**Walmart Ticket Service Assignment**

I have focused on the **logic and designing the classes** keeping in mind **scalability, reusability, handling exceptions and multithreading** rather than on configuring persistence, Spring and REST API which I could have done with Spring Boot in STS**.**

I made all the shared resources thread-safe by using java concurrent collections and used an executor service to create a thread pool to time the seat holds, after the hold time the seats will be released if not reserved within that time. **Default seat hold time in the application is 10 seconds.**

**Implemented the 3 methods specified, which if placed in a REST web service, will work.**

ONLY for testing purpose (since there is a timer involved) I have combined fetch and reserve seats. But they are two completely different methods in the application.

**Maven Test:**

**Cd into the project folder and execute: *mvn clean install* to build the target jar and *mvn test* to run the test case. I have written one test case which test one scenario of complete cycle of discovery, hold and reserve of seats.**

**Testing:**

I have written a tester class that will run different scenarios of the application. For each scenario you would have to pass different set of arguments as explained below.

Cd into the target folder where the **ticketservice-0.0.1-ticketservice-SNAPSHOT.jar** is present.

Now you can run the commands in bold and you will see the results.

I have created 5 different scenarios to be executed as below:

1. Check availability of seats without passing a level: 1st argument = scenario to run

Abhisheks-MacBook-Pro-Retina:target abhishekashwathnarayanvenkat$ **java -jar ticketservice-0.0.1-ticketservice-SNAPSHOT.jar 1**

Seats available = 6250 across all levels

1. Check availability of seats by passing a level number: 1st argument = scenario to run, 2nd arg = level to search

Abhisheks-MacBook-Pro-Retina:target abhishekashwathnarayanvenkat$ **java -jar ticketservice-0.0.1-ticketservice-SNAPSHOT.jar 2 1**

Seats available = 1250 @ level 1

1. Find and hold seats (without min and max level): 1st argument = scenario to run, 2nd arg = seats to be held, 3rd arg = customer email ID

Abhisheks-MacBook-Pro-Retina:target abhishekashwathnarayanvenkat$ **java -jar ticketservice-0.0.1-ticketservice-SNAPSHOT.jar 3 5 abh@gmail.com**

5 seats held for abh@gmail.com- Reference Code is:d931f8bd-38b7-4615-a3bb-ab9004893524

holding seats with seatholdID:0

Seats held with ID:0 was putback due to timeout completed holding

1. Find and hold seats with a customer wait time (without min and max level):

1st argument = scenario to run, 2nd arg = seats to be held, 3rd arg = customer email ID, 4th arg = customer wait time in seconds

Abhisheks-MacBook-Pro-Retina:target abhishekashwathnarayanvenkat$ **java -jar ticketservice-0.0.1-ticketservice-SNAPSHOT.jar 4 5 abh 5**

5 seats held for abh- Reference Code is:8f5ed1b4-5089-4d0b-b848-ca9b38eac816

holding seats with seatholdID:0

customer starts waiting...

Seats available = 6245 across all levels

customer starts reservation

The held seats have been reserved for:abh:Reference Code is:8f5ed1b4-5089-4d0b-b848-ca9b38eac816

Seats held with ID:0 was reserved with refcode:8f5ed1b4-5089-4d0b-b848-ca9b38eac816

completed holding

1. Find and hold seats with a customer wait time (with min and max level):

1st argument = scenario to run, 2nd arg = seats to be held, 3rd arg = customer email ID, 4th arg = minlevel, 5th arg = max level, 6th arg = customer wait time in seconds

Abhisheks-MacBook-Pro-Retina:target abhishekashwathnarayanvenkat$ **java -jar ticketservice-0.0.1-ticketservice-SNAPSHOT.jar 5 5 abh 1 2 5**

5 seats held for abh- Reference Code is:75f80188-42fb-4c49-b556-600e742a4d03

holding seats with seatholdID:0

customer starts waiting...

Seats available = 6245 across all levels

customer starts reservation

The held seats have been reserved for:abh:Reference Code is:75f80188-42fb-4c49-b556-600e742a4d03

Seats held with ID:0 was reserved with refcode:75f80188-42fb-4c49-b556-600e742a4d03

completed holding

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**