***Non Functional Test Plan***

*Table of Contents*

1. *Introduction*
2. *Test Scenarios*
   1. *numSeatsAvailable API Cases*
   2. *findAndHoldSeats API Cases*
   3. *reserveSeats API Cases*
   4. *Common Cases*
3. *Automation*
4. *To-Do*
5. *References*

*Introduction*

The purpose of this document is to capture use cases for functional tests which are required to test all components of the Ticket Service system according to the given requirements.

**Test Scenarios**

*FT1: numSeatsAvailable API Cases*

|  |  |  |  |
| --- | --- | --- | --- |
| *Test#* | *Description* | *Steps* | *Expected Results* |
| *FT1-1* | *Basic Test: No reservation/holds are available check the number of seats available* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> | *# of seats returned by API=Max* |
| *FT1-2* | *Basic Test: All seats are either reserved or on holds check the number of seats available* | *Precondition:*  *#of seats available for booking = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> | *# of seats returned by API=0* |
| *FT1-3* | *All seats are either reserved or on hold except the 4PM shows on all days* | *Precondition:*  *#of seats available for booking at 8PM on all days = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT1-4* | *All seats are either reserved or on hold except the 8PM shows on all days* | *Precondition:*  *#of seats available for booking at 4PM on all days = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT1-5* | *All seats are either reserved or on hold except the 4PM shows on a selected day* | *Precondition:*  *#of seats available for booking at 8PM on 1/1/2018 = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT1-6* | *All seats are either reserved or on hold except the 8PM shows on a selected day* | *Precondition:*  *#of seats available for booking at 4PM on 1/1/2018 = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT1-7* | *Check seat calculation logic in Hold queue* | *Precondition:*  *#of seats in Hold queue = 4*  *#of seats in Reserve queue = 0*  *#of seats booked = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT1-8* | *Check seat calculation logic in Reserve queue* | *Precondition:*  *#of seats in Reserve queue = 10*  *#of seats in Hold queue = 0*  *#of seats booked = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT1-9* | *Check Seat selection logic in Hold and Reserve queue* | *Precondition:*  *#of seats in Hold queue = 4 and Reserve queue = 10*  *#of seats booked = 0*  *Test:*  <https://ticketsystem.com/numSeatsAvailable> |  |

*FT2: findAndHoldSeats API Cases*

|  |  |  |  |
| --- | --- | --- | --- |
| *Test#* | *Description* | *Steps* | *Expected Results* |
| *FT2-1* | *Basic Test: User holds 15 seats with customerEmail address=abc@abc.com* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=15&customerEmail=abc@abc.com* | *# of seats held by API=15 for emaol=abc@abc.com* |
| *FT2-2* | *Call API with empty/Null/non integer numSeats parameter* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=abc&customerEmail=abc@abc.com*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=&customerEmail=abc@abc.com*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:customerEmail=abc@abc.com* |  |
| *FT2-3* | *Call API with empty/Null/non match customerEmail parameter* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=15&customerEmail=*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=15&customerEmail=1*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=15* |  |
| *FT2-4* | *Call API with empty/Null/non match numSeats and customerEmail parameters* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:numSeats=&customerEmail=*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:numSeats=sfs&customerEmail=fs* |  |
| *FT2-5* | *Call API with numSeats, customerEmail and 2 extra parameters* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=15&customerEmail=abc@abc.com&extra1=12&extra2=23* |  |
| *FT2-6* | *Combine API Scenario: A User selects 2 seats, the tickets were the last available tickets in the system. Another User calls Discover API* | *Precondition:*  *#of seats available for booking = 2*  *Test:*  *User 1:* [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=2&customerEmail=abc@abc.com*  *User 2:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT2-7* | *Combine API Scenario: User selects 2 seats, does not like the seats returned by the system, clicks refresh again. Another user calls Discover API* | *Precondition:*  *#of seats available for booking = 2*  *Test:*  *User 1:* [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=2&customerEmail=abc@abc.com*  *Wait 400 seconds*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=2&customerEmail=abc@abc.com*  *User 2:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT2-8* | *Combine API Scenario: User selects 2 seats and before the*  *server response is received, user loses internet, the tickets were the last available tickets in the system. Another user calls Discover API* | *Precondition:*  *#of seats available for booking = 2*  *Test:*  *User 1:* [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*findAndHoldSeats:seats=2&customerEmail=abc@abc.com*  *Disconnect from internet*  *User 2:*  <https://ticketsystem.com/numSeatsAvailable> |  |

*FT3: reserveSeats API Cases*

|  |  |  |  |
| --- | --- | --- | --- |
| *Test#* | *Description* | *Steps* | *Expected Results* |
| *FT3-1* | *Basic Test: User reserves 15 seats with customerEmail address=abc@abc.com* | *Precondition:*  *#of seats available for booking = Max*  *holdid=4*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4&customerEmail=abc@abc.com* |  |
| *FT3-2* | *Call API with empty/Null/non integer holdId parameter* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=&customerEmail=abc@abc.com*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4ljjf&customerEmail=abc@abc.com*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:customerEmail=abc@abc.com* |  |
| *FT3-3* | *Call API with empty/Null/non match customerEmail parameter* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4&customerEmail=*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4&customerEmail=78* |  |
| *FT3-4* | *Call API with empty/Null/non match holdId and customerEmail parameters* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=&customerEmail=*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4jk&customerEmail=90*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats* |  |
| *FT3-5* | *Call API with holdId, customerEmail and 2 extra parameters* | *Precondition:*  *#of seats available for booking = Max*  *Test:*  [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=4&customerEmail=abc@abc.com&extra1=12&extra2=23* |  |
| *FT3-6* | *Combine API Scenario: A User registers 2 seats, the tickets were the last available tickets in the system. Another User calls Discover API* | *Precondition:*  *#of seats available for booking = 2*  *Test:*  *User 1:* [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=2&customerEmail=abc@abc.com*  *User 2:*  <https://ticketsystem.com/numSeatsAvailable> |  |
| *FT3-8* | *Combine API Scenario: User selects 2 seats and before the*  *server response is received, user loses internet, the tickets were the last available tickets in the system. Another user calls Discover API* | *Precondition:*  *#of seats available for booking = 2*  *Test:*  *User 1:* [https://ticketsystem.com/](https://ticketsystem.com/numSeatsAvailable)*reserveSeats:holdId=2&customerEmail=abc@abc.com*  *Disconnect from internet*  *User 2:*  <https://ticketsystem.com/numSeatsAvailable> |  |

*FT4: Common Cases*

*These tests are common to all 3 API above*

|  |  |  |  |
| --- | --- | --- | --- |
| *Test#* | *Description* |  | *Expected Results* |
| *FT4-1* | *Check Malformed URL* |  |  |
| *FT4-2* | *Check Return Type* |  |  |
| *FT4-3* | *Check Response Code* |  |  |
| *FT4-4* | *Check JSON format* |  |  |
| *FT4-5* | *Check Request Header* |  |  |
| *FT4-6* | *Check Response Header* |  |  |
| *FT4-7* | *Log File Verification* | *Benchmark Info, warnings, errors and exceptions. Enable debug mode* |  |
| *FT4-8* | *Miscellaneous* | *Using browsers developers tool check console logs, network logs, performance logs, Memory logs, Security logs Tab. API time outs should be handled gracefully.* |  |

*Automation:*

Using Java and TestNG along with JSON parser, the tests marked with for automation can be scripted.

*To-Do:*

Document exit criteria

Document mobile cases

*References:*