Table of Contents

[INTRODUCTION/TEST OBJECTIVE: 2](#_Toc445025716)

[ASSUMPTIONS: 2](#_Toc445025717)

[ENVIRONMENT DETAILS: 2](#_Toc445025718)

[ENTRANCE CRITERIA: 2](#_Toc445025719)

[SYSTEM TESTING: 3](#_Toc445025720)

[TEST SCENARIO: 5](#_Toc445025721)

[TEST CASES: 6](#_Toc445025722)

[EXIT CRITERIA: 12](#_Toc445025723)

[TESTING TOOL SUGGESTED 12](#_Toc445025724)

# INTRODUCTION/TEST OBJECTIVE:

The test objective is to determine if the developed software satisfies the business requirements. The main focus in this phase is to validate the following functionalities / Use cases:

1. Number of seats available within the venue
2. Hold the best available seats on behalf of a customer
3. Reserve and commit a specific group of held seats for a customer

# ASSUMPTIONS:

1. Functional requirements are locked and there are no changes involved during the testing phase.
2. Technical design review has been completed and approved.
3. The level’s, price, rows and the seats in row remain constant as stated in the below table:



1. The wait time to hold the reservation is set to be 5 minutes
2. Email id of the customer is the unique identifier to the system.
3. The customer can either login to discover, hold and reserve the seats or need to login at the point when they are ready to reserve the seats.
4. Credit or Debit is the mode of payment
5. Table structures to hold the customer details, status of seats are defined.
6. ‘STATUS’ will be the column that will hold the current status of the seats (FREE,HOLD,RESERVED)

# ENVIRONMENT DETAILS:

The software is developed using JAVA and a RDBMS.

# ENTRANCE CRITERIA:

* Ticket Service software meets all the Hardware / Software requirements.
* The final code is available for QA deployment before the initiation of testing cycle.
* Unit testing have been conducted and results inspected.
* System test plan have been approved and signed off.
* Test data have been mapped to the scenarios designed.

# SYSTEM TESTING:

System testing will focus on testing the functional requirements through the test scenarios designed. Testing techniques are selected based on the scenarios concluded for the application and they encompass Active testing, Agile testing, Age testing, Assertion testing, All – pair testing, Basis path testing.

\\ITWFPS05\mbalraj$\Downloads\ticketservice.png

# TEST SCENARIO:

UC001: Number of seats available within the venue

|  |  |
| --- | --- |
| **Scenario ID** | **Scenario description** |
|  | Validate that the customer is given the choice to view the available seats across all levels or a specific seating level. |
|  | Validate that the customer inputs a valid Level to view the seating details. |
|  | Validate that the correct number of seats available are retrieved for a given seating level |
|  | Validate when multiple customers are accessing the same details, the seats held by one customer are not available for another customer. |

UC002: Hold the best available seats on behalf of a customer

|  |  |
| --- | --- |
| **Scenario ID** | **Scenario description** |
|  | Validate that the customer chooses only the seats that are not already reserved or on hold. |
|  | Validate that the customer is able to select 1 or more available seat across different rows and different levels |
|  | Validate that the seats are held only during the ‘wait time’ and are released after the time expires. |

UC003: Reserve and commit a specific group of held seats for a customer

|  |  |
| --- | --- |
| **Scenario ID** | **Scenario description** |
|  | Validate that the customer’s email id is already available; if not request before reserving. |
|  | Validate that the customer selects a valid payment option which should include the credit / debit number validation to reserve the seats. |
|  | Validate that the customer reserves the held seats within the ‘wait time’ and the reservation is committed |

