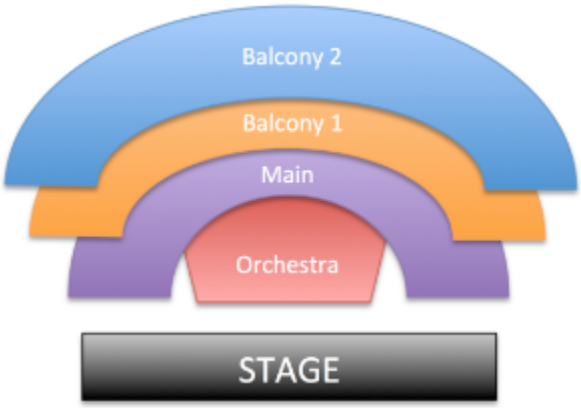
Ticket Service Homework

Implement a simple ticket service that facilitates the discovery, temporary hold, and final reservation of seats within a high-demand performance venue.

For example, see the seating arrangement and pricing details for a simple venue below.



Level Id	Level Name	Price	Rows	Seats in Row
1	Orchestra	$100.00	25	50
2	Main	$75.00	20	100
3	Balcony 1	$50.00	15	100
4	Balcony 2	$40.00	15	100

Instructions

Your homework assignment is to design and write a Ticket Service that provides the following functions:

- Find the number of seats available within the venue, optionally by seating level
  Note: available seats are seats that are neither held nor reserved.
- Find and hold the best available seats on behalf of a customer, potentially limited to specific levels
  Note: each ticket hold should expire within a set number of seconds.
- Reserve and commit a specific group of held seats for a customer

Requirements

- The ticket service implementation should be written in Java
- The solution and tests should build and execute entirely via the command line using either Maven or Gradle as the build tool
- A README file should be included in your submission that includes instructions for building the solution and executing the tests
- Implementation mechanisms such as disk-based storage, a REST API, and a front-end GUI are not strictly required

Your solution will be reviewed by engineers that you will be working with if you join the Walmart Technology team. We are interested in seeing your software design, coding, and testing skills.

For simplicity, we recommend that you implement the following interface. The design of the SeatHold object is entirely up to you.

```
public interface TicketService {

/**
* The number of seats in the requested level that are neither held nor reserved
*
* @param venueLevel a numeric venue level identifier to limit the search
* @return the number of tickets available on the provided level
*/
  int numSeatsAvailable(Optional<Integer> venueLevel);

/**
* Find and hold the best available seats for a customer
*
* @param numSeats the number of seats to find and hold
* @param minLevel the minimum venue level
* @param maxLevel the maximum venue level
* @param customerEmail unique identifier for the customer
* @return a SeatHold object identifying the specific seats and related
information
*/
  SeatHold findAndHoldSeats(int numSeats, Optional<Integer> minLevel,
Optional<Integer> maxLevel, String customerEmail);

/**

* Commit seats held for a specific customer
*
* @param seatHoldId the seat hold identifier
* @param customerEmail the email address of the customer to which the seat hold
is assigned
* @return a reservation confirmation code
*/
String reserveSeats(int seatHoldId, String customerEmail);

}
```