



Ticket Service Application



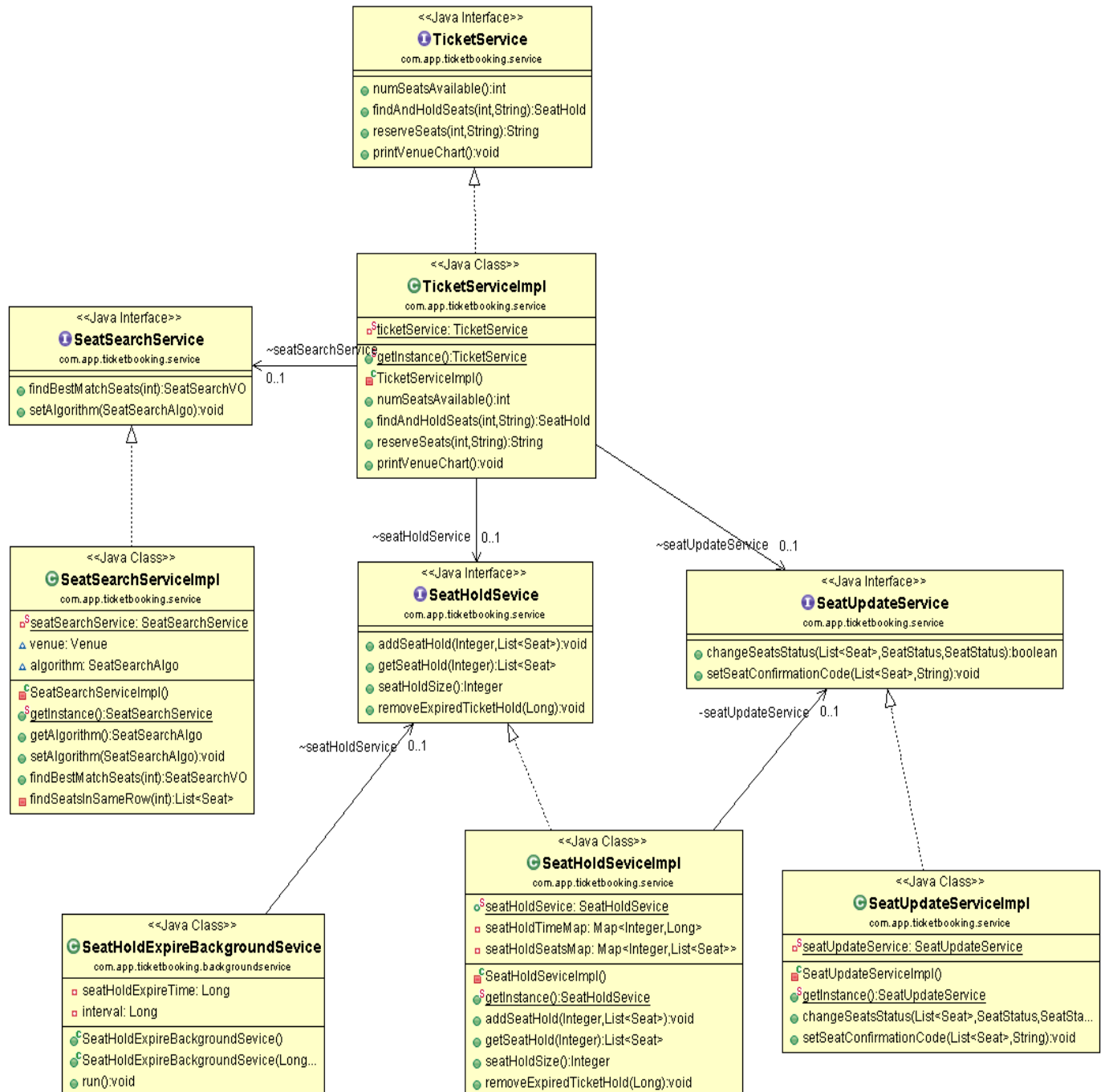
Bhaumik Patel
bkpatel3108@gmail.com

Table of Contents

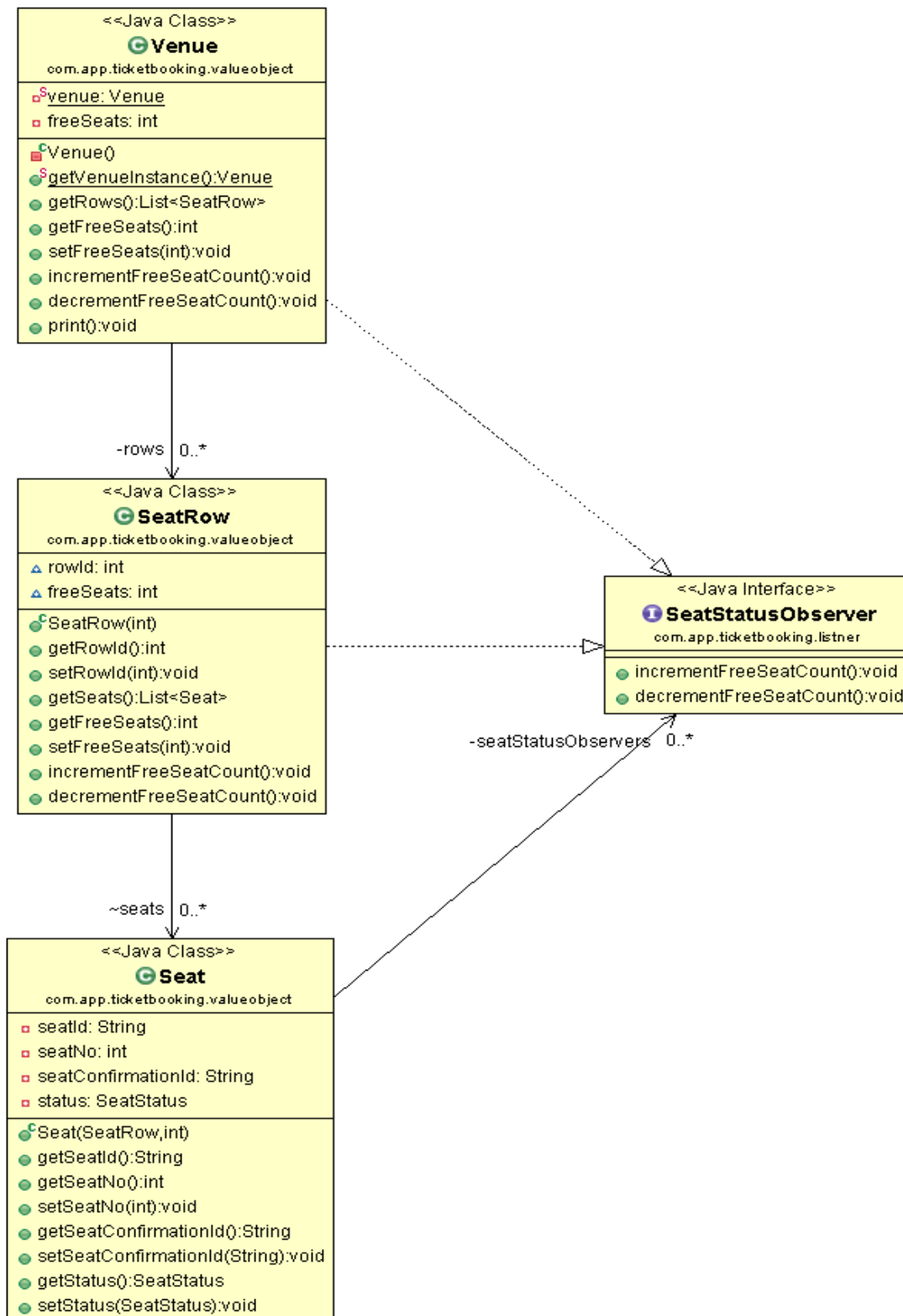
1. Understanding Class Design	3
1.1. Class Diagram - Service Classes	3
1.2. Class Diagram - Value Object Classes	4
1.3. Services Understanding	5
1.3.1. SeatHoldExpireBackgroundService	5
1.3.2. SeatHoldService	5
1.3.3. SeatUpdateService.....	5
1.3.4. SeatSearchService.....	5
1.4. Value objects understanding	5
1.4.1. Venue.....	5
1.4.2. SeatRow	5
1.4.3. Seat	6
1.4.4. SeatHold	6
2. Application Test and Build Instructions.....	7
2.1. Clone Repository	7
2.2. Change Directory to Application	7
2.3. To run tests	7
2.4. To Run Integration Tests and Build project	8
3. Application Run Instructions.....	9
4. Example of Using Application	10
4.1. Ticket Service Configuration	10
4.1.1. Using Default Configuration	10
4.1.2. Using Manual Configuration.....	11
4.2. Ticket Booking Console	11
4.2.1. To check no of seats available	12
4.2.2. To find and hold seats.....	12
4.2.3. To confirm hold seats	13
4.2.4. To print venue chart	13