# TEST CASES:

| **Scenario ID** | **Test case ID** | **Test case Name** | **Test case Description** | **Test Step #** | **Test Description** | **Expected Result** | **Path covered** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TSD001 | TCD001 | Verify available Choices | Ticket Service provides the customer a choice to view the available seats across all levels or a specific seating level. | Pre  Condition | 1. Ensure the application under test is available and stable to test 2. Test data condition is understood (where there are only 4 applicable levels). |  | 0,1 |
|  |  |  |  | Step 1 | Run the application and select to discover seat availability | Application runs successfully and customer selects Discover Seat availability. |
|  |  |  |  | Step 2 | Customer is prompted if they need the availability of seats in all levels or from a specific level | On Choosing to discover available seats, application waits for customer input to either select ‘All’ or give a valid level id to view. |
| TSD002 | TCD002a | Verify Valid Level | Validate that the customer inputs a valid Level ID to view the seating details. | Pre-condition | Customer chooses to select a specific level |  | 0,1 |
|  |  |  |  | Step 1 | Customer chooses the level from 1/2/3/4 | The available seats are displayed from the specific level customer has requested.  Total available seats from specific levels = total available seats from specific levels – (total seats on hold + total seats reserved)  *DB Check: number of entries fetched with status = ‘FREE’ for a specific level* | 0,1,2 |
| TSD002 | TCD002b | Verify Valid Level | Validate that the customer inputs a valid Level ID to view the seating details. | Pre-condition | Customer chooses to select a specific level |  |  |
|  |  |  |  | Step 1 | Customer chooses a level apart from 1-4. | Error Message should be displayed as “Inappropriate Level ID, choose values from 1-4” | 0,1,2,3 |
| TSD003 | TCD003 | Seat Availability | Validate that the correct number of seats available are retrieved for a given seating level | Step 1 | Customer chooses to view the availability from all levels  Input value: All | Available seats across all levels should be listed which should not include either the reserved or on hold seats.  *Note: the total available seats at any time should not exceed 6500. (Sum of (cross product of rows & seats) from all levels.)*  *Total available seats for holding / reserving from all levels = total available seats from all levels – (total seats on hold from all levels + total seats reserved from all levels)*  *DB Check: number of entries fetched with status = ‘FREE’ for all levels* | 0,1,4 |
|  |  |  |  | Step 2 | Customer chooses to view the availability in a specific level  Input values from (1/2/3/4). | Available seats from the selected specific level should be listed which should not include either the reserved or on hold seats  Note: the total available seats at any time should not exceed the cross product of rows and seats in the opted level.  Total available seats from specific levels = total available seats from specific levels – (total seats on hold + total seats reserved)  *DB Check: number of entries fetched with status = ‘FREE’ for a specific level* | 0,1,2,4 |
| TSD004 | TCD004 | Multiple Customers | Validate when 2 or more customers are accessing the same details, the seats held by one customer are not available for the other customer. | Pre-Condition | 2 Customers are viewing the seats available for a specific level at the same time; where customer1 has held few seats. |  |  |
|  |  |  |  | Step 1 | Customer2 is viewing the details for the same level as customer 1. | The seats held by customer1 should not be displayed for customer2 | 0,1,2 |
| TSH001 | TCH001a | Verify Seat Selection | Validate that the customer chooses only the seats that are not already reserved or on hold. | Pre-condition | Seat details are displayed as per the customer’s choice of level |  |  |
|  |  |  |  | Step 1 | Customer selects the seat number from the specific level which are available (not held or reserved) | The selected seats are put on hold for the customer.  The status is flipped from ‘Free’ to ‘Hold’ in the backend | 0,1,2,4,5,6,7 |
|  | TCH001b | Verify Seat Selection | Validate that the customer chooses only the seats that are not already reserved or on hold. | Pre-condition | Seat details are displayed as per the customer’s choice of level |  |  |
|  |  |  |  | Step 1 | Customer selects the seat number from the specific level which are held or reserved | Customer is informed that “The selected seats are either reserved or on hold” | 0,1,2 |
| TSH002 | TCH002 | Verify Multiple selection of seats | Validate that the customer is able to select more than 1 available seat across different rows and different levels | Pre-condition | All the available seats are displayed for customer selection |  |  |
|  |  |  |  | Step 1 | Customer chooses multiple seats across different rows and different level | The selected seats are put on Hold for the customer for a preset time before he confirms his reservation.  The status is flipped from ‘Free’ to ‘Hold’ in the backend | 0,1,4,5 |
| TSH003 | TCH003 | Wait time | Validate that the seats are held only during the ‘wait time’ and are released after the time expires. | Pre-Condition | Customer has selected the available seats displayed from the list |  |  |
|  |  |  |  | Step 1 | Customer has chosen multiple seats across different levels | 1. Customer is informed that the selected seats will be held for them for only 5 minutes 2. The timer is set on from the moment the customer has selected the seats and their status remains ‘Hold’ for a wait time of 5 minutes. 3. Once 5 minutes are exhausted, the customer is informed to discover available seats again 4. The status in the backend flips back to ‘Free’ | 0,1,4,5,6,7,8 |
| TSR001 | TCR001 | Verify Email ID | Validate that the customer’s email id is already available; if not request before reserving. | Pre-condition | Customer proceeds to reserve the seats that were held by him before the wait time exhausts. |  |  |
|  |  |  |  | Step 1 | Customer has logged in to discover / hold the seats | Customer will be let to complete reservation after following the valid payment method. | 0,1,4,5,6,7,8,9,10,11,13 |
|  |  |  |  | Step 2 | Customer was directly viewing the details without logging in | Customer will be prompted to login with his email id. |  |
| TSR002 | TCR002 |  | Validate that the customer selects a valid payment option to reserve the seats. | Pre-condition | Customer has provided his email id and his wait time has not yet exhausted |  |  |
|  |  |  |  | Step 1 | Customer is given the choice of Credit / Debit as the payment option and he chooses one of them. | On choosing the Payment option, customer is prompted to input further details to complete the transaction | 0,1,4,5,6,7,8,9,10,11,12 |
| TSR003 | TCR003 | Reserve Seats and Commit | Validate that the customer reserves the held seats within the ‘wait time’ and that the reservations are committed | Pre-Condition | Customer moves to reserve the seats within the WAIT time and has selected the valid payment option |  |  |
|  |  |  |  | Step 1 | Customer inputs valid payment details and the held seats are reserved | The Seats held by the customer are reserved and the Status in Backend is flipped to “RESERVED”. | 0,1,4,5,6,7,8,9,10,11,13,14,15 |
|  |  |  |  | Step 2 | Once the Seats are reserved another customer is not allowed to select them again | Application should not list the reserved seats when another customer chooses to discover the available seats. | 0,1,4,5,6,7,8,9,10,11,13,14,15,16 |

# EXIT CRITERIA:

* All the functional requirements are tested successfully
* All the defects logged during the testing phase have a fix and have been successfully retested and closed.
* IT Evidence of the test scenarios are collected and inspected
* Results are incorporated in to Test summary report

# TESTING TOOL SUGGESTED

Selenium is the suggested testing tool that can be used for automation testing as it’s adaptable to any changes that might be introduced in the requirements. Data driven / Module based framework would best suit this application.