at the same distance, the one that becomes available the earliest should be assigned.
- If no taxis are available, the system should add the user to a waiting list.

(ii) End a Ride:

Input: Taxi ID and end time.

Constraints:

- The system should mark the taxi as available.
- If there are users on the waiting list, the system should allocate the taxi to the next user in the queue.

(iii) View Available Taxis:

Output: A list of available taxis and their current locations.

(iv) View Ongoing Rides:

Output: A list of all ongoing rides including user details (name, pickup location, drop location, taxi ID, and start time).

(v) View Waiting List:

Output: A list of users on the waiting list including user details (name, pickup location, drop location, and booking time).

Detailed Requirements

Booking a Taxi

- The system should allow users to book a taxi by specifying their name, pickup location, drop location, and booking time.
- The system should determine the nearest available taxi to the pickup location.
- If multiple taxis are equidistant, the system should assign the one that becomes available the earliest.

- If no taxis are available, the system should add the user to a waiting list.
- The system should generate a unique booking ID for each booking.

Ending a Ride

- The system should allow drivers to mark a ride as completed by providing the taxi ID and end time.
- Upon ride completion, the system should update the taxi's availability.
- If there are users on the waiting list, the system should allocate the newly available taxi to the next user based on the queue.

Viewing Available Taxis

- The system should display a list of all available taxis and their current locations.

Viewing Ongoing Rides

- The system should display a list of all ongoing rides with the following details:
 - User's name
 - Pickup location
 - Drop location
 - Taxi ID
 - Start time

Viewing Waiting List

- The system should display a list of users on the waiting list with the following details:
 - User's name
 - Pickup location
 - Drop location
 - Booking time

Constraints

- The total number of taxis is fixed.
- The system should handle multiple users booking and ending rides concurrently.
- The waiting list should have a maximum capacity. If the waiting list is full, the system should notify the user that no more bookings can be taken at the moment.

Example Scenarios

Scenario 1: Booking a Taxi

Input: Name: Alice, Pickup Location: A, Drop Location: B, Booking Time: 10:00 AM

Output: Booking successful, Booking ID: 101, Assigned Taxi ID: T1

Scenario 2: Ending a Ride

Input: Taxi ID: T1, End Time: 10:30 AM

Output: Ride completed, Taxi T1 is now available

Scenario 3: Viewing Available Taxis

Output:

Taxi ID: T2, Current Location: C

Taxi ID: T3, Current Location: D

Scenario 4: Viewing Ongoing Rides

Output:

Name: Bob, Pickup Location: E, Drop Location: F, Taxi ID: T4, Start Time: 9:30 AM

Name: Charlie, Pickup Location: G, Drop Location: H, Taxi ID: T5, Start Time: 9:45 AM

Scenario 5: Viewing Waiting List

Output:

Name: Dave, Pickup Location: I, Drop Location: J, Booking Time: 10:05 AM

Name: Eve, Pickup Location: K, Drop Location: L, Booking Time: 10:10 AM

Assumptions

- The distances between locations are pre-determined and available for the system to calculate.
- Each taxi can only accommodate one ride at a time.
- The system maintains the integrity and consistency of booking and ride completion operations.