1. Understanding Class Design

1.1. Class Diagram - Service Classes



1.2. Class Diagram - Value Object Classes



1.3. Services Understanding

1.3.1. SeatHoldExpireBackgroundService

This service runs on separate thread, and check for expired ticket after configured interval time.

Ticket considered expired if they are not booked within configured Ticket Hold Time.

1.3.2. SeatHoldService

This service used to hold new tickets for customer.
It is also used to remove hold form the tickets.

1.3.3. SeatUpdateService

This Service is used to update status of seats. It will first check given seats are currently in old status, and if they are in old status (double check), then it will update them with new status.

This service is used to add confirmation code to seats.

1.3.4. SeatSearchService

This service is used to find best seats for customer. It can be configured to use different algorithm. It uses default algorithm – Only book in same row.

1.4. Value objects understanding

1.4.1. Venue

It consists multiple seat rows (which can be configured).
It listens for seat status change and based on that it updated available free seats in venue.

1.4.2. SeatRow

It consists multiple seats (which can be configured).
It listens for seat status change and based on that it updated available free seats in seat row.

1.4.3. Seat

It consists seatId (unique identifier), confirmationId (to track which customer has reserved this seat - reservation code), and Observers for seat status change (venue and seat row object).

1.4.4. SeatHold

It consists seatHoldId (unique identifier), TickedHoldStatus (HELD, TIMEOUT), customer emailId, booked seat ids and error message (if any)

2. Application Test and Build Instructions

2.1. Clone Repository

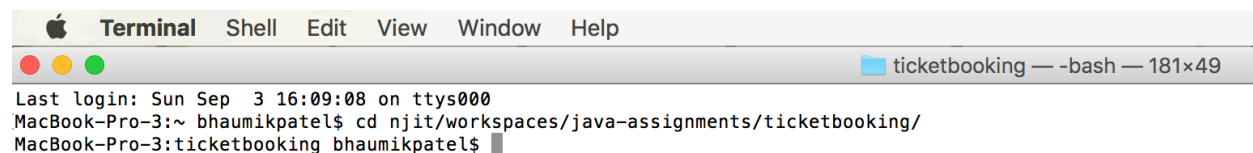
Clone following Repository to your computer.

<<GitHub URL given in email>>

2.2. Change Directory to Application

Open Terminal or Cmd, and change directory to Application (TicketService)

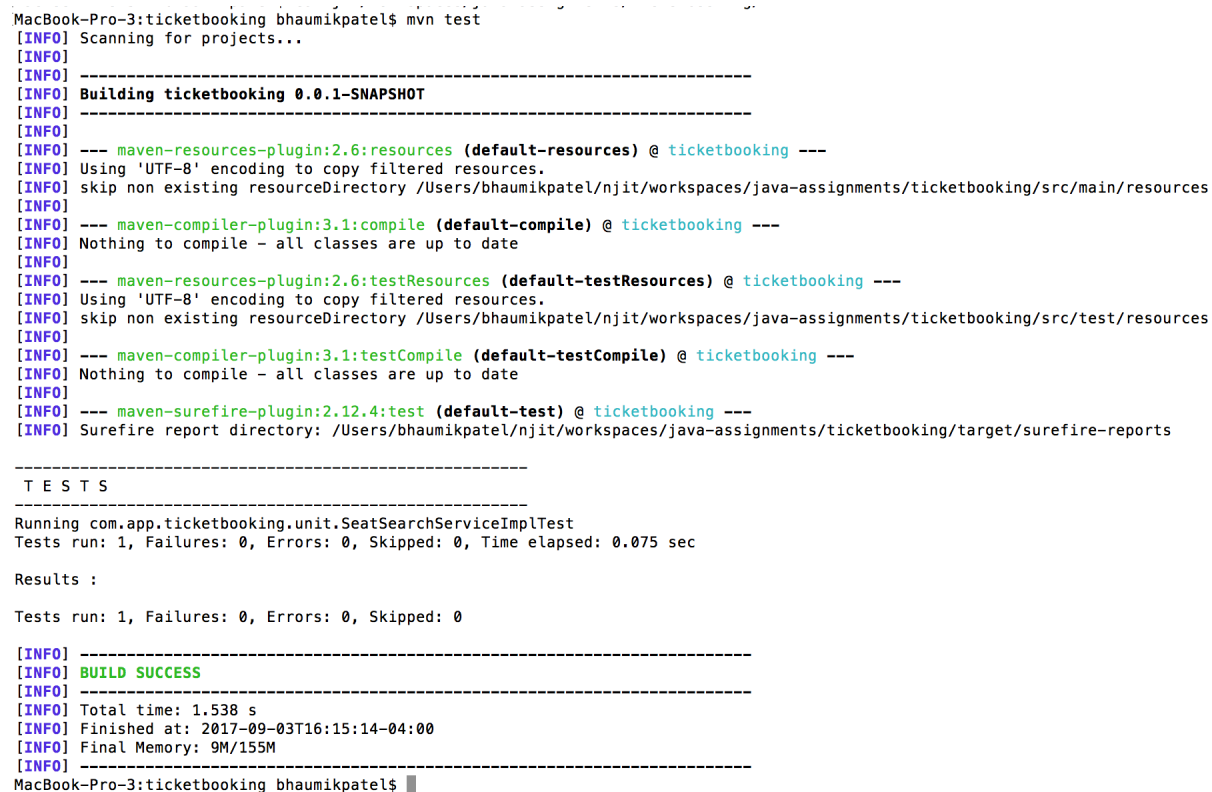
Ex : `$ cd njit/workspaces/java-assignments/ticketbooking/`

A screenshot of a macOS Terminal window. The title bar shows 'Terminal' with menu options 'Shell', 'Edit', 'View', 'Window', and 'Help'. The window title is 'ticketbooking — -bash — 181x49'. The terminal text shows the user 'bhaumikpatel' at 'MacBook-Pro-3' changing the directory to 'njit/workspaces/java-assignments/ticketbooking/' using the 'cd' command.

```
MacBook-Pro-3:~ bhaumikpatel$ cd njit/workspaces/java-assignments/ticketbooking/
MacBook-Pro-3:ticketbooking bhaumikpatel$
```

2.3. To run tests

`mvn test`

A screenshot of a macOS Terminal window showing the output of the 'mvn test' command. The output includes Maven lifecycle messages for scanning, building, compiling, and testing. It shows that the build was successful with one test passing. The terminal text is as follows:

```
MacBook-Pro-3:ticketbooking bhaumikpatel$ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building ticketbooking 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ticketbooking ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ticketbooking ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ticketbooking ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ticketbooking ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ticketbooking ---
[INFO] Surefire report directory: /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/target/surefire-reports

-----
T E S T S
-----
Running com.app.ticketbooking.unit.SeatSearchServiceImplTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.075 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.538 s
[INFO] Finished at: 2017-09-03T16:15:14-04:00
[INFO] Final Memory: 9M/155M
[INFO]
MacBook-Pro-3:ticketbooking bhaumikpatel$
```

2.4. To Run Integration Tests and Build project

\$ mvn install

```
MacBook-Pro-3:ticketbooking bhaumikpatel$ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building ticketbooking 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ticketbooking ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ ticketbooking ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ticketbooking ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ ticketbooking ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ticketbooking ---
[INFO] Surefire report directory: /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/target/surefire-reports

-----
T E S T S
-----
Running com.app.ticketbooking.unit.SeatSearchServiceImplTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.075 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ ticketbooking ---
[INFO] Building jar: /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/target/ticketbooking-0.0.1-SNAPSHOT.jar
[INFO] META-INF/maven/com.app.ticketbooking/pom.xml already added, skipping
[INFO] META-INF/maven/com.app.ticketbooking/pom.properties already added, skipping
[INFO]
[INFO] --- maven-failsafe-plugin:2.20:integration-test (failsafe-integration-tests) @ ticketbooking ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.app.ticketbooking.integration.TicketServiceIT
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 10.053 s - in com.app.ticketbooking.integration.TicketServiceIT
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent! The file encoding for reports output files should be provided by the POM property ${project.reporting.outputEncoding}.
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ ticketbooking ---
[INFO] Installing /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/target/ticketbooking-0.0.1-SNAPSHOT.jar to /Users/bhaumikpatel/.m2/repository/com/app/ticketbooking/0.0.1-SNAPSHOT/ticketbooking-0.0.1-SNAPSHOT.jar
[INFO] Installing /Users/bhaumikpatel/njit/workspaces/java-assignments/ticketbooking/pom.xml to /Users/bhaumikpatel/.m2/repository/com/app/ticketbooking/0.0.1-SNAPSHOT/ticketbooking-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 13.253 s
[INFO] Finished at: 2017-09-03T16:16:55-04:00
[INFO] Final Memory: 12M/202M
[INFO] -----
MacBook-Pro-3:ticketbooking bhaumikpatel$
```


3. Application Run Instructions

```
$ cd target/
```

```
$ java -cp ticketbooking-0.0.1-SNAPSHOT.jar com.app.ticketbooking.App
```

```
MacBook-Pro-3:ticketbooking bhaumikpatel$ ls
pom.xml src target
MacBook-Pro-3:ticketbooking bhaumikpatel$ cd target/
MacBook-Pro-3:target bhaumikpatel$ java -cp ticketbooking-0.0.1-SNAPSHOT.jar com.app.ticketbooking.App

----- Ticket Service Configuration -----
1 - For Default
2 - For Manual
0 - To Quit
Enter a number: █
```

4. Example of Using Application

4.1. Ticket Service Configuration

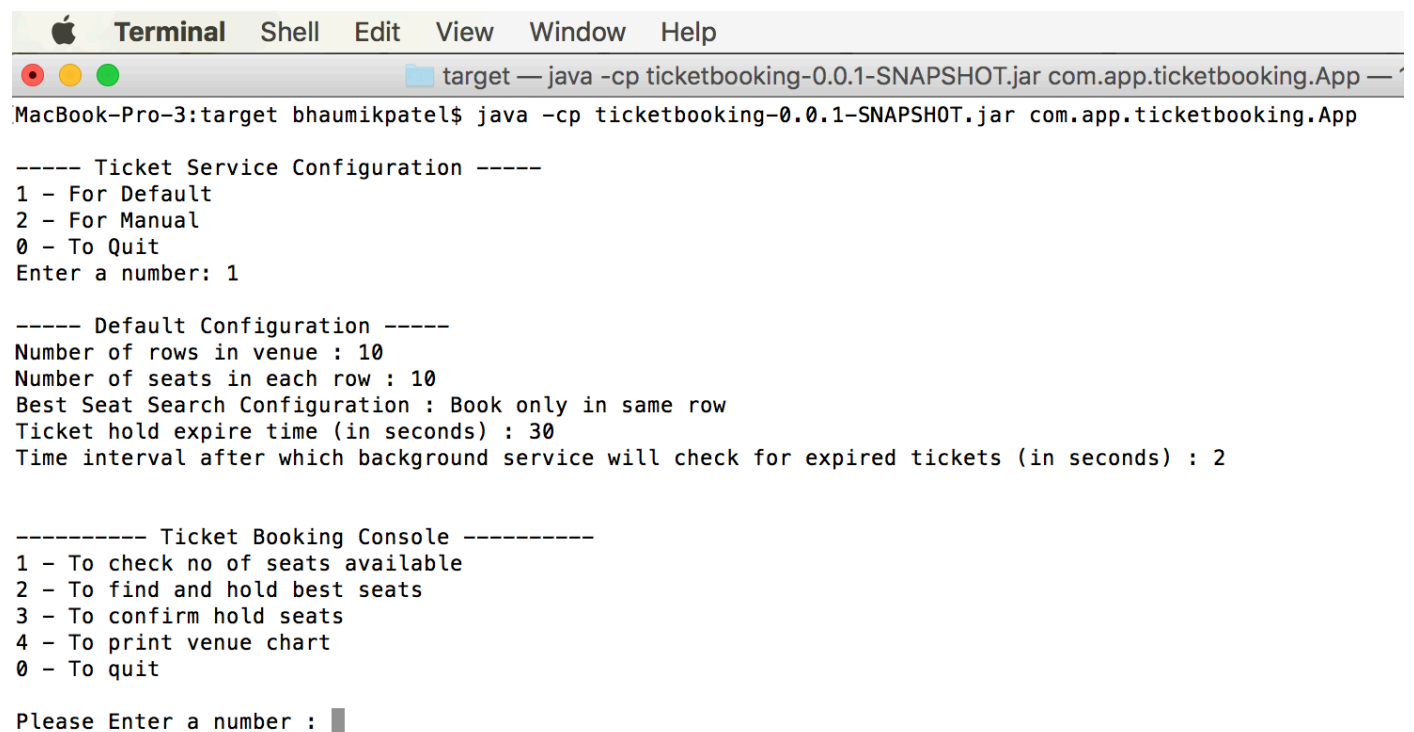
You can configure this application in 2 ways. Using quick configuration and manual configuration.

----- Ticket Service Configuration -----

- 1 - For Default
- 2 - For Manual
- 0 - To Quit

4.1.1. Using Default Configuration

Enter a number: 1



```
MacBook-Pro-3:target bhaumikpatel$ java -cp ticketbooking-0.0.1-SNAPSHOT.jar com.app.ticketbooking.App

----- Ticket Service Configuration -----
1 - For Default
2 - For Manual
0 - To Quit
Enter a number: 1

----- Default Configuration -----
Number of rows in venue : 10
Number of seats in each row : 10
Best Seat Search Configuration : Book only in same row
Ticket hold expire time (in seconds) : 30
Time interval after which background service will check for expired tickets (in seconds) : 2

----- Ticket Booking Console -----
1 - To check no of seats available
2 - To find and hold best seats
3 - To confirm hold seats
4 - To print venue chart
0 - To quit

Please Enter a number : █
```

4.1.2. Using Manual Configuration

Enter a number: 2

----- Ticket Service Configuration -----

1 - For Default

2 - For Manual

0 - To Quit

Enter a number: 2

----- Venue Configuration -----

Please enter number of rows in venue : 5

Please enter number of seats in each row : 8

----- Best Seat Search Configuration -----

1 - Book only in same Row

2 - Book max in Same Row(Coming Soon)

3 - Divide seats in equal Rows(Coming Soon)

Enter a number: 1

----- Ticket/Seat Hold Timeout Configuration -----

Please enter ticket hold expire time (in seconds) : 60

Please enter time interval after which background service will check for expired tickets (in seconds) : 5

4.2. Ticket Booking Console

On Ticket Booking Console, there are four options available.

----- Ticket Booking Console -----

1 - To check no of seats available

2 - To find and hold best seats

3 - To confirm hold seats

4 - To print venue chart

0 - To quit

Please Enter a number : █

4.2.1. To check no of seats available

```
----- Ticket Booking Console -----
1 - To check no of seats available
2 - To find and hold best seats
3 - To confirm hold seats
4 - To print venue chart
0 - To quit
```

Please Enter a number : 1

No of seats available : 100

4.2.2. To find and hold seats

```
----- Ticket Booking Console -----
1 - To check no of seats available
2 - To find and hold best seats
3 - To confirm hold seats
4 - To print venue chart
0 - To quit
```

Please Enter a number : 2

Please enter number of seats to find and hold : 9

Please enter email id of customer : b1

After entering number of seats to hold and customer email id, system will return ticket/seat hold details.

```
----- Ticket/Seat Hold Details -----
Seat Hold Id : 1000000000
Customr email id : b1
Seat hold status : HELD
Error message : null
1-1    1-2    1-3    1-4    1-5    1-6    1-7    1-8    1-9
```

4.2.3. To confirm hold seats

```
----- Ticket Booking Console -----
1 - To check no of seats available
2 - To find and hold best seats
3 - To confirm hold seats
4 - To print venue chart
0 - To quit
```

Please Enter a number : 3

Please enter seat hold id : 1000000001

Please enter email id of customer :b1

After entering seat hold id and customer email id, system will return confirmation code.

```
----- Ticket Reservation Confirmation Code -----
Confirmation code : Success#2000000000
```

4.2.4. To print venue chart

```
----- Ticket Booking Console -----
1 - To check no of seats available
2 - To find and hold best seats
3 - To confirm hold seats
4 - To print venue chart
0 - To quit
```

Please Enter a number : 4

```
----- Current Venue Booking -----
RESV  RESV  RESV  RESV  RESV  RESV  RESV  RESV  RESV  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE  FREE
-----
```